Journal of Machine Learning Research 25 (2024) 1-91 Submitted 4/19; Revised 12/23; Published 01/24

On Truthing Issues in Supervised Classification*

Jonathan K. Su SU@LL.MIT.EDU
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02421-6426, USA

Editor: Shivani Agarwal, Francis Bach

Abstract

Ideal supervised classification assumes known correct labels, but various truthing issues can arise in practice: noisy labels; multiple, conflicting labels for a sample; missing labels; and different labeler combinations for different samples. Previous work introduced a *noisy-label model*, which views the observed noisy labels as random variables conditioned on the unobserved correct labels. It has mainly focused on estimating the conditional distribution of the noisy labels and the class prior, as well as estimating the correct labels or training with noisy labels. In a complementary manner, given the conditional distribution and class prior, we apply estimation theory to classifier testing, training, and comparison of different combinations of labelers. First, for binary classification, we construct a testing model and derive approximate marginal posteriors for accuracy, precision, recall, probability of false alarm, and F-score, and joint posteriors for ROC and precision-recall analysis. We propose *minimum mean-square error* (MMSE) *testing*, which employs empirical Bayes algorithms to estimate the testing-model parameters and then computes optimal point estimates and credible regions for the metrics. We extend the approach to multi-class classification to obtain optimal estimates of accuracy and individual confusion-matrix elements. Second, we present a unified view of training that covers probabilistic (i.e., discriminative or generative) and non-probabilistic models. For the former, we adjust maximum-likelihood or maximum *a posteriori* training for truthing issues; for the latter, we propose *MMSE training*, which minimizes the MMSE estimate of the empirical risk. We also describe suboptimal training that is compatible with existing infrastructure. Third, we observe that mutual information lets one express any labeler combination as an equivalent single labeler, implying that *multiple mediocre labelers can be as informative as, or more informative than, a single expert labeler*. Experiments demonstrate the effectiveness of the methods and confirm the implication.

Keywords: supervised classification, truth errors, noisy labels, Bayesian estimation, empirical Bayes, mutual information, crowdsourcing

*. DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

1. Introduction

Supervised classification uses labeled data to train and test a predictive model or *classifier*, which will be used to predict the labels of unlabeled data. The predictive model is a mapping $g : \mathcal{X} \to \mathcal{Y}$, parameterized by $\boldsymbol{\theta}$, from a feature space $\mathcal{X}$ to a set $\mathcal{Y} = \{0, 1, \ldots, C-1\}$ of C mutually exclusive and exhaustive *classes* or *labels*. The input to the model is a *feature vector* $\boldsymbol{x} \in \mathcal{X}$, and the output of the model is the *predicted label* $\hat{y} = g(\boldsymbol{x}; \boldsymbol{\theta}) \in \mathcal{Y}$, which may or may not agree with the *correct label* $y \in \mathcal{Y}$. A *labeled sample* $(\boldsymbol{x}, y)$ consists of a feature vector $\boldsymbol{x}$ and its corresponding correct label y, while an *unlabeled sample* is just a feature vector $\boldsymbol{x}$. A set of N labeled samples is denoted as $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$, where $\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = (\boldsymbol{x}_i)_{i=1}^N$, $\boldsymbol{y} = (y_i)_{i=1}^N$, and for each i, $\boldsymbol{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, where y_i is the correct label associated with $\boldsymbol{x}_i$. The result of applying a learned model to each feature vector in $\underline{\boldsymbol{x}}$ is the list of corresponding predicted labels $\hat{\boldsymbol{y}} = (\hat{y}_i)_{i=1}^N = (g(\boldsymbol{x}_i; \boldsymbol{\theta}))_{i=1}^N$.

Training learns the model from a training set of labeled samples. *Testing* uses a separate testing set of labeled samples, applies the learned model to its feature vectors, and calculates metrics that quantify the agreement between each predicted label and the corresponding correct label. Our notation does not distinguish between the training and testing sets, since the intended set will be clear from context.

1.1 Truthing Issues

It is usually assumed that the correct labels are known during training and testing; we refer to this situation as the *ideal case*. However, this critical assumption is often violated in practice (see Everingham et al. (2006, p. 172), Frénay and Verleysen (2014), or Northcutt et al. (2021b), for example), which can degrade the trained model and produce misleading testing metrics. Deviation from the ideal case can invalidate many hours of hard work and computer processing, as well as make users and clients skeptical of the utility of a classifier or its performance.

Consequently, substantial resources are often devoted to *truthing*, the process of labeling data as correctly as possible. We use the term *labeler* to mean an entity that assigns labels to samples; some authors use terms like "teacher" or "annotator." A labeler could be a human, a sensor, a laboratory test, or even another classifier. Despite a labeler's best efforts, errors can and do still occur. Humans make mistakes, become fatigued, and have varying amounts of expertise, attentiveness, and motivation; sensors are subject to noise, occlusion, and other degradations; laboratory test results are not always definitive; and classifiers are rarely perfect predictors. In addition, labeling via crowdsourcing means that more than one labeler could assign a label to the same feature vector $\boldsymbol{x}_i$, and if the labels conflict, then at least one of them must be incorrect.

In short, a number of *truthing issues* can arise: truth errors, multiple labelers who provide conflicting labels for the same sample, missing labels, and different combinations of labelers for different samples. We introduce some notation here and include an example of it in Table 1. We assume there are T labelers, indexed from 1 to T, and we let $\mathcal{T} = \{1, 2, \ldots, T\}$. Let $z_{i,t} \in \{\emptyset\} \cup \mathcal{Y}$, $t \in \mathcal{T}$, denote the *noisy label* assigned to $\boldsymbol{x}_i$ by the t^{th} labeler, where $z_{i,t} = \emptyset$ if no label was assigned. We require that at least one labeler assigns a label to each sample; that is, $z_{i,t} \in \mathcal{Y}$ for some $t \in \mathcal{T}$. Hence, a labeler can only assign one label

			Labeler Index t				
			1	2	3	4	5
Sample Index i	Feature Vector $\boldsymbol{x}_i$	Correct Label y_i	Noisy Labels $\boldsymbol{z}_i$				
			$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$z_{i,4}$	$z_{i,5}$
1	$\boldsymbol{x}_1$	0	0	1	0	0	$\emptyset$
2	$\boldsymbol{x}_2$	1	1	$\emptyset$	$\emptyset$	1	$\emptyset$
3	$\boldsymbol{x}_3$	0	0	0	0	0	0
4	$\boldsymbol{x}_4$	1	1	1	0	0	1
5	$\boldsymbol{x}_5$	0	0	$\emptyset$	$\emptyset$	0	0
$\vdots$	$\vdots$	$\vdots$			$\vdots$		
N	$\boldsymbol{x}_N$	0	0	0	$\emptyset$	0	$\emptyset$

Table 1: Example of notation for binary classification and five labelers. This example corresponds to the training example in Section 5.4.

to a sample. Of course, $z_{i,t}$ might be incorrect and differ from y_i. Denote the set of noisy labels for $\boldsymbol{x}_i$ by $\boldsymbol{z}_i = (z_{i,t})_{t\in\mathcal{T}}$; the noisy labels need not agree. Finally, let $\underline{\boldsymbol{z}} = (\boldsymbol{z}_i)_{i=1}^N$.

A natural approach, introduced by Dawid and Skene (1979), is to treat the correct labels and noisy labels as *random variables* (RVs). We adopt this viewpoint and use capitalization (e.g., Y) for an RV and lowercase (e.g., y) for its non-random counterpart. Hence, Y_i is the *correct-label RV* for the i^{th} sample, so $\boldsymbol{Y} = (Y_i)_{i=1}^N$ is the list of correct-label RVs, and $\boldsymbol{y}$ is the list of correct labels themselves. Similarly, the feature-vector RVs are $\boldsymbol{X}_i$ and $\underline{\boldsymbol{X}} = (\boldsymbol{X}_i)_{i=1}^N$, and their realizations are $\boldsymbol{x}_i$ and $\underline{\boldsymbol{x}}$. Also, $Z_{i,t}$, $\boldsymbol{Z}$, $\boldsymbol{Z}_i = (Z_{i,t})_{t\in\mathcal{T}}$, and $\underline{\boldsymbol{Z}} = (\boldsymbol{Z}_i)_{i=1}^N$ indicate *noisy-label RVs*, while $z_{i,t}$, $\boldsymbol{z}$, $\boldsymbol{z}_i$, and $\underline{\boldsymbol{z}}$ indicate their respective realizations.

We assume that the RVs associated with different samples are independent and identically distributed and that the correct-label RVs have class prior $\boldsymbol{\pi} = \big(\pi(y)\big)_{y=0}^{C-1}$, where $\pi(y) = p(y) = \Pr(Y = y)$. We further assume that the noisy-label RVs are conditionally dependent given the correct-label RV, and we denote the *noisy-label conditional distribution* by $p(\boldsymbol{z}|y;\boldsymbol{\psi})$, which is parameterized by $\boldsymbol{\psi}$, and where a semi-colon separates RVs from non-random parameters. We refer to $p(\boldsymbol{z}|y;\boldsymbol{\psi})\pi(y)$ as a *noisy-label model*. For the i^{th} sample, the parameters are $\boldsymbol{\psi}_i$. We assume that $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ follow the same conditional distribution $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ if $\boldsymbol{\psi}_i = \boldsymbol{\psi}_j = \boldsymbol{\psi}$. At this stage, we let the parameters remain completely general. Also, let $\underline{\boldsymbol{\psi}} = (\boldsymbol{\psi}_i)_{i=1}^N$ be the list of parameters for all samples.

Much of the related work, described in Section 1.2, has focused on constructing a model for $p(\boldsymbol{z}|y;\boldsymbol{\psi})$, estimating the parameters $\boldsymbol{\psi}$ and class prior $\boldsymbol{\pi}$, and estimating the correct-label RVs $\boldsymbol{Y}$. This paper and the related work implicitly assume that $p(\boldsymbol{z}|y;\boldsymbol{\psi}) \neq p(\boldsymbol{z};\boldsymbol{\psi})$ so that the noisy-label RVs provide some information about the correct-label RV. If $\boldsymbol{z}$ is non-informative, then none of these methods will be effective. Denote the *estimated correct label* for the i^{th} sample as $\check{y}_i$, and let $\check{\boldsymbol{y}} = (\check{y}_i)_{i=1}^N$.

In this paper, we seek answers to the following questions:

1. *How can one test a classifier in the presence of truthing issues*[1] *?*
2. *How can one train a classifier in the presence of truthing issues?*
3. *How can one compare different combinations of labelers with different abilities?*

1.2 Related Work

We organize the related work into noisy-label modeling and learning, training with truthing issues, testing with truthing issues, and comparing combinations of labelers.

1.2.1 Noisy-Label Modeling and Learning

Several authors have proposed noisy-label models $p(\boldsymbol{z}|y;\boldsymbol{\psi})\pi(y)$ and developed methods for learning them,[2] i.e., estimating $\boldsymbol{\psi}$ and $\boldsymbol{\pi}$. Often, they also estimated the correct-label RVs $\boldsymbol{Y}$. Table 2 presents a summary. All but one set of authors allowed for multiple labelers. The table shows that most of them assumed that the noisy-label RVs are conditionally independent given the correct-label RV.

Dawid and Skene (1979) made an initial, influential foray into this area. They did not address supervised classification, but they considered the problem of compiling observations of patients from multiple clinicians who might disagree or make mistakes. In this context, they introduced the conditional-RV formulation that is now commonplace.

When the correct labels are available, they showed that *maximum-likelihood* (ML) estimation of $\boldsymbol{\psi}$ and $\boldsymbol{\pi}$ is straightforward. With the benefit of hindsight, it is clear that this situation amounts to learning a classifier from an auxiliary data set $\{\underline{\boldsymbol{z}}', \boldsymbol{y}'\}$ where the noisy labels $\boldsymbol{z}'$ serve as the feature vector. For example, one might have a small auxiliary data set containing noisy labels from clinicians and correct labels from a separate, gold-standard laboratory test, and a large amount of data with only noisy labels.

More important, when the correct labels are unavailable, Dawid and Skene demonstrated the utility of the *expectation-maximization* (EM) algorithm of Dempster et al. (1977) for estimating $\boldsymbol{\psi}$, $\boldsymbol{\pi}$, and $\boldsymbol{Y}$. Many other authors have built upon this work.

Donmez et al. (2010) considered classification and regression, and they also proposed using the EM algorithm for learning the noisy-label model. They studied conditions under which ML estimates of the noisy-label model parameters are consistent. For classification, they showed that consistency holds under certain conditions, such as when the labelers are weak learners or better and the class prior $\boldsymbol{\pi}$ is not equiprobable.

Whitehill et al. (2009) proposed a model for $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ that considers sample difficulty and labeler expertise. A Bayesian extension of the model allowed priors for these parameters. They used both the EM algorithm and a Bayesian version of it to estimate $\boldsymbol{\psi}$ and $\boldsymbol{Y}$. **Welinder and Perona (2010)** and **Welinder et al. (2010)** expanded the model to include labeler bias and to support multi-class labels and continuously-valued annotations.

1. We tackle testing before training because our testing approach applies regardless of how a classifier was trained. Also, if truthing issues are present, then there is little point in training a classifier if one cannot evaluate its performance.
2. One could propose learning a model of the form $p(\boldsymbol{z}|\boldsymbol{x},y)p(\boldsymbol{x},y)$ or $p(\boldsymbol{z}|y,\boldsymbol{x})p(y|\boldsymbol{x})$ from an auxiliary set $\{\underline{\boldsymbol{x}}', \boldsymbol{y}', \underline{\boldsymbol{z}}'\}$. However, learning it would be harder than learning a predictive model for $p(\boldsymbol{x},y)$ or $p(y|\boldsymbol{x})$, so if such a set were available, then one could just learn the predictive model from $\{\underline{\boldsymbol{x}}', \boldsymbol{y}'\}$.

Reference	Description	Number of Classes	Labeler Dependence
Dawid and Skene (1979)	• Seminal work • ML estimation of noisy-label model, if correct labels available • EM estimation of noisy-label model and correct labels, if correct labels unavailable	Multiple	Independent
Donmez et al. (2010)	• Also regression • EM estimation of noisy-label models • Consistency conditions for ML estimates	Multiple	Independent
Whitehill et al. (2009)	• *Sample-difficulty and labeler-expertise model* • EM or Bayesian estimation of noisy-label model	Binary	Independent
Welinder and Perona (2010); Welinder et al. (2010)	Extension of Whitehill et al. (2009) to include labeler bias, multiple classes, and continuously-valued annotations	Binary or Multiple	Independent
Branson et al. (2017)	• Extension of Welinder et al. (2010); Welinder and Perona (2010) to part-keypoint and bounding-box annotations • Sequential acquisition of noisy labels	Binary	Independent
Van Horn et al. (2018)	Extension of Branson et al. (2017) to multiple classes and sequentially dependent labelers	Multiple	Sequentially dependent
Karger et al. (2014)	• Allocation of labelers to unlabeled samples • Iterative message passing	Binary	Independent
Zhou et al. (2015)	Minimax conditionl entropy principle	Multiple	Independent
Platanios et al. (2016)	• Hierarchical models • Gibbs sampling	Binary	Independent or dependent
Northcutt et al. (2021a)	• Assume previously-trained classifier • Estimate joint dist. $p(z, y)$ on new data set • Correct or remove mislabeled samples	Multiple	Not applicable

Table 2: Related work on noisy-label modeling and learning. References that jointly learn a noisy-label model and train a predictive model appear in Table 3.

Branson et al. (2017) extended the models by Welinder and Perona and by Welinder et al. to include other forms of annotation. They also introduced an algorithm that sequentially acquires more noisy labels until the risk of the estimated correct label falls below a threshold. **Van Horn et al. (2018)** extended the work by Branson et al. to multi-class classification and sequentially dependent labelers.

Karger et al. (2014) considered the problem of allocating labelers to unlabeled samples and proposed a message-passing algorithm for estimating the correct labels. **Zhou et al. (2015)** included sample difficulty in their model of $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ and estimated $\boldsymbol{\psi}$, $\boldsymbol{\pi}$, and $\boldsymbol{Y}$ by optimizing a minimax criterion on the conditional entropy of $\boldsymbol{Z}$ given Y. **Platanios et al. (2016)** proposed a variety of generative models for $p(\boldsymbol{z}|y;\boldsymbol{\psi})$, and they applied Gibbs sampling to infer $\boldsymbol{\pi}$, $\boldsymbol{\psi}$, and $\boldsymbol{Y}$.

None of these authors considered training or testing, and they used the known correct labels to compute estimation errors in their experiments. For our purposes, these works offer noisy-label models that could be learned—even without correct labels—and used with our training and testing methods. The model introduced by Whitehill et al. (2009) is

italicized because, in Section 5.1.2, we present a similar model for $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ that includes sample difficulty and labeler fallibility; however, our purpose is not to estimate $\boldsymbol{\psi}$ but to demonstrate training and testing when $\boldsymbol{\psi}$ is already available.

Northcutt et al. (2021a) took a different viewpoint and assumed the availability of an existing classifier, previously trained on an auxiliary data set with enough correctly-labeled samples to overcome the presence of some noisily-labeled ones. Given a feature vector, the existing classifier predicts class probabilities for *all* classes, unlike a human labeler who provides a noisy label indicating a single class. For a new data set with noisy labels, Northcutt et al. leveraged this property to estimate the *joint* distribution $p(z, y)$ as well as $p(z|y)$ and $\boldsymbol{\pi}$. They used these estimates to identify samples in the new data set that were likely mislabeled and to correct or remove them. A new classifier can then be trained or tested using the corrected labels. This approach offers a different way to obtain a noisy-label model, and our methods can complement it by addressing samples whose noisy labels cannot be resolved.

Some other authors perform joint learning of a noisy-label model and training of a predictive model. These references are discussed next.

1.2.2 Training (and Joint Learning) with Truthing Issues

Works on training with truthing issues are listed in Table 3 and reviewed here.

Cid-Sueiro (2012) and **Cid-Sueiro et al. (2014)** took a Bayesian viewpoint and examined weak losses for training with partial labels, which are modeled slightly differently than the noisy labels considered here. The authors related the weak loss to an equivalent loss for correct labels and studied theoretical aspects of constructing a weak loss from a given equivalent loss. Their approach does not conform to the unified view of training that we offer in Section 3.2; it could be interpreted as the reverse of the minimum mean-square error training method proposed in Section 3.4.1. The two approaches are discussed in Section 3.6.1.

Natarajan et al. (2013, 2018) took a classical (i.e., frequentist) view and trained binary classifiers on noisy labels from a single labeler by forming a proxy loss function for $\boldsymbol{Z}$ that is an unbiased estimator (in the classical sense) of the ideal loss function for y. Their approach does not fit into our unified view; further discussion appears in Section 3.6.2. **van Rooyen and Williamson (2018)** presented an abstract framework for learning with noisy labels, parts of which generalize the work by Cid-Sueiro (2012), Cid-Sueiro et al. (2014), and Natarajan et al. (2013, 2018).

Ratner et al. (2016, 2017) proposed a technique for programmatically generating multiple noisy labels for many unlabeled samples and subsequently training discriminative classifiers with the noisy labels. To model $p(\boldsymbol{z}|y;\boldsymbol{\psi})$, they introduced generative models with independent or dependent labelers, and they estimated $\boldsymbol{\psi}$ while assuming an equiprobable class prior. Training minimized the expected empirical risk with respect to $p(\boldsymbol{z}|y;\boldsymbol{\psi})$.

The remaining authors considered joint learning and training with noisy labels. **Raykar et al. (2010)** used the EM algorithm to jointly estimate $\boldsymbol{\psi}$, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}$ and train a logistic regression binary classifier. They also presented a Bayesian form of the algorithm with priors on the labelers' error probabilities and provided approximate posteriors of the error prob-

abilities, and they extended the approach to multi-class classification, ordinal regression, and regression.

Khetan et al. (2018) presented a version of the EM algorithm that alternates between training a binary classifier with $\underline{\boldsymbol{z}}$ and the current estimates of $\boldsymbol{\psi}$ and $\boldsymbol{\pi}$ and then updating the estimates of $\boldsymbol{\psi}$ and $\boldsymbol{\pi}$ using the current predictions. They weight the training loss function using the correct-label posterior for each possible correct-label value, effectively marginalizing out the correct-label RV.

To train a *convolutional neural network* (CNN) on noisy labels from a single labeler and estimate $p(z|y)$, **Sukhbaatar et al. (2015)** and **Jindal et al. (2016)** appended an additional layer to the softmax output of a base network. During training, the base network learns to predict the unknown correct labels while the additional layer estimates $p(z|y)$ and predicts the noisy label from the output of the base network. Following training, the additional layer can be excised and the base network used for prediction. This training method lies outside our unified view of training and is discussed in Section 3.6.3.

Tanno et al. (2019) jointly trained a CNN and estimated the confusion matrices for multiple independent labelers. They used the confusion matrices to convert the CNN outputs into predictions of the noisy labels, which is similar to the work by Sukhbaatar et al. (2015). Notably, Tanno et al. introduced a trace-regularization term, which minimizes the traces of the confusion matrices and, under certain conditions, ensures proper estimation. The authors compared their method to several other methods, including those of Raykar et al. (2010), Khetan et al. (2018), and Sukhbaatar et al. (2015). This approach does not fit our unified view of training, and further discussion appears in Section 3.6.4

All of these authors used the correct labels during testing or other experimental evaluations; they did not consider testing with truthing issues. As Table 3 shows, many of them do not consider multiple, possibly dependent labelers, which our work allows. Some of them jointly learn a noisy-label model and a predictive model, which we do not consider. A number of these works fit within our unified view of training (Section 3). The table italicizes parts of Ratner et al. (2016, 2017) and Khetan et al. (2018) that are consistent with our use of minimum mean-square error estimation of the empirical risk (Section 3.4.1). In the work of Raykar et al. (2010), logistic regression is italicized because we also use it as an illustrative example (Section 5.4).

1.2.3 Testing with Truthing Issues

As noted above, all of the related work on training used a reserved set of correct labels for testing. Related work on testing with truthing issues appears in Table 4.

Smyth et al. (1994) considered testing with noisy labels assigned by a scientist or classifier to synthetic aperture imagery of the planet Venus to indicate the presence or absence of types of volcanoes. Clearly, absolute ground truth is unavailable in this case. To test an individual scientist's predicted labels, the authors used the EM algorithm proposed by Dawid and Skene (1979) with the noisy labels from all other scientists to estimate the correct labels, which were then treated as if correct. A classifier was also trained using consensus labels from two scientists (Burl et al., 1994). It was tested by comparing its predictions to the estimated correct labels from EM using all scientists' noisy labels.

Reference	Description	Number of Classes	Labeler Dependence	Joint with noisy-label model learning?	Fits into our unified view?
Cid-Sueiro (2012); Cid-Sueiro et al. (2014)	• Theoretical Bayesian view • Weak loss designed for partial labels • Equivalent loss for correct labels	Binary or multiple	Not applicable (Partial labels)	No	No
Natarajan et al. (2013, 2018)	• Classical or frequentist view • Proxy loss function that is unbiased in classical sense	Binary	Not applicable (Single labeler)	No	No
van Rooyen and Williamson (2018)	• Abstract framework • Generalization of Cid-Sueiro (2012); Cid-Sueiro et al. (2014); Natarajan et al. (2013, 2018)	Binary	Not applicable (Single labeler)	No	No
Ratner et al. (2016, 2017)	• Logistic regression • Also long short-term memory network • Programmatic noisy labeling • *Minimization of expected empirical risk*	Binary	Independent or dependent	No	Yes
Raykar et al. (2010)	• *Binary logistic regression* as main example • Also regression, ordinal regression • EM or Bayesian estimation	Binary or multiple	Independent	Yes	Yes
Khetan et al. (2018)	• Empirical risk minimization • *Marginalization of loss function using correct-label posterior*	Binary or multiple	Independent	Yes	Yes
Sukhbaatar et al. (2015); Jindal et al. (2016)	• CNN • Base network learns $p(y\|\boldsymbol{x})$ • Extra layer learns $p(z\|y)$	Multiple	Not applicable (Single labeler)	Yes	No
Tanno et al. (2019)	• CNN • Regularization of trace of labelers' confusion matrices	Multiple	Independent	Yes	No

Table 3: Related work on training with truthing issues.

Reference	Description	Number of Classes	Labeler Dependence
Smyth et al. (1994)	• Comparison of one labeler's noisy labels with estimated correct labels obtained by applying EM to other labelers' noisy labels • Multi-class labels (volcano type or non-volcano) reduced to binary labels (volcano or non-volcano)	Multiple, reduced to binary	Independent
Lam and Stork (2003)	• Effect of noisy labels on probability of error • Variance of estimated probability of error as a function of labeler error probability, number of samples	Multiple, reduced to binary	Not applicable (Single labeler)
Carlotto (2009)	• Study of the effect of noisy labels on accuracy • Relationship between accuracies calculated on correct labels and on noisy labels • Rough estimate of ideal accuracy	Multiple	Not applicable (Single labeler)
Holodnak et al. (2018)	Empirical study of methods for estimating accuracy from noisy labels	Multiple	Independent or dependent

Table 4: Related work on testing with truthing issues.

For binary classification, **Lam and Stork (2003)** related the ideal probability of error $\Pr(\hat{y} \neq y) = 1 - \textit{accuracy}$ to a labeler's error probability $\varepsilon = \Pr(z \neq y)$ and the classifier's apparent probability of error $\Pr(\hat{y} \neq z)$. They provided an estimate of $\Pr(\hat{y} \neq y)$ given an assumed value of ε, and they examined the variance of this estimate as a function of ε and the number of samples N.

Carlotto (2009) analyzed how measured accuracy is affected by truth errors for a single labeler. Carlotto obtained an expression that relates ideal accuracy to accuracy calculated against noisy labels and suggested a rough estimate of accuracy when the labeler's error probability is known.

Holodnak et al. (2018) conducted an empirical study with real and simulated data to compare a variety of techniques for estimating the accuracy of a classifier from noisy labels. They introduced two models that incorporate dependencies between the labelers or noisy-label RVs, and they demonstrated that estimation techniques that assume conditional independence provide less reliable estimates as labeler dependence increases.

In a survey paper on classification with label noise (Frénay and Verleysen, 2014, p. 862), the authors remark, "a problem that is seldom mentioned in the literature is that model validation can be difficult in the presence of label noise." Indeed, the number of references on testing is considerably smaller than that on learning or training, and such work has mainly examined accuracy. Nevertheless, the above works reflect the main approaches: Calculate estimated correct labels $\check{\boldsymbol{y}}$ and use them as the reference; estimate the ideal accuracy in some way; and comparative studies. Our work includes many common testing metrics, including accuracy, precision, recall or probability of detection, and probability of false alarm. We focus on estimating the testing metrics rather than the correct labels, and we develop algorithms for computing Bayesian optimal estimates of scalar and joint metrics. Our experiments use conditionally independent labelers for implementational convenience, but our approaches accommodate noisy-label models with conditionally dependent labelers.

1.2.4 Comparing Combinations of Labelers

Regarding the comparison of different combinations of labelers, we have found one rather distantly-related publication.

For binary classification and a single labeler, **Lugosi (1992)** viewed the correct-label RV and noisy-label RV as the respective input and output of a communications channel. Lugosi examined purely theoretical aspects of the effects of noisy labels on accuracy if a classifier uses the maximum *a posteriori* or nearest-neighbor decision rule.

In Section 4, we make the channel analogy for one or more labelers, but we proceed in a different direction. We suggest that mutual information can be used to compare combinations of labelers, which implies that the information conveyed by multiple mediocre labelers can equal or exceed that provided by a single expert labeler.

1.3 Supervised Classification and Estimation Theory

This work applies *estimation theory* to supervised classification with truthing issues, and Figure 1 presents conceptual diagrams for these two fields. The diagrams look similar but differ in a fundamental way: *In supervised classification, the actual process is unknown, while in estimation theory, the actual variable is unknown.* Supervised classification is concerned with finding a good predictive model that generalizes to future, out-of-sample realizations from an unknown process, given a set of labeled samples from the process and a family of models. In this work, estimation theory is concerned with making a good guess at the current, in-sample value of an unobserved variable, given noisy measurements and a model for the measurement process.

These fields also use similar terms and objectives. Supervised classification learns model parameters $\boldsymbol{\theta}$, estimation theory finds an estimate $\hat{y}$ or estimator $\hat{Y}$, and both fields seek an answer that is best according to some criterion. We present a few examples. In these examples, the criterion is essentially the same; the differences lie in the components that are assumed to be known, the solution space, and the optimization methods.

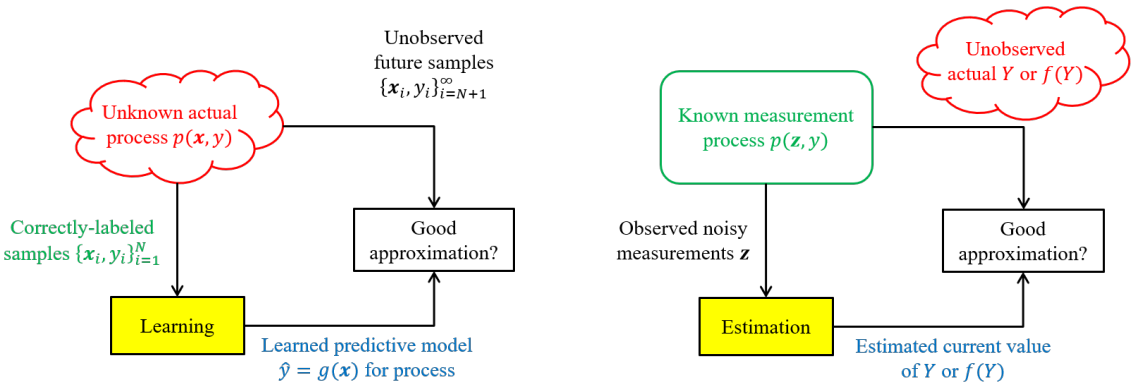
First, both fields employ ML estimation: classification selects non-random parameters $\boldsymbol{\theta}$ to maximize $p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta})$ or $p(\boldsymbol{y},\underline{\boldsymbol{x}};\boldsymbol{\theta})$, and estimation chooses $\hat{y}$ to maximize $p_{\boldsymbol{Z}|Y}(\boldsymbol{z}|\hat{y})$.

Second, when classification treats the model parameters as RVs $\boldsymbol{\Theta}$ with prior $p(\boldsymbol{\theta})$, it uses the maximum *a posteriori* (MAP) criterion and chooses $\boldsymbol{\theta}$ to maximize $p(\boldsymbol{\theta}|\boldsymbol{y},\underline{\boldsymbol{x}})$. When estimation adopts the MAP criterion, it selects $\hat{y} = \arg\max_y p(y|\boldsymbol{z})$.

Third, classification may minimize the average square loss $N^{-1}\sum_{i=1}^{N}(y_i - g(\boldsymbol{x}_i;\boldsymbol{\theta}))^2$, while estimation may minimize the *mean-square error* (MSE) $\mathrm{E}[(Y - h(\boldsymbol{Z}))^2]$. We make repeated use of *minimum mean-square error* (MMSE) estimation, which seeks the estimator $h(\boldsymbol{Z})$ that minimizes the MSE. A convenient standard result (see Appendix C) is that the MMSE estimator is the conditional mean of Y given $\boldsymbol{Z}$:

$$h^{\mathrm{MMSE}}(\boldsymbol{Z}) = \arg\min_h \mathrm{E}[(Y - h(\boldsymbol{Z}))^2] = E[Y|\boldsymbol{Z}]. \tag{1}$$

Finally, classification may minimize the average zero-one loss $N^{-1}\sum_{i=1}^{N}\mathbb{1}(y_i \neq g(\boldsymbol{x}_i;\boldsymbol{\theta}))$, where $\mathbb{1}(w)$ is the *indicator function*: $\mathbb{1}(w) = 0$ if w is false, and $\mathbb{1}(w) = 1$ if w is true. Likewise, estimation may adopt the *minimum probability-of-error* (MPE) criterion and minimize $\mathrm{E}[\mathbb{1}(Y \neq h(\boldsymbol{Z}))]$, the probability of error. Another standard result from estimation



(a) Supervised classification (b) Estimation theory

Figure 1: Comparison of supervised classification and estimation theory. In supervised classification, the goal is to use a set of correctly-labeled samples from an unknown actual process to learn a predictive model that generalizes to future, out-of-sample realizations from the process. In estimation theory, the goal is to estimate the in-sample value of an unobserved variable from noisy measurements produced by a known measurement process. For each problem, the key unknown element appears in red, the key known element in green, and the desired element in blue.

theory (see Appendix D) is that the MPE estimator corresponds to the MAP estimator:

$$h^{\text{MPE}}(\boldsymbol{Z}) = \arg\min_{h} \text{E}[\mathbb{1}(Y \neq h(\boldsymbol{Z}))] = \arg\max_{y} p(y|\boldsymbol{Z}) = h^{\text{MAP}}(\boldsymbol{Z}). \tag{2}$$

Our application of estimation theory begins with the assumption that a good noisy-label model $p(\boldsymbol{z}|y;\boldsymbol{\psi})\pi(y)$ is available. If an auxiliary data set $\{\underline{\boldsymbol{z}}', \boldsymbol{y}'\}$ is available, then such a model could be obtained via supervised learning. Much of the related work addresses the problem of learning a model when correct labels are unavailable but some knowledge about the labelers' error behavior and/or class prior is available. Consequently, this paper is complementary to and broadly compatible with much of the related work.

1.4 Novel Contributions and Organization

We use boldface to highlight specific novel contributions. We discuss some conceptual contributions and then use the three questions raised in Section 1.1 to cover the organization of the rest of the paper and mention other contributions. For reference, Tables 5, 6, and 7 list abbreviations, symbols, and important distributions used throughout this paper.

Supervised classification with truthing issues involves three different components, which each require a model. The truthing or labeling process requires a noisy-label model, training learns the predictive model, and testing calls for a testing model. Much of the related work has investigated ways to learn a noisy-label model and/or use a noisy-label model to train a predictive model.

The greatest conceptual contribution of this paper is its **application of Bayesian estimation theory to training and testing**. To our knowledge, this paper is the first one

to take this viewpoint and pursue its possibilities so extensively. We concentrate on training and testing, and we start with the assumption that a noisy-label model $p(\boldsymbol{z}|y, \boldsymbol{\psi})\pi(y)$ is available, so this work complements much of the related work. The Bayesian approach naturally allows for multiple labelers, different combinations of labelers for each sample, and missing and/or conflicting noisy labels. It also supports noisy-label models with conditionally dependent labelers.[3]

In this work, we say that an estimator is *optimal* only if it meets three requirements: it employs an appropriate estimand, it fully exploits all available information, and it minimizes a well-defined penalty criterion (or maximizes a well-defined utility criterion). An estimator that fails any of these requirements is considered *suboptimal*.

Our technical contributions consist of testing and training approaches that are optimal in this strict sense. Our optimal testing approach focuses on estimating the metrics, introduces a testing model that enables thorough exploitation of the available information, and employs the MMSE criterion. In contrast, some suboptimal methods omit estimation theory entirely, others estimate the correct labels instead of the metrics, and still others omit a testing model and fail to take full advantage of the available information.

Our optimal training methods select an appropriate likelihood function, posterior, or risk that is faithful to the original objective from ideal training; they use the noisy-label model to exploit the available information completely; and they apply the ML, MAP, or MMSE criterion. Some suboptimal methods ignore estimation theory, while others estimate the correct labels rather than the risk.

Our proposed methods never estimate the correct labels because, in our view, they are not the right estimand. This viewpoint marks another novel conceptual contribution: Our conscious choice to **refrain from estimating the correct labels**. We deliberately avoid making hard decisions about the unobserved correct labels because doing so would produce errors that would propagate into training and testing. In this way, our work differs from existing, suboptimal approaches that estimate the correct labels and then proceed as if the estimated labels were correct.

1. *How can one test a classifier in the presence of truthing issues?*

   This question is deeply investigated in Section 2. *For binary classification*, we propose a **testing model** for the noisy and predicted labels (Section 2.1, Figure 2). We then derive the estimation-theoretic testing methods as follows:

   (a) We express various metrics in terms of two common RVs that are independent and approximately normally distributed (Section 2.2).

   (b) We derive approximate **marginal posteriors for accuracy, precision, recall or probability of detection, probability of false alarm, and F-score** and the approximate **joint posteriors for probability of detection and probability of false alarm** as well as for **precision and recall** (Section 2.3).

   (c) We propose ***MMSE testing*** and develop **empirical Bayes algorithms for estimating the testing-model parameters** via iterative MMSE estimation

3. For ease of implementation, our simulations and experiments employ conditionally independent labelers, but the derivations and algorithms do not.

(Algorithms 1 and 2, Figure 3), and we discuss their relation to the EM algorithm and convergence (Section 2.4),

(d) We explain how to calculate Bayesian optimal estimates (MMSE, MAP, and credible regions) of the metrics from the estimated testing-model parameters and posteriors (Section 2.5).

We also describe some alternative testing approaches (Section 2.6), such as MPE or MAP estimation of the correct labels (Section 2.6.1, Algorithm 3) and fully Bayesian estimation (Section 2.6.2).

For multi-class classification (Section 2.7), we **extend the testing model** (Section 2.7.1), provide another **empirical Bayes algorithm for MMSE testing** (Algorithm 4, Figure 4), and derive approximate **posteriors for accuracy and individual elements of the confusion matrix** (Section 2.7.2).

2. *How can one train a classifier in the presence of truthing issues?*

   We consider this question in Section 3. We restate the assumption of independent samples (Section 3.1), and we present a **unified view of training** that encompasses and organizes some of the related work (Section 3.2).

   For probabilistic (e.g., discriminative or generative) models, we derive the **likelihood function or posterior of the predictive model parameters for truthing issues** such that the original, ideal training principle (ML or MAP) is preserved (Section 3.3).

   For non-probabilistic models (Section 3.4), which are trained by minimizing the empirical risk, we propose ***MMSE training*, which minimizes the MMSE estimate of the empirical risk**, and we demonstrate that this approach leads to the same training objective proposed by some related work (Section 3.4.1). We review properties associated with MMSE estimation (Section 3.4.2), mention its convenient form for **gradient calculation** (Section 3.4.3), and consider some special cases (Section 3.4.4). We provide a basic **condition for consistency of the MMSE estimator** (Section 3.4.5).

   Next, we mention some aspects of MMSE training that make it more appealing than ML and MAP training (Section 3.5). We also discuss some alternative training approaches that do not fit into the unified view (Section 3.6). Finally, we describe ways to do training with infrastructure that was not designed for truthing issues (Section 3.7).

3. *How can one compare different combinations of labelers with different abilities?*

   This question is briefly studied in Section 4. We make a simple analogy between the noisy-label model and a communications channel, which suggests mutual information as a basis for comparing combinations of labelers. We focus on the binary symmetric broadcast channel (Section 4.1), and we suggest expressing the mutual information for a set of labelers in terms of that for a **single equivalent labeler** (Section 4.2). We observe that, in theory, **multiple mediocre labelers can be as informative as—or more informative than—a single expert labeler** (Section 4.3).

Experimental results appear in Section 5. We relied on simulation to generate many of the correct, noisy, and predicted labels (Section 5.1). We simulated correct and predicted

Abbreviation	Expansion
BSBC	binary symmetric broadcast channel
BSC	binary symmetric channel
CLT	central limit theorem
CNN	convolutional neural network
EM	expectation-maximization
ERM	empirical risk minimization
MAC	moment-approximation condition
MAD	maximum absolute difference
MAP	maximum *a posteriori*
ML	maximum likelihood
MMSE	minimum mean-square error
MPE	minimum probability-of-error
MSE	mean-square error
OP	operating point
P-R	precision-recall
ROC	receiver operating characteristic
RV	random variable

Table 5: List of abbreviations.

labels in a straightforward manner (Section 5.1.1), and we employed a particular noisy-label model (Section 5.1.2) that is similar to the one by Whitehill et al. (2009). For testing, we review results for several experiments on binary classification (Section 5.2), and we report on one experiment on multi-class classification (Section 5.3). For training, we provide an example involving binary logistic regression (Section 5.4). For the comparison of labelers, we show experiments that use the proposed training and testing methods to verify the possibility of equivalent mutual information (Section 5.5).

Section 6 provides a *summary and conclusions* (Section 6.1), a *workflow* for supervised classification with truthing issues (Section 6.2), and suggested *future directions* (Section 6.3). Several *appendices* are also included. Appendix A derives testing metrics in terms of the common RVs from Section 2.2. Appendix B summarizes results on ratios of jointly normal RVs; they are useful for calculating posteriors of scalar metrics. Appendices C and D review MMSE, MPE, and MAP estimation. Appendix E provides derivations for the training approaches in Sections 3.3 and 3.4. Appendix F gives details for the logistic regression training example in Section 5.4.

2. Testing with Truthing Issues: Grading with Dirty Answer Keys

We cover testing before training because the results and methods presented here apply regardless of how a classifier was trained. They can be employed even if training did not consider truthing issues. Additionally, if one needs to train a model with truthing issues, then one very likely needs to test the trained model with truthing issues, too. One might be reluctant to embark on training if one suspects that truthing issues will invalidate the testing metrics. This section provides reassurance that reliable testing is possible.

Conventional testing calculates metrics over an ideal testing set $\{\hat{\boldsymbol{y}}, \boldsymbol{y}\}$ to measure the (dis)agreement between the predicted labels $\hat{\boldsymbol{y}}$ and the correct labels $\boldsymbol{y}$. In machine learning,

Symbol	Description
B	Bernoulli distribution
Beta	beta distribution
C	number of classes
$\mathrm{C}^{\mathrm{emp}}$, $[\boldsymbol{C}]$	confusion matrix
Cat	categorical distribution
$\mathrm{F}_{\beta}^{\mathrm{emp}}$, $[F_\beta]$	F-score
I	mutual information
J_{ideal}	ideal training objective
J_{pri}	primary term
J_{reg}	regularization term
$\mathbf{K}^{\mathrm{emp}}$, $[\boldsymbol{K}]$	cond. conf. matrix
$\tilde{\mathbf{K}}$	cond. conf. matrix param.
L	loss function
$\hat{L}^{\mathrm{MMSE}}$	MMSE est. of loss function
M	no. draws in sampling algs.
N	number of samples
$\hat{N}_1$	number of times $\hat{y}_i = 1$
$\mathrm{R}^{\mathrm{emp}}$, $[R]$	empirical risk
$\hat{R}^{\mathrm{MMSE}}$	MMSE est. of emp.-risk RV
T	number of labelers
$[U, V]$	common RVs for metrics
$\mathcal{N}$	normal distribution
$\mathcal{T}$	set of labelers
$\mathcal{U}$	uniform distribution
$\mathcal{X}$	feature-vector space
$\mathcal{Y}$	set of classes
f	arbitrary function
g	predictive model

Symbol	Description
$\tilde{g}$	pre-threshold pred. model
h	arbitrary estimator
h^{MMSE}	MMSE estimator
i	sample index
j	iteration or generic index
p_{D}, $[P_{\mathrm{D}}]$	probabilty of detection
p_{FA}, $[P_{\mathrm{FA}}]$	probability of false alarm
$\tilde{p}_{\mathrm{D}}$, $\tilde{p}_{\mathrm{FA}}$, $[\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}}]$	OP parameters
s	pre-threshold pred. stat.
t	labeler index
x, $\boldsymbol{x}$, $\underline{\boldsymbol{x}}$, $[X, \boldsymbol{X}, \underline{\boldsymbol{X}}]$	feature vectors
y, $\boldsymbol{y}$, $[Y, \boldsymbol{Y}]$	correct labels
$\hat{y}$, $\hat{\boldsymbol{y}}$, $[\hat{Y}, \hat{\boldsymbol{Y}}]$	predicted labels
$\check{y}$, $\check{\boldsymbol{y}}$	estimated correct labels
z, $\boldsymbol{z}$, $\underline{\boldsymbol{z}}$, $[Z, \boldsymbol{Z}, \underline{\boldsymbol{Z}}]$	noisy labels
α	clipping value
β	F-score parameter
δ, $\boldsymbol{\delta}$	sample difficulty
ε	error prob. (57) or (60)
η	prob. provide noisy label
θ, $\boldsymbol{\theta}$, $\boldsymbol{\theta}^*$, $[\boldsymbol{\Theta}]$	pred.-model parameters
λ	regularization weight
π, $\boldsymbol{\pi}$	class prior
τ	decision threshold
ϕ, $\boldsymbol{\phi}$	labeler fallibility
ψ, $\boldsymbol{\psi}$, $\underline{\boldsymbol{\psi}}$	noisy-label params.
$\mathbf{0}$	zero vector
$\mathbb{1}(\cdot)$	indicator function

Table 6: List of main symbols. Brackets indicate RVs.

Distribution	Description
$p(\boldsymbol{z}\|y;\boldsymbol{\psi})$	noisy-label conditional distribution
$p(\boldsymbol{z}\|y;\boldsymbol{\psi})\pi(y)$	noisy-label model
$p(\hat{y}_i\|y_i;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})$	predicted-label conditional distribution (3), (4), (5), (6)
$p(\hat{y}_i\|y_i;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})p(\boldsymbol{z}_i\|y_i;\boldsymbol{\psi}_i)$	testing model (7)
$p(y_i\|\hat{y}_i,\boldsymbol{z}_i;\boldsymbol{\psi}_i,\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})$	testing class posterior (10)
$p(y_i\|\boldsymbol{z}_i;\boldsymbol{\psi}_i)$	training class posterior (41)

Table 7: List of key distributions.

$\hat{\boldsymbol{y}}$ is obtained by applying the learned model g to each feature vector $\boldsymbol{x}_i \in \underline{\boldsymbol{x}}$. However, testing does not actually involve the feature vectors $\underline{\boldsymbol{x}}$, so the results of this section apply to problems outside of machine learning, such as reconciling observations by multiple clinicians (for examples, see Dawid and Skene, 1979; Raykar et al., 2010).

With truthing issues, testing must work with the testing set $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$ instead of $\{\hat{\boldsymbol{y}}, \boldsymbol{y}\}$. The metrics are functions of the correct-label RVs, which are unobserved, so *the metrics are RVs whose values remain uncertain.* We therefore treat testing as an estimation problem: we want to know the in-sample values of the metrics on the testing set. We seek the posteriors of the metric RVs, conditioned on $\hat{\boldsymbol{y}}$ and $\underline{\boldsymbol{z}}$. Optimal estimates or credible regions for the metric RVs can then be calculated from the posteriors.

Our approach exploits all available information, including the predicted labels $\hat{\boldsymbol{y}}$. The problem is analogous to grading a multiple-choice quiz using answer keys from one or more teaching assistants who have poor handwriting. The answer keys provide information about the correct answers, but the student's answers do, too. We can obtain the best estimate of the grade by consulting the student's answers along with the answer keys, rather than relying on the answer keys alone.

Most of this section covers binary classification; Section 2.7 discusses extensions to multi-class classification. For binary classification, we consider several common scalar metrics: accuracy, precision, recall or probability of detection,[4] probability of false alarm,[5] and F-score. Each of these metrics takes on values in $[0, 1]$. For probability of false alarm, smaller values indicate better performance; for the other metrics, larger values correspond to better performance. Table 8 gives the empirical forms for these metrics, denoted as *acc*, *prec*, p_{D} or *rec*, p_{FA}, and $\mathrm{F}_\beta^{\mathrm{emp}}$, respectively. We also consider two common joint metrics: the *receiver operating characteristic* (ROC) and *precision-recall* (P-R) *operating points*, namely $(p_{\mathrm{D}}, p_{\mathrm{FA}})$ and $(prec, rec)$. The empirical metrics are used in the ideal case when $\{\hat{\boldsymbol{y}}, \boldsymbol{y}\}$ is available. The corresponding RV forms are ACC, $PREC$, P_{D} or REC, P_{FA}, F_β, as well as $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ and $(PREC, REC)$. We overload the lowercase symbols to mean either an empirical metric or a realization of a metric RV.

2.1 Testing Assumptions

In ideal testing, the metrics do not involve the feature vectors, so we eliminate $\underline{\boldsymbol{X}}$ and $\underline{\boldsymbol{x}}$ from consideration. If we had a good *testing model* for $p(\hat{y}, \boldsymbol{z}|y)$, then we could use it to estimate the metric RVs. One might propose learning it from an auxiliary set $\{\hat{\boldsymbol{y}}', \boldsymbol{y}', \underline{\boldsymbol{z}}'\}$, but if such a set were available, then one could just do ideal testing with $\{\hat{\boldsymbol{y}}', \boldsymbol{y}'\}$.[6]

Instead, we must devise a model for $p(\hat{y}, \boldsymbol{z}|y)$ and estimate its parameters from $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$, with no prospect of learning them from auxiliary data. We tackle this problem by applying techniques from estimation theory rather than machine learning.

We state our assumptions here, and Figure 2 shows the graphical model depicting them. We assume that $\hat{\boldsymbol{y}}$, $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{\psi}}$, and $\boldsymbol{\pi}$ are available. We let $\hat{N}_1$ denote the number of times that $\hat{y}_i = 1$; it is immediately available from $\hat{\boldsymbol{y}}$. We further assume that the noisy-label RVs $\boldsymbol{Z}_i$

4. Recall, probability of detection, sensitivity, and true positive rate are equivalent terms.
5. Probability of false alarm is equivalent to (1 − specificity) and false positive rate.
6. We have chosen to omit any dependence on the feature vectors. If one were to propose learning a testing model such as $p(\hat{y}, \boldsymbol{z}|\boldsymbol{x}, y)p(\boldsymbol{x}, y)$ from $\{\underline{\boldsymbol{x}}', \hat{\boldsymbol{y}}', \boldsymbol{y}', \underline{\boldsymbol{z}}'\}$, then a similar contradiction would arise.

Metric	Empirical Form		Random-Variable Form	
	Symbol	Expression	Symbol	Expression
Accuracy	acc	$\frac{\text{no. of times } \hat{y}_i = y_i}{N}$	ACC	$U - V - \frac{\hat{N}_1}{N} + 1$
Precision	$prec$	$\frac{\text{no. of times } \hat{y}_i = 1 \text{ and } y_i = 1}{\text{no. of times } \hat{y}_i = 1}$	$PREC$	$\frac{N}{\hat{N}_1} U$
Prob. of Detection, Recall	p_{D}, rec	$\frac{\text{no. of times } \hat{y}_i = 1 \text{ and } y_i = 1}{\text{no. of times } y_i = 1}$	P_{D}, REC	$\frac{U}{U+V}$
Prob. of False Alarm	p_{FA}	$\frac{\text{no. of times } \hat{y}_i = 1 \text{ and } y_i = 0}{\text{no. of times } y_i = 0}$	P_{FA}	$\frac{\hat{N}_1/N - U}{1-(U+V)}$
F-Score ($\beta > 0$)	$\mathrm{F}_\beta^{\mathrm{emp}}$	$(1+\beta^2)\frac{prec \cdot rec}{\beta^2 prec + rec}$	F_β	$\frac{(1+\beta^2)U}{\beta^2(U+V) + \hat{N}_1/N}$

Table 8: Binary-classification metrics: Empirical forms and RV forms in terms of the common RVs U and V from (8) and (9).

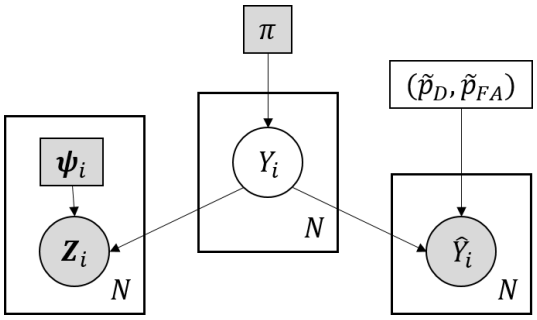


Figure 2: Graphical model for testing approach. Small rectangles indicate non-random variables; circles indicate RVs. Shading indicates a variable that is fully observed. Large rectangles indicate N independent instances of the enclosed variables indexed by i.

have conditional distribution $p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)$ and do not depend on the predicted label. As is common practice, we also assume independent samples, so $p(\underline{\boldsymbol{z}}|\boldsymbol{y};\underline{\boldsymbol{\psi}}) = \prod_{i=1}^{N} p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)$.

Next, we observe that accuracy, precision, and F-score can each be written solely in terms of the class prior, probability of detection, and probability of false alarm. For example, $acc = \pi(0)(1-p_{\mathrm{FA}}) + \pi(1)p_{\mathrm{D}}$, and $prec = \pi(1)p_{\mathrm{D}}/(\pi(0)p_{\mathrm{FA}} + \pi(1)p_{\mathrm{D}})$. Consequently, we introduce the ROC *operating-point (OP) parameters* $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, which suffice to determine the other metrics because the class prior is assumed known. The OP parameters represent the anticipated performance on the testing set before $\hat{\boldsymbol{Y}}$ and $\underline{\boldsymbol{Z}}$ are observed.

We then treat each predicted label $\hat{y}_i$ as a realization of a *predicted-label RV* $\hat{Y}_i$ conditioned on the correct-label RV Y_i and $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. The *predicted-label conditional distribution*

is simply

$$p_{\hat{Y}_i|Y_i}(0|0;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = 1-\tilde{p}_{\mathrm{FA}}, \tag{3}$$

$$p_{\hat{Y}_i|Y_i}(1|0;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = \tilde{p}_{\mathrm{FA}}, \tag{4}$$

$$p_{\hat{Y}_i|Y_i}(0|1;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = 1-\tilde{p}_{\mathrm{D}}, \tag{5}$$

$$p_{\hat{Y}_i|Y_i}(1|1;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = \tilde{p}_{\mathrm{D}}. \tag{6}$$

The assumption of independent samples means $p(\hat{\boldsymbol{y}}|\boldsymbol{y};\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = \prod_{i=1}^{N} p(\hat{y}_i|y_i;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})$.

Our final assumption is that, given Y_i, the RVs $\hat{Y}_i$ and $\boldsymbol{Z}_i$ are conditionally independent, where $(\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})$ and $\boldsymbol{\psi}_i$ only influence $\hat{Y}_i$ and $\boldsymbol{Z}_i$, respectively. This assumption may be a strong one, but it reflects the typical case in which $\hat{Y}_i$ is generated without access to $\boldsymbol{Z}_i$. For example, $\hat{Y}_i$ could be the output of a predictive model whose only input is $\boldsymbol{X}_i$, or $\hat{Y}_i$ could be a diagnosis made by a clinician who has not seen the diagnoses from other clinicians.

Therefore, our *testing model* is

$$p(\hat{y}_i,\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i,\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}) = p(\hat{y}_i|y_i;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i). \tag{7}$$

We excluded $\boldsymbol{X}_i$, so Y_i is the only RV that can link $\boldsymbol{Z}_i$ and $\hat{Y}_i$, and the model includes such a connection. The model is parsimonious, having just two parameters. As explained at the top of this section, they will not be determined via supervised learning. Section 2.4 presents iterative methods for estimating them from $\{\hat{\boldsymbol{y}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi}\}$.

2.2 Metric RVs in Terms of Common RVs

Here we demonstrate that each metric RV can be written in terms of two RVs that are independent and approximately normal.

2.2.1 Metric RVs in Terms of Common RVs

We begin by considering a single metric, namely probability of detection. Given truthing issues, it becomes an RV conditioned on $\hat{\boldsymbol{y}}$, $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{\psi}}$, and $(\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})$, which we write as

$$P_{\mathrm{D}} = \frac{\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(\hat{y}_i=1 \text{ and } Y_i=1)}{\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(Y_i=1)}.$$

In this expression, the predicted label is shown in lowercase because it is observed, and the correct label is capitalized because it is an unobserved RV.

We define the RVs U and V, conditioned on $\{\hat{\boldsymbol{y}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}}\}$, as follows:

$$\begin{aligned} U &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(\hat{y}_i=1 \text{ and } Y_i=1) \\ &= \frac{1}{N}\sum_{i:\hat{y}_i=1}\mathbb{1}(Y_i=1), \end{aligned} \tag{8}$$

and

$$V = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = 0 \text{ and } Y_i = 1)$$
$$= \frac{1}{N}\sum_{i:\hat{y}_i=0} \mathbb{1}(Y_i = 1). \tag{9}$$

The numerator of P_{D} is exactly U. The denominator is

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = 1) = \frac{1}{N}\sum_{i:\hat{y}_i=1} \mathbb{1}(Y_i = 1) + \frac{1}{N}\sum_{i:\hat{y}_i=0} \mathbb{1}(Y_i = 1)$$
$$= U + V.$$

Thus,

$$P_{\mathrm{D}} = \frac{U}{U+V}.$$

By similar manipulations (see Appendix A), we can write each of the other metric RVs in terms of the *common RVs* U and V. Table 8 summarizes the relationships.

2.2.2 Independence and Approximate Normality

We immediately conclude that U and V are independent since they involve summations over disjoint subsets of $\hat{\boldsymbol{y}}$. We now consider their distributions.

In (8) and (9), each $\mathbb{1}(Y_i = 1)$ is a Bernoulli RV with success probability equal to $p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. Using Bayes' rule, we obtain this probability from the *testing class posterior*:

$$p(y_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) = \frac{\pi(y_i)p(\hat{y}_i, \boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})}{\sum_{y_i' \in \mathcal{Y}} \pi(y_i')p(\hat{y}_i, \boldsymbol{z}_i|y_i'; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})}$$
$$= \frac{\pi(y_i)p(\hat{y}_i|y_i; \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i)}{\sum_{y_i' \in \mathcal{Y}} \pi(y_i')p(\hat{y}_i|y_i'; \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})p(\boldsymbol{z}_i|y_i'; \boldsymbol{\psi}_i)}, \tag{10}$$

where the appropriate value for $p(\hat{y}_i|y_i; \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ can be determined from (3)–(6).

We denote a Bernoulli RV with success probability $p \in [0, 1]$ by $\mathrm{B}(p)$, so each $\mathbb{1}(Y_i = 1)$ in (8) is distributed $\mathrm{B}(p_i)$ with

$$p_i = p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|1, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}), \quad i = 1, \ldots, N. \tag{11}$$

Let $\mathcal{N}(\mu, \sigma^2)$ denote a normal RV with mean μ and variance σ^2. By the *central limit theorem* (CLT), U is approximately distributed $\mathcal{N}(\mu_U, \sigma_U^2 | \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ with

$$\begin{aligned}
\mu_U(\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) &= \frac{1}{N} \sum_{i:\hat{y}_i=1} p_i \\
&= \frac{1}{N} \sum_{i:\hat{y}_i=1} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|1, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}), \qquad (12) \\
\sigma_U^2(\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) &= \frac{1}{N^2} \sum_{i:\hat{y}_i=1} p_i(1-p_i) \\
&= \frac{1}{N^2} \sum_{i:\hat{y}_i=1} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|1, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \\
&\qquad \cdot \big(1 - p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|1, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})\big), \qquad (13)
\end{aligned}$$

where we have indicated the dependence on $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\}$.

Likewise, each $\mathbb{1}(Y_i = 1)$ in (9) is distributed $\mathrm{B}(q_i)$ with

$$q_i = p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|0, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}), \quad i = 1, \ldots, N, \qquad (14)$$

so V is approximately distributed $\mathcal{N}(\mu_V, \sigma_V^2 | \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ with

$$\begin{aligned}
\mu_V(\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) &= \frac{1}{N} \sum_{i:\hat{y}_i=0} q_i \\
&= \frac{1}{N} \sum_{i:\hat{y}_i=0} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|0, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}), \qquad (15) \\
\sigma_V^2(\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) &= \frac{1}{N^2} \sum_{i:\hat{y}_i=0} q_i(1-q_i) \\
&= \frac{1}{N^2} \sum_{i:\hat{y}_i=0} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|0, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \\
&\qquad \cdot \big(1 - p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(1|0, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})\big). \qquad (16)
\end{aligned}$$

These expressions readily accommodate some correctly-labeled samples. For the i^{th} sample, if y_i is known or can be exactly recovered from $\boldsymbol{z}_i$, then $p_i = \mathbb{1}(y_i = 1)$ if $\hat{y}_i = 1$, and $q_i = \mathbb{1}(y_i = 1)$ if $\hat{y}_i = 0$. Correctly-labeled samples add a constant to the summations for μ_U and μ_V and contribute zero to the summations for σ_U^2 and σ_V^2.

2.3 Posteriors of Metric RVs

We can now obtain the posteriors of the metric RVs.

2.3.1 Scalar Metric RVs

We immediately find that ACC is approximately normal with mean $\mu_U - \mu_V - \hat{N}_1/N + 1$ and variance $\sigma_U^2 + \sigma_V^2$. Similarly, $PREC$ is approximately normal with mean $N\mu_U/\hat{N}_1$ and variance $N^2\sigma_U^2/\hat{N}_1^2$.

From the "Random-Variable Form" section of Table 8, we observe that each remaining scalar metric RV is a ratio of jointly, approximately normal RVs. Closed-form expressions for these posteriors are unavailable, but the work of Marsaglia (1965, 2006) explains how the posteriors can be computed. Appendix B reviews the procedure, and Table 16 in the appendix covers the necessary parameters for each of these metrics.

Moreover, Marsaglia (2006, §4) and Appendix B provide closed-form approximations for the mean and variance of a ratio of jointly normal RVs. The approximations are valid if a simple *moment-approximation condition* (MAC) is satisfied.[7]

In summary, for any scalar metric RV $Q \in \{ACC, PREC, P_{\mathrm{D}}, P_{\mathrm{FA}}, F_\beta\}$, we can compute the moments μ_U, σ_U^2, μ_V, σ_V^2 from $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\}$ and use them to get approximations for $p(q|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, $\mathrm{E}[Q|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}]$, and $\mathrm{var}(Q|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$.

2.3.2 Joint Metric RVs

For ROC analysis, we observe that P_{D} and P_{FA} are functions of two independent, approximately normal RVs U and V. We apply the standard transformation for two functions of two RVs (see Papoulis, 1991, §6-3) to obtain the joint posterior of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$:

$$p(p_{\mathrm{D}}, p_{\mathrm{FA}}|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \approx \left| \frac{(\hat{N}_1/N - p_{\mathrm{FA}})(\hat{N}_1/N - p_{\mathrm{D}})}{(p_{\mathrm{D}} - p_{\mathrm{FA}})^3} \right| \frac{1}{2\pi\sigma_U\sigma_V}$$
$$\cdot \exp\left[-\frac{1}{2} \left(\frac{\left(\frac{\hat{N}_1/N - p_{\mathrm{FA}}}{p_{\mathrm{D}} - p_{\mathrm{FA}}} p_{\mathrm{D}} - \mu_U\right)^2}{\sigma_U^2} + \frac{\left(\frac{\hat{N}_1/N - p_{\mathrm{FA}}}{p_{\mathrm{D}} - p_{\mathrm{FA}}} (1 - p_{\mathrm{D}}) - \mu_V\right)^2}{\sigma_V^2} \right) \right],$$
$$0 \le p_{\mathrm{D}} \le 1, 0 \le p_{\mathrm{FA}} \le 1, p_{\mathrm{D}} \ne p_{\mathrm{FA}}, \quad (17)$$

where the moments of U and V are computed from (11)–(16).

Clearly, the posterior is non-Gaussian. This expression is well-defined except when $p_{\mathrm{D}} = p_{\mathrm{FA}}$, which is the *chance line* in ROC space (see Saito and Rehmsmeier, 2015). Although we have not succeeded in finding a closed-form expression for $\lim_{p_{\mathrm{D}} \to p_{\mathrm{FA}}} p(p_{\mathrm{D}}, p_{\mathrm{FA}}|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, we do not consider the lack of this limit to be problematic. First, Section 2.3.3 explains how we can use sampling to approximate the posterior of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ without using (17). Second, a binary classifier is only useful if it operates far from the chance line, so one will likely only be interested in classifiers with negligible probability density near the chance line. Finally, in Section 2.4.1, we present an estimation algorithm that only uses the marginal posteriors of P_{D} and P_{FA} and does not require (17).

7. In general, the distribution of a ratio of jointly normal RVs cannot be approximated well by a normal distribution. However, Marsaglia (2006) and the appendix also provide a test for when a normal approximation is reasonable.

For P-R analysis, $PREC$ and REC are also functions of U and V. We apply the transformation of functions of RVs to obtain the joint posterior of $(PREC, REC)$:

$$p(prec, rec|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \approx \left(\frac{\hat{N}_1}{N}\right)^2 \frac{prec}{rec^2} \frac{1}{2\pi\sigma_U\sigma_V} \cdot \exp\left[-\frac{1}{2}\left(\frac{\left(\frac{\hat{N}_1}{N} prec - \mu_U\right)^2}{\sigma_U^2} + \frac{\left(\frac{\hat{N}_1}{N} prec \left(\frac{1-rec}{rec}\right) - \mu_V\right)^2}{\sigma_V^2}\right)\right], \quad 0 < rec \leq 1, 0 \leq prec \leq 1, \quad (18)$$

with the moments of U and V computed from (11)–(16). This distribution is also non-Gaussian. The chance line in P-R space is $prec = \pi(1)$ (see Saito and Rehmsmeier, 2015), and it poses no difficulties. L'Hôpital's rule can be applied to show that $\lim_{rec\to 0} p(prec, rec|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) = 0$, so this expression is well-defined over the entire unit square.

2.3.3 Posteriors via Sampling

We can also use sampling to approximate the posteriors. We generate M length-N realizations $\{\boldsymbol{y}^{(m)}\}_{m=1}^{M}$ of the correct-label RVs $\boldsymbol{Y}$. For each realization $\boldsymbol{y}^{(m)}$, each $y_i^{(m)}$ is drawn $\mathrm{B}\big(p_{Y|\hat{Y}_i,\boldsymbol{Z}_i}(1|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})\big)$ using (10). For each $\boldsymbol{y}^{(m)}$, we compute the desired empirical metric (scalar or joint), which yields M realizations of the metric RV. The approximate posterior can then be computed as a one-dimensional or two-dimensional histogram.

2.4 Empirical Bayes Estimation of Operating-Point Parameters (MMSE Testing)

The validity of the posteriors depends on how well we can estimate the moments of U and V, which in turn depend upon the OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. We present two iterative algorithms for finding the MMSE estimate of the OP parameters, so we refer to this approach as *MMSE testing*. The algorithms belong to the class of *empirical Bayes* estimators. Standard Bayesian estimation handles unknown nuisance variables by specifying a prior for them and marginalizing them out to obtain the posterior estimate of the estimand. In contrast, empirical Bayes estimation does not assume a prior;[8] it instead consults the available data to estimate nuisance variables (Casella, 1992), often using MAP or ML estimation. It alternates between estimating the nuisance variables and the estimand. This iterative process successively improves each estimate and is similar in spirit to the EM algorithm of Dempster et al. (1977). In the proposed algorithms, the OP parameters are the nuisance variables, the estimand consists of the probability-of-detection and probability-of-false alarm RVs, and MMSE estimation is employed. Section 2.4.4 explores the relationship between these algorithms and the EM algorithm.

2.4.1 Motivation

Suppose that, on the j^{th} iteration, we have the previous OP parameters $\big(\tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$, along with $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. The preceding results enable us to get the moments of U and

8. As a result, empirical Bayes methods are sometimes said not to be "fully Bayesian."

V conditioned on $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\}$ and then to get the approximate posterior of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$. We can improve on the OP parameters by updating $\big(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\big)$ with an optimal estimate of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ given $\big(\tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ and $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. We can repeat this procedure until the OP parameters converge.

We adopt the MMSE criterion from Bayesian estimation theory. Let $\boldsymbol{h}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) = (h_{\mathrm{D}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}), h_{\mathrm{FA}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}))$ be an *estimator* of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ given $\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}$. The MSE of such an estimator is $\mathrm{E}\big[\big(h_{\mathrm{D}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - P_{\mathrm{D}}\big)^2\big] + \mathrm{E}\big[\big(h_{\mathrm{FA}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - P_{\mathrm{FA}}\big)^2\big]$, and the MMSE estimator is defined as $\boldsymbol{h}^{\mathrm{MMSE}} = \arg\min_{\boldsymbol{h}} \mathrm{E}\big[\big(h_{\mathrm{D}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - P_{\mathrm{D}}\big)^2\big] + \mathrm{E}\big[\big(h_{\mathrm{FA}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - P_{\mathrm{FA}}\big)^2\big]$. The standard result is that the MMSE estimator is the conditional mean (see (1) or Appendix C), so

$$\boldsymbol{h}^{\mathrm{MMSE}}(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) = \mathrm{E}_{p(p_{\mathrm{D}}, p_{\mathrm{FA}} | \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)})}\big[P_{\mathrm{D}}, P_{\mathrm{FA}} \,\big|\, \hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big],$$

where the relevant distribution is shown as a subscript (cf. Cover and Thomas (1991, Equation 2.2)). Therefore, given $\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}}$ and $\underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}$, we set

$$\begin{aligned}
\tilde{p}_{\mathrm{D}}^{(j)} &= \mathrm{E}\big[P_{\mathrm{D}} \big|\, \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big], \\
\tilde{p}_{\mathrm{FA}}^{(j)} &= \mathrm{E}\big[P_{\mathrm{FA}} \big|\, \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big].
\end{aligned}$$

Each conditional mean involves a *marginal* posterior. For example,

$$\mathrm{E}\big[P_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big] = \int_0^1 p_{\mathrm{D}}\, p\big(p_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)\, dp_{\mathrm{D}}. \tag{19}$$

Thus, as mentioned previously, the lack of a limit along the chance line in (17) does not prevent us from calculating the conditional mean of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$.

We can compute the conditional mean of P_{D} or P_{FA} in three ways. First, if the MAC is satisfied, we can use the closed-form expression in Marsaglia (2006) or Appendix B. Second, we can perform one-dimensional numerical integration. Third, we can use sampling, as described in Section 2.3.3.

2.4.2 Choice of Initial Operating-Point Parameters

This section discusses the initial OP parameters $\big(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\big)$. We take a Bayesian viewpoint with initial OP RVs $\big(\tilde{P}_{\mathrm{D}}^{(0)}, \tilde{P}_{\mathrm{FA}}^{(0)}\big)$ and a non-informative prior—namely, the uniform distribution over the unit square. Then the initial OP parameters $\big(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\big)$ are given by the conditional mean, which is just $(1/2, 1/2)$.

Consequently, $p(\hat{y}_i | y_i; \tilde{p}_{\mathrm{D}} = 1/2, \tilde{p}_{\mathrm{FA}} = 1/2)$ equals $1/2$ for any combination of $\hat{y}_i$ and y_i, so the testing class posterior (10) reduces to

$$p(y_i | \hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}} = 1/2, \tilde{p}_{\mathrm{FA}} = 1/2) = \frac{\pi(y_i) p(\boldsymbol{z}_i | y_i; \boldsymbol{\psi}_i)}{\sum_{y_i' \in \mathcal{Y}} \pi(y_i') p(\boldsymbol{z}_i | y_i'; \boldsymbol{\psi}_i)}. \tag{20}$$

The right-hand side matches the ordinary class posterior, which does not condition on $\hat{y}_i$, $\tilde{p}_{\mathrm{D}}$, and $\tilde{p}_{\mathrm{FA}}$:

$$p(y_i | \boldsymbol{z}_i; \boldsymbol{\psi}_i) = \frac{\pi(y_i) p(\boldsymbol{z}_i | y_i; \boldsymbol{\psi}_i)}{\sum_{y_i' \in \mathcal{Y}} \pi(y_i') p(\boldsymbol{z}_i | y_i'; \boldsymbol{\psi}_i)}. \tag{21}$$

Thus, choosing $\big(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\big) = (1/2, 1/2)$ has the same effect as if we had not included the predicted label at all.

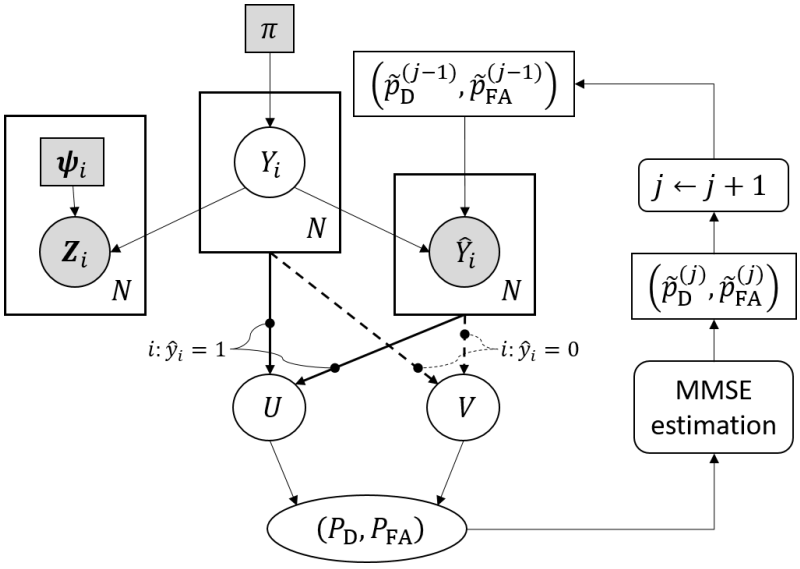


Figure 3: Graphical model of iterative estimation for testing. The common RVs U and V depend on separate partitions of $\boldsymbol{Y}$ and $\hat{\boldsymbol{y}}$.

2.4.3 Estimation Algorithms

The recursive relationship between successive OP parameters leads to two iterative empirical Bayes estimators for MMSE testing, namely Algorithms 1 and 2. Figure 3 depicts the graphical model and the procedure. On each iteration, each algorithm updates its OP parameters by computing the MMSE estimate—the conditional mean—of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ given $\hat{\boldsymbol{y}}$, $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{\psi}}$, and the previous OP parameters. On the next iteration, the MMSE estimate provides the new OP parameters.

Algorithm 1 takes advantage of the fact that P_{D} and P_{FA} are both ratios of jointly approximately normal RVs, so their conditional means can be computed without sampling. It begins the j^{th} iteration by using $\hat{y}_i$, $\boldsymbol{z}_i$, $\boldsymbol{\psi}_i$, $\tilde{p}_{\mathrm{D}}^{(j-1)}$, and $\tilde{p}_{\mathrm{FA}}^{(j-1)}$ to compute p_i and q_i for $i = 1, \ldots, N$, and then it computes the moments of U and V. To get the improved OP parameters $\left(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\right)$, it uses the closed-form approximation for the conditional mean of P_{D} or P_{FA} if the MAC holds, and it falls back to numerical integration if not. To prevent numerical degeneracy, the new OP parameters are clipped to $[\alpha, 1-\alpha]$. The procedure repeats until the maximum absolute difference between successive estimates falls below some tolerance or a maximum number of iterations $j_{\max}$ is reached. It then returns the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) = \left(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\right)$.

Algorithm 2 uses sample realizations of the correct-label RVs $\boldsymbol{Y}$ to approximate the conditional means of P_{D} and P_{FA}. On the j^{th} iteration, it generates M length-N realizations $\boldsymbol{y}^{(m)}$, $m = 1, \ldots, M$, by drawing each $y_i^{(m)} \sim \mathrm{B}\left(p_{Y|\hat{Y}_i, \boldsymbol{Z}_i}(1|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)})\right)$ using (10). Then it computes the empirical ROC operating point $\left(\tilde{p}_{\mathrm{D}}^{(j,m)}, \tilde{p}_{\mathrm{FA}}^{(j,m)}\right)$ for each realization. The conditional mean of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ is obtained by averaging the M operating points, which yields the new OP parameters $\left(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\right)$.

Algorithm 1 MMSE testing with empirical Bayes estimation of $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ via ratios of jointly normal RVs.

function EmpiricalBayesViaRatios($\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}$)
  Initialize $\tilde{p}_{\mathrm{D}}^{(0)} \leftarrow 0.5, \tilde{p}_{\mathrm{FA}}^{(0)} \leftarrow 0.5, j \leftarrow 0, MAD \leftarrow \infty$
  while $j < j_{\max}$ and $MAD \geq tol$ **do**
    $j \leftarrow j + 1$
    for $i \leftarrow 1 : N$ **do**
      Compute p_i and q_i from $\hat{y}_i, \boldsymbol{z}_i, \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}$, and $\tilde{p}_{\mathrm{FA}}^{(j-1)}$ ▷ Eqs. (10), (11), (14)
    Use $\{p_i\}_{i=1}^N$ and $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\}$ to compute μ_U, σ_U^2 ▷ Eqs. (12), (13)
    Use $\{q_i\}_{i=1}^N$ and $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\}$ to compute μ_V, σ_V^2 ▷ Eqs. (15), (16)
    if mean approximation for P_{D} from $\mu_U, \sigma_U^2, \mu_V, \sigma_V^2$ is valid **then** ▷ Marsaglia (2006), App. B
      $\tilde{p}_{\mathrm{D}}^{(j)} \leftarrow \mathrm{E}\big[P_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big]$ from mean approximation
    else
      Get $p\big(p_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ from $\mu_U, \sigma_U^2, \mu_V, \sigma_V^2$ ▷ Marsaglia (1965, 2006), App. B
      $\tilde{p}_{\mathrm{D}}^{(j)} \leftarrow \mathrm{E}\big[P_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big]$ by 1-D numerical integration
    if mean approximation for P_{FA} from $\mu_U, \sigma_U^2, \mu_V, \sigma_V^2$ is valid **then**
      $\tilde{p}_{\mathrm{FA}}^{(j)} \leftarrow \mathrm{E}\big[P_{\mathrm{FA}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big]$ from mean approximation
    else
      Get $p\big(p_{\mathrm{FA}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ from $\mu_U, \sigma_U^2, \mu_V, \sigma_V^2$
      $\tilde{p}_{\mathrm{FA}}^{(j)} \leftarrow \mathrm{E}\big[P_{\mathrm{FA}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big]$ by 1-D numerical integration
    Clip $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$ to $[\alpha, 1-\alpha]$
    $MAD \leftarrow \max\big\{\big|\tilde{p}_{\mathrm{D}}^{(j)} - \tilde{p}_{\mathrm{D}}^{(j-1)}\big|, \big|\tilde{p}_{\mathrm{FA}}^{(j)} - \tilde{p}_{\mathrm{FA}}^{(j-1)}\big|\big\}$
  $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \leftarrow \big(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\big)$ ▷ Final OP parameters
  return $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$

Both algorithms used $tol = 10^{-3}$, $j_{\max} = 30$, $\alpha = 10^{-3}$, and Algorithm 2 used $M = 5000$. They also allow for some correctly-labeled samples. If y_i is known or can be recovered exactly from $\boldsymbol{z}_i$, then p_i, q_i, or $p_{Y|\hat{Y}_i,\boldsymbol{Z}_i}(1|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)})$ equals $\mathbb{1}(y_i = 1)$. Algorithm 1 will add the proper constants to the moments of U and V, and Algorithm 2 will always draw the proper realization of the correct-label RV.

2.4.4 Relation to EM Algorithm and Remarks on Convergence

Like many empirical Bayes methods, our iterative algorithms are similar to the EM algorithm but differ in that the hidden or latent variables—namely, the correct labels—are RVs instead of non-random quantities. The above relationships let us obtain the approximate conditional means of P_{D} and P_{FA} directly, so our "E-step" does not employ an auxiliary function like the EM algorithm does. Also, the other latent variables—namely the OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$—are determined using MMSE rather than ML estimation, so our "M-step" performs minimization of the MSE rather than maximization of an auxiliary function.

Algorithm 2 MMSE testing with empirical Bayes estimation of $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ via sampling.

function EmpiricalBayesViaSampling($\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}, M$)
  Initialize $\tilde{p}_{\mathrm{D}}^{(0)} \leftarrow 0.5, \tilde{p}_{\mathrm{FA}}^{(0)} \leftarrow 0.5, j \leftarrow 0, MAD \leftarrow \infty$
  while $j < j_{\max}$ and $MAD \geq tol$ **do**
    $j \leftarrow j + 1$
    for $m \leftarrow 1 : M$ **do** ▷ Generate M length-N realizations of $\boldsymbol{Y}$
      for $i \leftarrow 1 : N$ **do** ▷ Generate m^{th} realization $\boldsymbol{y}^{(m)}$ of $\boldsymbol{Y}$
        Draw $y_i^{(m)} \sim \mathrm{B}\big(p_{Y|\hat{Y}_i,\boldsymbol{Z}_i}(1|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)})\big)$ ▷ Eq. (10)
      $\big(\tilde{p}_{\mathrm{D}}^{(j,m)}, \tilde{p}_{\mathrm{FA}}^{(j,m)}\big) \leftarrow$ empirical metrics for $\{\hat{\boldsymbol{y}}, \boldsymbol{y}^{(m)}\}$
    $\big(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\big) \leftarrow$ mean of $\big\{\big(\tilde{p}_{\mathrm{D}}^{(j,m)}, \tilde{p}_{\mathrm{FA}}^{(j,m)}\big)\big\}_{m=1}^{M}$ ▷ Empirical conditional mean
    Clip $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$ to $[\alpha, 1-\alpha]$
    $MAD \leftarrow \max\big\{\big|\tilde{p}_{\mathrm{D}}^{(j)} - \tilde{p}_{\mathrm{D}}^{(j-1)}\big|, \big|\tilde{p}_{\mathrm{FA}}^{(j)} - \tilde{p}_{\mathrm{FA}}^{(j-1)}\big|\big\}$
  $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \leftarrow \big(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\big)$ ▷ Final OP parameters
  return $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$

We speculate that, regardless of the initial OP parameters, the algorithms should converge to the global optimum $(\tilde{p}_{\mathrm{D}}^{*}, \tilde{p}_{\mathrm{FA}}^{*})$, the minimum of the MSE $\mathrm{E}\big[\big(P_{\mathrm{D}} - \tilde{p}_{\mathrm{D}}\big)^2 + \big(P_{\mathrm{FA}} - \tilde{p}_{\mathrm{FA}}\big)^2 \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\big]$. The argument is based on the following reasoning: The estimand $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ is bounded; generally, the conditional mean of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ is unique for any OP parameters $\big(\tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$; and the MSE decreases on each iteration, unless $\big(\tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ is already at the global optimum. Therefore, the algorithms should converge to the global optimum regardless of $\big(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\big)$. This property contrasts with the EM algorithm, which can only be said to converge to a local optimum, and this local optimum may vary significantly depending on initialization (Koller and Friedman, 2009, §19.2). We can also view the algorithms as seeking the unique fixed point $\big(\tilde{p}_{\mathrm{D}}^{*}, \tilde{p}_{\mathrm{FA}}^{*}\big)$ such that $\mathrm{E}[P_{\mathrm{D}}, P_{\mathrm{FA}} | \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}^{*}, \tilde{p}_{\mathrm{FA}}^{*}] = \big(\tilde{p}_{\mathrm{D}}^{*}, \tilde{p}_{\mathrm{FA}}^{*}\big)$.

A technicality prevents us from making a stronger statement regarding convergence. Both P_{D} and P_{FA} are approximated as ratios of jointly normal RVs, for which the mean and variance do not exist (see Marsaglia (1965, 2006) and Appendix B). As a result, uniqueness of the MMSE estimator of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ cannot be guaranteed, so a strict proof of convergence or existence of a unique fixed point may not be possible.

This obstacle might mainly be a theoretical concern, and, except for a few pathological situations, the algorithms might always converge in practice. Section 5.2.3 presents some empirical evidence of global convergence.

2.5 Optimal Estimation of Metric RVs

Algorithms 1 and 2 each produce final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. It still remains to calculate optimal estimates of the metric RVs given $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\}$. We do so by computing the moments of the common RVs U and V (Section 2.2.2), and then by using the estimated posteriors of the metric RVs (Section 2.3). Given the approximate posteriors, optimal estimation is a simple matter; we briefly discuss it for completeness.

We consider two point estimates and one range estimate. The first point estimate is the conditional mean, which is optimal in the MMSE sense. The accuracy and precision RVs are approximately normal, so it is just the mean. For *REC* or P_{D}, P_{FA}, or F_β, we can use the mean approximation if the MAC holds or resort to numerical integration or sampling. For P-R or ROC analysis, we can just use the conditional means of the individual elements.

The second point estimate is the MAP estimate, the most-probable value of the metric RV. It can be obtained by computing the posterior at a fine resolution and finding the peak.

The optimal range estimate is the p%-credible region, which specifies a region such that, with probability $p/100$, the metric RV lies inside the region. The credible region is not necessarily unique, but one way to obtain a reasonable credible region is to apply binary search to find a threshold c such that the numerical integral of the posterior over the points where the posterior exceeds c is equal to $p/100$ within some tolerance.

Finally, we reiterate that Algorithms 1 and 2 and the subsequent optimal-estimation calculations never attempt to estimate the correct-label RVs $\boldsymbol{Y}$. Making a hard decision about $\boldsymbol{Y}$ at any point could introduce errors into downstream processing, as remarked in Section 1.4. Our approach avoids doing so while using all available information to produce its estimates.[9]

2.6 Alternative Testing Approaches

For comparison purposes, we mention some alternative approaches to testing. We first present four suboptimal methods, and then we describe a fully Bayesian approach, which is optimal but handles the OP parameters differently than the empirical Bayes approach.

2.6.1 Suboptimal Testing Approaches

The first suboptimal approach is to *estimate the correct labels* and then treat the estimates as if they were correct to improve on the OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. We present an iterative method in Algorithm 3 and denote its estimate of the correct-label RVs $\boldsymbol{Y}$ as $\check{\boldsymbol{y}}$. On the j^{th} iteration, the algorithm uses the previous OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) = \big(\tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ and estimates $\boldsymbol{Y}$ according to the MPE criterion. Let $h_i(\hat{Y}_i, \boldsymbol{Z}_i)$ be an estimator of Y_i given $\hat{Y}_i$, $\boldsymbol{Z}_i$, and parameters $\boldsymbol{\psi}_i$, $\tilde{p}_{\mathrm{D}}^{(j-1)}$, $\tilde{p}_{\mathrm{FA}}^{(j-1)}$, which are suppressed because they are not RVs. The probability of error of h_i is $p_{\mathrm{error}}(h_i(\hat{Y}_i, \boldsymbol{Z}_i), Y_i) = \mathrm{E}[\mathbb{1}(h_i(\hat{Y}_i, \boldsymbol{Z}_i) \neq Y_i)] = \sum_{\hat{y}_i, \boldsymbol{z}_i} \sum_y \mathbb{1}(h_i(\hat{y}_i, \boldsymbol{z}_i) \neq y_i) p(\hat{y}_i, \boldsymbol{z}_i, y_i)$. This approach seeks $(h_1^{\mathrm{MPE}}, \ldots, h_N^{\mathrm{MPE}}) = \arg\min_{(h_1,\ldots,h_N)} N^{-1} \sum_{i=1}^N p_{\mathrm{error}}(h_i(\hat{Y}_i, \boldsymbol{Z}_i), Y_i)$.

The solution consists of finding the MPE estimator for each individual Y_i, and the standard result is that the MAP estimator minimizes the probability of error (see (2) or Appendix D). Thus, the algorithm computes $\check{\boldsymbol{y}}^{(j)}$ using:

$$\check{y}_i^{(j)} = \arg\max_{y \in \mathcal{Y}} p\big(y | \hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big), \quad i = 1, \ldots, N. \tag{22}$$

The algorithm then uses these estimates to compute the empirical probabilities of detection and false alarm $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$. This approach is similar to Algorithms 1 and 2 except

9. If one wants to estimate $\boldsymbol{Y}$, then one can do so using (10) after $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ has been determined. However, subsequently using this estimate to compute testing metrics runs counter to our approach.

Algorithm 3 Suboptimal estimation of $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ by estimating the correct-label RVs $\boldsymbol{Y}$.

function EstimateOPParametersViaEstimatedCorrectLabels($\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}$)
  Initialize $\tilde{p}_{\mathrm{D}}^{(0)} \leftarrow 0.5, \tilde{p}_{\mathrm{FA}}^{(0)} \leftarrow 0.5, j \leftarrow 0, MAD \leftarrow \infty$
  while $j < j_{\max}$ and $MAD \geq tol$ **do**
    $j \leftarrow j + 1$
    for $i \leftarrow 1 : N$ **do**
      $\check{y}_i^{(j)} \leftarrow \arg\max_{y \in \mathcal{Y}} p_{Y_i|\hat{Y}_i, \boldsymbol{Z}_i}\big(y|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}} = \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}} = \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ ▷ Eq. (22)
    Compute empirical $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$ from $\hat{\boldsymbol{y}}$ and $\check{\boldsymbol{y}}^{(j)} = \big(\check{y}_i^{(j)}\big)_{i=1}^N$
    Clip $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$ to $[\alpha, 1-\alpha]$
    $MAD \leftarrow \max\big\{|\tilde{p}_{\mathrm{D}}^{(j)} - \tilde{p}_{\mathrm{D}}^{(j-1)}|, |\tilde{p}_{\mathrm{FA}}^{(j)} - \tilde{p}_{\mathrm{FA}}^{(j-1)}|\big\}$
  $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) \leftarrow \big(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)}\big)$ ▷ Final OP parameters
  return $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$

that it makes a hard decision about the correct labels on every iteration. This approach leverages the testing model and minimizes a well-defined penalty criterion, but it is suboptimal according to Section 1.4 because it estimates $\boldsymbol{Y}$ when it should estimate $(P_{\mathrm{D}}, P_{\mathrm{FA}})$. We make no claims about its convergence properties. We again used $tol = 10^{-3}$, $j_{\max} = 30$, $\alpha = 10^{-3}$.

Like Algorithms 1 and 2, Algorithm 3 only estimates the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. Following this step, this approach applies (22) to get the final MPE or MAP estimate $\check{\boldsymbol{y}}$ of $\boldsymbol{Y}$ given $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\}$. Final estimates of the metrics are obtained by treating $\check{\boldsymbol{y}}$ as if it were correct and computing empirical metrics for $\{\hat{\boldsymbol{y}}, \check{\boldsymbol{y}}\}$.

The second suboptimal approach neglects estimation theory completely; it just merges the metrics calculated for each *individual labeler's labels*. For each $t \in \mathcal{T}$, it treats the labels from the t^{th} labeler as if they were correct, and it computes the empirical metric for these samples only. Doing so produces T instances of a metric. The final estimate is then obtained using a centrality statistic, such as the mean or median, of the T instances.

The third and fourth suboptimal approaches use the noisy-label conditional distribution $p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i)$ instead of the testing model (7), so they omit $p(\hat{y}_i|y_i; \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ and $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. They are suboptimal because they neglect the relationship between $\hat{Y}_i$ and Y_i and thus fail to exploit the predicted labels $\hat{\boldsymbol{y}}$ fully. The third approach uses MMSE estimation but replaces $p(y_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ with $p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$ in (11)–(16). The fourth approach estimates the correct labels and makes the same substitution in (22). (Equivalently, one can set $j_{\max} = 0$ in Algorithm 1, 2, or 3, so $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}) = (1/2, 1/2)$ and (10) reduces to (21).)

2.6.2 Fully Bayesian Estimation

Another alternative is a *fully Bayesian* approach, which treats $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ as an unobserved realization of nuisance-parameter RVs $(\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}})$ with prior $p(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. A uniform distribution over the unit square serves as a non-informative prior.

This approach estimates each metric RV by marginalizing out $(\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}})$, so the estimate is not conditioned on a particular instance of $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, and no iteration is required. For example, consider MMSE estimation of the accuracy RV. For an estima-

tor $h(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}})$ of ACC, the MSE is $\mathrm{E}\big[\big(h(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - ACC\big)^2\big]$, and the MMSE estimator is $h^{\mathrm{MMSE}} = \arg\min_h \mathrm{E}\big[\big(h(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}) - ACC\big)^2\big]$. The standard result (see (1) or Appendix C) is that the solution is the conditional mean, so the fully Bayesian estimate of ACC is

$$\mathrm{E}_{p(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})}[ACC \,|\, \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}] = \int_0^1 \int_0^1 \mathrm{E}[ACC \,|\, \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}; \underline{\boldsymbol{\psi}}]\, p(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})\, d\tilde{p}_{\mathrm{D}}\, d\tilde{p}_{\mathrm{FA}}.$$

This estimate minimizes the MSE under the assumption that the OP parameters are RVs $(\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}})$ rather than non-random quantities.

In contrast, the empirical Bayes approach does not view $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ as realizations of RVs; it treats them as unknown, non-random parameters of $p(\hat{y}_i | y_i; \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, so it does not marginalize them out, and its estimates depend on them. For example, it computes $\mathrm{E}\big[P_{\mathrm{D}} \big| \hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\big]$ in (19), where the moments of U and V are conditional on $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}}\}$, which depends on the specific choice of $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$.

Both approaches are optimal according to Section 1.4, but they differ in their treatment of the OP parameters. Marginalization over $(\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}})$ is a form of averaging, so the fully Bayesian approach computes an *average grade*. The empirical Bayes approach is more faithful to the grading analogy at the beginning of this section: The quiz has *a particular grade*, represented by $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$, rather than an average grade. Nevertheless, experimentation is required to compare the performance of these approaches; it appears in Section 5.2.1.

2.7 MMSE Testing for Multi-Class Classification

This section discusses the extension of MMSE testing to multi-class classification ($C > 2$) and some of the challenges associated with it. An immediate challenge is that it might be difficult to model $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ or estimate $\boldsymbol{\psi}$ and $\boldsymbol{\pi}$, but the related work described in Sections 1.2.1 and 1.2.2 is very encouraging.

We begin by introducing the $C \times C$ *conditional confusion matrix*. We use $\mathbf{K}^{\mathrm{emp}}$ to denote its empirical form, and we indicate an element of it by $\mathbf{K}^{\mathrm{emp}}_{n|\ell}$ to emphasize its conditional nature.[10] Then $\mathbf{K}^{\mathrm{emp}}_{n|\ell}$ is defined as

$$\mathbf{K}^{\mathrm{emp}}_{n|\ell} = \frac{\text{no. of times } \hat{y}_i = n \text{ and } y_i = \ell}{\text{no. of times } y_i = \ell}, \quad \ell, n \in \mathcal{Y}. \tag{23}$$

The RV form of the matrix is $\boldsymbol{K}$, with

$$\boldsymbol{K}_{n|\ell} = \frac{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = n \text{ and } Y_i = \ell)}{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = \ell)}, \quad \ell, n \in \mathcal{Y}.$$

2.7.1 Empirical Bayes via Sampling (Multi-Class Classification)

We make the same assumptions as for binary classification in Section 2.1. We are given $\{\underline{\boldsymbol{z}}, \hat{\boldsymbol{y}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$, each $\boldsymbol{Z}_i$ has conditional distribution $p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)$, and the samples are independent. A graphical model appears in Figure 4. Each predicted label $\hat{y}_i$ is a realization of an RV $\hat{Y}_i$, which for multi-class classification has conditional distribution $p(\hat{y}_i|y_i;\tilde{\mathbf{K}})$, where

10. We use non-standard, zero-based, column-major matrix indexing; $\mathbf{K}^{\mathrm{emp}}_{n|\ell}$ corresponds to $\mathbf{K}^{\mathrm{emp}}(\ell+1, n+1)$ in standard matrix notation.

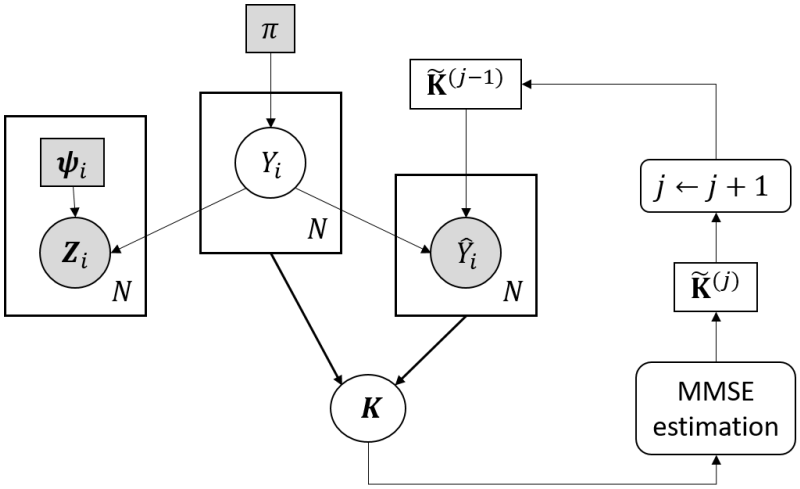


Figure 4: Multi-class classification: Graphical model of iterative estimation for testing.

$\tilde{\mathbf{K}}$ is a matrix of conditional confusion-matrix *parameters* that are analogous to the OP parameters, and therefore

$$p(\hat{y}_i|y_i;\tilde{\mathbf{K}}) = \tilde{\mathbf{K}}_{\hat{y}_i|y_i}. \tag{24}$$

Each row of $\boldsymbol{K}$ must form a $(C-1)$-probability simplex: For each $\ell \in \mathcal{Y}$, $\boldsymbol{K}_{n|\ell} \geq 0$, $n \in \mathcal{Y}$, and $\sum_{n\in\mathcal{Y}} \boldsymbol{K}_{n|\ell} = 1$. When $C = 2$, it was sufficient to consider a single element in each row of $\boldsymbol{K}$, and we used $P_{\mathrm{FA}} = \boldsymbol{K}_{1|0}$ and $P_{\mathrm{D}} = \boldsymbol{K}_{1|1}$; this property enabled us to use the mean approximations or integrate the marginal posteriors of P_{D} and P_{FA} in Algorithm 1. This technique is not viable for $C > 2$ because, for each row of $\boldsymbol{K}$, we would have to determine the joint posterior of $(C-1)$ RVs, and we would have to do $(C-1)$-dimensional numerical integration.

On the other hand, for moderate values of C and a class prior that is not highly skewed, we can readily extend the empirical Bayes sampling procedure of Algorithm 2 to multi-class classification. Pseudocode appears in Algorithm 4. Using the parameter matrix $\tilde{\mathbf{K}}$, we draw M realizations of $\boldsymbol{Y}$ according to the multi-class testing class posterior, which is obtained in the same way as (10):

$$\begin{aligned} p(y_i|\hat{y}_i,\boldsymbol{z}_i;\boldsymbol{\psi}_i,\tilde{\mathbf{K}}) &= \frac{\pi(y_i)p(\hat{y}_i,\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i,\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})}{\sum_{y_i'\in\mathcal{Y}}\pi(y_i')p(\hat{y}_i,\boldsymbol{z}_i|y_i';\boldsymbol{\psi}_i,\tilde{\mathbf{K}})} \\ &= \frac{\pi(y_i)p(\hat{y}_i|y_i;\tilde{\mathbf{K}})p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)}{\sum_{y_i'\in\mathcal{Y}}\pi(y_i')p(\hat{y}_i|y_i';\tilde{\mathbf{K}})p(\boldsymbol{z}_i|y_i';\boldsymbol{\psi}_i)} \\ &\overset{(a)}{=} \frac{\pi(y_i)\,\tilde{\mathbf{K}}_{\hat{y}_i|y_i}\,p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)}{\sum_{y_i'\in\mathcal{Y}}\pi(y_i')\,\tilde{\mathbf{K}}_{\hat{y}_i|y_i'}\,p(\boldsymbol{z}_i|y_i';\boldsymbol{\psi}_i)}, \end{aligned} \tag{25}$$

where (a) is from (24). We compute the empirical conditional confusion matrix for each length-N realization of $\boldsymbol{Y}$, and we average the M matrices to get an estimate of the conditional mean of $\boldsymbol{K}$ given $\{\hat{\boldsymbol{y}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\tilde{\mathbf{K}}\}$. For Algorithm 4, we set $tol = 10^{-3}$, $j_{\max} = 50$, $\alpha = 10^{-3}$, and $M = 2500 \times C$.

Algorithm 4 MMSE testing for multi-class classification with empirical Bayes estimation of $\boldsymbol{K}$ via sampling.

function MultiClassEmpiricalBayesViaSampling($\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}, M$)
  Initialize $\tilde{\mathbf{K}}^{(0)}_{n|\ell} = 1/C$ for $(n, \ell) \in \mathcal{Y} \times \mathcal{Y}$, and $j \leftarrow 0$
  repeat
    $j \leftarrow j + 1$
    for $m \leftarrow 1 : M$ **do**
      for $i \leftarrow 1 : N$ **do**
        Draw $y_i^{(m)} \sim p_{Y|\hat{Y}_i, \boldsymbol{Z}_i}(y|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}}^{(j-1)})$ ▷ Eq. (25)
      $\tilde{\mathbf{K}}^{(j,m)} \leftarrow$ empirical matrix from $\{\hat{\boldsymbol{y}}, \boldsymbol{y}^{(m)}\}$ ▷ Eq. (23)
    $\tilde{\mathbf{K}}^{(j)} \leftarrow$ mean of $\{\tilde{\mathbf{K}}^{(j,m)}\}_{m=1}^{M}$
    Adjust each row of $\tilde{\mathbf{K}}^{(j)}$ so each element lies in $[\alpha, 1-\alpha]$ and row sums to one
  until $\left|\left|\tilde{\mathbf{K}}^{(j)} - \tilde{\mathbf{K}}^{(j-1)}\right|\right|_{\max} < tol$ or $j \geq j_{\max}$
  $\tilde{\mathbf{K}} \leftarrow \tilde{\mathbf{K}}^{(j)}$ ▷ Final estimate of $\boldsymbol{K}$
  return $\tilde{\mathbf{K}}$

This method allows for some correctly-labeled samples. If the correct label for the i^{th} sample is known to be equal to ℓ, then $p_{Y|\hat{Y}_i, \boldsymbol{Z}_i}(y|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}}^{(j-1)}) = \mathbb{1}(y = \ell)$, and sampling will always produce ℓ when drawing a realization for this sample.

The initial parameter matrix $\tilde{\mathbf{K}}^{(0)}$ is the result of using a non-informative prior for each row of the matrix—namely, a Dirichlet distribution of order C with all concentration parameters equal to unity. The algorithm returns $\tilde{\mathbf{K}}$, the approximate conditional mean of $\boldsymbol{K}$. The elements of $\boldsymbol{K}$ are bounded and MMSE estimation is employed, so the remarks in Section 2.4.4 on convergence may also be applicable.

2.7.2 Posteriors of Metric RVs (Multi-Class Classification)

Once the estimate $\tilde{\mathbf{K}}$ is available, we can obtain the posteriors of different metric RVs.

Accuracy is defined in the same way for binary and multi-class classification. Empirical accuracy is $acc =$ (no. of times $\hat{y}_i = y_i)/N$, and its RV form is $ACC = N^{-1} \sum_{i=1}^{N} \mathbb{1}(Y_i = \hat{y}_i)$. In the multi-class case, each $\mathbb{1}(Y_i = \hat{y}_i)$ is a Bernoulli RV with success probability $p_{Y_i|\hat{Y}_i, \boldsymbol{Z}_i}(\hat{y}_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}})$. The Bernoulli RVs are independent, so ACC is approximately normal by the CLT.

In addition, the ordinary $C \times C$ *confusion matrix* has empirical form $\mathbf{C}^{\text{emp}}$ with[11] $\mathbf{C}^{\text{emp}}_{n,\ell} =$ (no. of times $\hat{y}_i = n$ and $y_i = \ell$) and RV form $\boldsymbol{C}$ with

$$\begin{aligned} \boldsymbol{C}_{n,\ell} &= \sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = n \text{ and } Y_i = \ell) \\ &= \sum_{i:\hat{y}_i = n} \mathbb{1}(Y_i = \ell). \end{aligned}$$

11. $\mathbf{C}^{\text{emp}}_{n,\ell}$ corresponds to $\mathbf{C}^{\text{emp}}(\ell + 1, n + 1)$ in standard matrix notation.

Hence, matrix element $\boldsymbol{C}_{n,\ell}$ is approximately normally distributed with $\mathrm{E}[\boldsymbol{C}_{n,\ell}|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{\mathbf{K}}] = \sum_{i:\hat{y}_i=n} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(\hat{y}_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}})$ and $\mathrm{var}(\boldsymbol{C}_{n,\ell}|\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}, \tilde{\mathbf{K}}) = \sum_{i:\hat{y}_i=n} p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(\hat{y}_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}}) \cdot\big(1-p_{Y_i|\hat{Y}_i,\boldsymbol{Z}_i}(\hat{y}_i|\hat{y}_i, \boldsymbol{z}_i; \boldsymbol{\psi}_i, \tilde{\mathbf{K}})\big)$. This result applies to the posterior of each *individual* element of $\boldsymbol{C}$; it does not describe the joint posterior of multiple elements of $\boldsymbol{C}$.

Finally, we remark that one could apply the ratio-of-normals procedure to compute the approximate posterior of $\boldsymbol{K}_{n|\ell}$, an individual element of $\boldsymbol{K}$.

3. Training with Truthing Issues: Learning from Noisy Labels

This section addresses training with truthing issues. Recall that we use $g(\boldsymbol{x};\boldsymbol{\theta})$ to denote a classifier or predictive model with model parameters $\boldsymbol{\theta}$. Given $\boldsymbol{x}$, the classifier calculates a statistic $s = \tilde{g}(\boldsymbol{x};\boldsymbol{\theta})$, which contains the model's calculation of the chance that $\boldsymbol{x}$ belongs to each class, and then it applies a decision rule to s to select $\hat{y}$. In binary classification, s is typically a scalar, and $\hat{y}$ is selected by comparing s to a threshold τ; i.e., $\hat{y} = \mathbb{1}(s > \tau)$. In multi-class classification, s is a usually vector $(s_0, s_1, \ldots, s_{C-1})$, and $\hat{y}$ corresponds to the index of the largest element of s; i.e., $\hat{y} = \arg\max_{y' \in \mathcal{Y}} s_{y'}$.

Training is the process of learning $\boldsymbol{\theta}$ from the training set. In the ideal case, the training set is $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$, and training seeks the parameters $\boldsymbol{\theta}^*$ that will produce the most accurate predictions when the trained model is applied to as-yet-unseen samples. Training is usually posed as an optimization problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ideal}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y}) = \arg\min_{\boldsymbol{\theta}} \big[J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y}) + \lambda J_{\mathrm{reg}}(\boldsymbol{\theta})\big], \tag{26}$$

where the *primary term* $J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y})$ imposes a cost or penalty for differences between the correct labels $y_i \in \boldsymbol{y}$ and either $g(\boldsymbol{x}_i;\boldsymbol{\theta})$ or $\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta})$, $\boldsymbol{x}_i \in \underline{\boldsymbol{x}}$; $J_{\mathrm{reg}}(\boldsymbol{\theta})$ is a *regularization term* that reduces overfitting to the training set and improves generalization to unseen samples; and the weight $\lambda \geq 0$ controls the level of regularization. The primary term depends on the predictive model, and the regularization term depends on the choice of regularization, such as L_2 regularization: $J_{\mathrm{reg}}(\boldsymbol{\theta}) \propto \sum_j \theta_j^2$, or L_1 regularization: $J_{\mathrm{reg}}(\boldsymbol{\theta}) \propto \sum_j |\theta_j|$.

3.1 Training Assumptions

In the presence of truthing issues, the correct labels are not available, and the training set becomes $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. With $\underline{\boldsymbol{\psi}}$ and $\boldsymbol{\pi}$ given, the noisy-label model $p(\boldsymbol{z}|y;\boldsymbol{\psi})\pi(y)$ is known for each sample. We make the usual assumption of independent samples, so $p(\underline{\boldsymbol{z}}|\boldsymbol{y}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}) = \prod_{i=1}^{N} p(\boldsymbol{z}_i|y_i, \boldsymbol{x}_i; \boldsymbol{\psi}_i)$. Next, we assume that, given y_i and $\boldsymbol{\psi}_i$, the noisy-label RVs $\boldsymbol{Z}_i$ do not depend on the feature vector $\boldsymbol{x}_i$, which gives $p(\boldsymbol{z}_i|y_i, \boldsymbol{x}_i; \boldsymbol{\psi}_i) = p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i)$. Thus,

$$p(\underline{\boldsymbol{z}}|\boldsymbol{y}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}) = \prod_{i=1}^{N} p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i). \tag{27}$$

This expression allows for conditionally dependent labelers.

We limit our attention to training that can be expressed as in (26). Then the optimization problem for training given $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$ becomes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}) = \arg\min_{\boldsymbol{\theta}} \big[J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}) + \lambda J_{\mathrm{reg}}(\boldsymbol{\theta})\big], \tag{28}$$

where only the primary term has been modified to become $J_{\text{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi})$. Since the regularization term is unchanged, we focus on the primary term below.

3.2 Unified View

We present a *unified view* of training with truthing issues that describes general approaches for training probabilistic or non-probabilistic predictive models.

Unified View of Training with Truthing Issues

1. *For probabilistic models, keep the optimality principle from ideal training, and modify the primary term to account for* $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$ *rather than* $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$*:*
   (a) *For ML training, which would ideally maximize the likelihood function* $p(\boldsymbol{y}|\underline{\boldsymbol{x}}; \boldsymbol{\theta})$ *or* $p(\boldsymbol{y}, \underline{\boldsymbol{x}}; \boldsymbol{\theta})$*, instead use* $p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}, \boldsymbol{\theta})$ *or* $p(\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}, \boldsymbol{\theta})$*, respectively.*
   (b) *For MAP training, which would ideally maximize the posterior distribution* $p(\boldsymbol{\theta}|\boldsymbol{y}, \underline{\boldsymbol{x}})$*, instead use* $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}})$*.*
2. *For non-probabilistic models that use a loss function and would ideally minimize the empirical risk* $\mathrm{R}^{\text{emp}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y})$*, perform* MMSE training*: retain the loss function and minimize* $\hat{R}^{\text{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}})$*, the MMSE estimate of the empirical-risk RV given* $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$*.*

The unified view is simple, elegant, and intuitively appealing. For probabilistic models, training remains true to the original optimality principle from ideal training. For non-probabilistic models, training retains the original loss function from ideal training and optimizes the MMSE estimate of the empirical risk.

Each approach is optimal according to Section 1.4. The estimands are appropriate: Training for probabilistic models targets the likelihood function or posterior, and training for non-probabilistic models targets the empirical risk. The use of the likelihood function, posterior, or MMSE estimator means that all available information in $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$ is fully exploited. The penalty or utility criterion is clearly defined. None of the methods try to estimate the correct labels.

The unified view also organizes some of the related work. The training method proposed by Raykar et al. (2010) corresponds to Item 1a. Ratner et al. (2016, 2017) and Khetan et al. (2018) proposed training approaches that are equivalent to Item 2, although they did not arrive at them by applying MMSE estimation. We assume that $p(\boldsymbol{z}|y, \boldsymbol{\psi})\pi(y)$ is known or has already been learned, but Raykar et al. (2010) and Khetan et al. (2018) have demonstrated that one can employ these training approaches while jointly learning the noisy-label model.

The next two sections derive the likelihood functions, posteriors, and MMSE estimator that provide the modified primary terms. The derivations are much simpler than those for testing because there are no predicted labels $\hat{\boldsymbol{y}}$ to consider. Indeed, the derivations amount to marginalizing over the correct labels. Marginalization introduces some implicit regularization by accounting for the uncertainty of the correct labels.[12]

12. The author thanks one of the anonymous reviewers for this observation.

3.3 Probabilistic Predictive Models (ML or MAP Training)

Here, we address predictive models based on a probabilistic viewpoint. There are two aspects to consider. One aspect is the form of the predictive model: discriminative or generative. A discriminative model assumes a form for the posterior $p(y|\boldsymbol{x};\boldsymbol{\theta})$ and directly uses $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) = p(y|\boldsymbol{x};\boldsymbol{\theta})$. A generative model assumes a form for the joint distribution $p(y,\boldsymbol{x};\boldsymbol{\theta})$ and uses $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) = p(y,\boldsymbol{x};\boldsymbol{\theta})/p(\boldsymbol{x};\boldsymbol{\theta}) \propto p(y,\boldsymbol{x};\boldsymbol{\theta})$. Many generative models apply the factorization $p(y,\boldsymbol{x};\boldsymbol{\theta}) = p(\boldsymbol{x}|y;\boldsymbol{\theta})p(y;\boldsymbol{\theta})$ and use $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) = p(\boldsymbol{x}|y;\boldsymbol{\theta})p(y;\boldsymbol{\theta})/p(\boldsymbol{x};\boldsymbol{\theta}) \propto p(\boldsymbol{x}|y;\boldsymbol{\theta})p(y;\boldsymbol{\theta})$. The choice of $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta})$ determines $J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y})$. Logistic regression and neural networks are common examples of discriminative models, and naïve Bayes is a classic example of a generative model (see Ng and Jordan, 2001).

The other aspect is the treatment of the predictive-model parameters $\boldsymbol{\theta}$: non-random or random. When the parameters are non-random, ML training is employed. For a discriminative model, this means finding $\boldsymbol{\theta}$ to maximize the likelihood function $p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta})$; for a generative model, this means finding $\boldsymbol{\theta}$ to maximize the likelihood function $p(\boldsymbol{y},\underline{\boldsymbol{x}};\boldsymbol{\theta})$. When the parameters are RVs $\boldsymbol{\Theta}$ with prior $p(\boldsymbol{\theta})$, MAP training is used. This means finding $\boldsymbol{\theta}$ that maximizes the posterior $p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{x})$; the posterior is expanded differently depending upon whether the model is discriminative or generative.

By considering both of these aspects, we can derive the primary term—the likelihood function or posterior—in the training objective function. The regularization term remains unchanged since it does not involve the correct labels.

3.3.1 Example Case: Discriminative Model, Non-Random Parameters

To illustrate the approach, we consider a discriminative model with non-random parameters $\boldsymbol{\theta}$. In the ideal case, we have access to $\{\underline{\boldsymbol{x}},\boldsymbol{y}\}$ and use ML training to find $\boldsymbol{\theta}$ to maximize the likelihood function $p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta})$, which is

$$p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}). \tag{29}$$

Equivalently, we can minimize the normalized negative log-likelihood function, which gives

$$\begin{aligned} J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) &= -\frac{1}{N}\log p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta}) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}). \end{aligned} \tag{30}$$

With truthing issues, we seek $\boldsymbol{\theta}$ to maximize $p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta})$, given by

$$\begin{aligned} p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta}) &= \prod_{i=1}^{N} p(\boldsymbol{z}_i|\boldsymbol{x}_i;\boldsymbol{\psi}_i,\boldsymbol{\theta}) \\ &\overset{(a)}{=} \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(y_i, \boldsymbol{z}_i|\boldsymbol{x}_i;\boldsymbol{\psi}_i,\boldsymbol{\theta}) \\ &= \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i|y_i,\boldsymbol{x}_i;\boldsymbol{\psi}_i,\boldsymbol{\theta}) p(y_i|\boldsymbol{x}_i;\boldsymbol{\psi}_i,\boldsymbol{\theta}) \\ &\overset{(b)}{=} \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i) p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}), \end{aligned} \tag{31}$$

where (a) marginalizes over the correct labels, and (b) is because the noisy-label RVs do not depend on the feature vector $\boldsymbol{x}_i$ or the parameters $\boldsymbol{\theta}$ and because the classifier makes its prediction based only on $\boldsymbol{x}_i$ and $\boldsymbol{\theta}$. We can equivalently minimize

$$\begin{aligned} J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi}) &= -\frac{1}{N} \log p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta}) \\ &= -\frac{1}{N} \sum_{i=1}^{N} \log \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i) p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}). \end{aligned} \tag{32}$$

Raykar et al. (2010) proposed (31) as part of joint estimation of $p(\boldsymbol{z}|y)$ and logistic regression training.

3.3.2 All Cases

The top four rows of Table 9 summarize the results for all possible cases; derivations appear in Appendices E.1, E.2, and E.3. The only difference between the ideal case and truthing issues is that the latter marginalizes over the possible values of the correct-label RV Y_i. The other differences are the same as in the ideal case. Switching from non-random to random parameters $\boldsymbol{\Theta}$ introduces another factor for the parameter prior $p(\boldsymbol{\theta})$ and changes the optimization goal from ML to MAP. Switching from a discriminative model to a generative model changes the function being maximized from one involving $p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta})$ to one involving $p(\boldsymbol{x}_i|y_i;\boldsymbol{\theta})\pi(y_i) = p(\boldsymbol{x}_i, y_i;\boldsymbol{\theta})$.

3.4 Non-Probabilistic Predictive Models (Empirical Risk Minimization)

Some classifiers, like support vector machines, do not have a probabilistic formulation, and sometimes the theoretical origins of a probabilistic model are not the main focus. In such cases, ideal training applies the *empirical risk minimization* (ERM) principle (Vapnik, 1991) and seeks $\boldsymbol{\theta}$ to minimize the *empirical risk* or average loss

$$J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) = \mathrm{R}^{\text{emp}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}), y_i), \tag{33}$$

Pred. Model	Par./ Crit.*	Labels and Training Set	
		Ideal: Use $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$	Noisy: Use $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$
Discriminative	NR/ ML	Likelihood $p(\boldsymbol{y}\|\underline{\boldsymbol{x}}; \boldsymbol{\theta})$ $= \prod_{i=1}^{N} p(y_i\|\boldsymbol{x}_i; \boldsymbol{\theta})$	Likelihood $p(\underline{\boldsymbol{z}}\|\underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}, \boldsymbol{\theta})$ $= \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i\|y_i; \boldsymbol{\psi}_i) p(y_i\|\boldsymbol{x}_i; \boldsymbol{\theta})$
	RV/ MAP	Posterior $p(\boldsymbol{\theta}\|\boldsymbol{y}, \underline{\boldsymbol{x}})$ $\propto p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(y_i\|\boldsymbol{x}_i, \boldsymbol{\theta})$	Posterior $p(\boldsymbol{\theta}\|\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}})$ $\propto p(\boldsymbol{\theta}) \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i\|y_i; \boldsymbol{\psi}_i) p(y_i\|\boldsymbol{x}_i, \boldsymbol{\theta})$
Generative	NR/ ML	Likelihood $p(\boldsymbol{y}, \underline{\boldsymbol{x}}; \boldsymbol{\theta})$ $= \prod_{i=1}^{N} p(\boldsymbol{x}_i\|y_i; \boldsymbol{\theta}) \pi(y_i)$	Likelihood $p(\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}, \boldsymbol{\theta})$ $= \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{x}_i\|y_i; \boldsymbol{\theta}) p(\boldsymbol{z}_i\|y_i; \boldsymbol{\psi}_i) \pi(y_i)$
	RV/ MAP	Posterior $p(\boldsymbol{\theta}\|\boldsymbol{y}, \underline{\boldsymbol{x}})$ $\propto p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\boldsymbol{x}_i\|y_i, \boldsymbol{\theta}) \pi(y_i)$	Posterior $p(\boldsymbol{\theta}\|\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}})$ $\propto p(\boldsymbol{\theta}) \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{x}_i\|y_i, \boldsymbol{\theta}) p(\boldsymbol{z}_i\|y_i; \boldsymbol{\psi}_i) \pi(y_i)$
Non-probabilistic	NR/ ERM	Empirical risk $\mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y})$ $= N^{-1} \sum_{i=1}^{N} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i)$	MMSE estimate of empirical-risk RV $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}})$ $= N^{-1} \sum_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) p(y_i\|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$

*"Par./Crit." indicates the treatment of the predictive-model parameters $\boldsymbol{\theta}$ (NR: non-random, RV: random variables) and the optimality criterion (ML: maximum likelihood, MAP: maximum *a posteriori*, ERM: empirical risk minimization).

For discriminative or generative models, the primary term equals the negative normalized log-likelihood or log-posterior; for example, the top-left entry corresponds to $J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y}) = -N^{-1} \log p(\boldsymbol{y}|\underline{\boldsymbol{x}}; \boldsymbol{\theta})$. For non-probabilistic models, the primary term appears in the bottom row.

Table 9: Comparison of training objective functions for predictive models.

where $L(s, y)$ is a *loss function* that penalizes deviations between $s = \tilde{g}(\boldsymbol{x}; \boldsymbol{\theta})$ and the correct label y. Examples for binary classification include support vector machines, which may be trained with the hinge loss, and logistic regression, which corresponds to a linear model trained with the logistic loss. As an example of multi-class classification, neural network training often uses the output vector from the network's final fully-connected layer for s and applies the cross-entropy loss.

In this section, we assume that $\boldsymbol{\theta}$ is not random, since random parameters would require probabilistic modeling. As before, the regularization term is unaffected by truthing issues.

3.4.1 MMSE Estimation of the Empirical-Risk RV (MMSE Training)

With truthing issues, the correct-label RVs are not observed, and the loss functions and empirical risk are functions of them, so their values are uncertain. Let $L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)$, $i = 1, \ldots, N$, be the i^{th} *loss-function RV*, a function of the correct-label RV Y_i, and from (33), write the *empirical-risk RV* as

$$J_{\text{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y}) = R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y}) = \frac{1}{N} \sum_{i=1}^{N} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i). \tag{34}$$

We reason that, if a good in-sample estimate of the empirical-risk RV can be obtained for the training set $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$, then training that minimizes this estimate should perform well. We therefore focus on estimating the empirical-risk RV.

Uncertainty about the empirical-risk RV arises because the correct-label RVs $\boldsymbol{Y}$ are unobserved. The feature vectors $\underline{\boldsymbol{x}}$ do not contribute to the uncertainty because they are known. Given the noisy-label model form $p(\boldsymbol{z}|y; \boldsymbol{\psi})\pi(y)$, Bayes' rule can provide $p(y|\boldsymbol{z}; \boldsymbol{\psi})$ but not $p(y|\boldsymbol{z}, \boldsymbol{x}; \boldsymbol{\psi})$. For these reasons, when estimating the empirical-risk RV, we treat $\boldsymbol{Y}$ and $\underline{\boldsymbol{Z}}$ as RVs, but not $\underline{\boldsymbol{x}}$. However, in the broader context of learning, the feature vectors may be treated as RVs.

We adopt the MMSE criterion when estimating the empirical-risk RV. Denote an estimator of $R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})$ by $\hat{R}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$, where we omit $\underline{\boldsymbol{\psi}}$ and $\boldsymbol{\pi}$ from $\hat{R}$ for brevity. We set the primary term equal to the MMSE estimator of the empirical-risk RV:

$$J_{\text{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}) = \hat{R}^{\text{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) = \arg\min_{\hat{R}} \mathrm{E}\big[\big(\hat{R}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big)^2\big].$$

The standard result is that the MMSE estimator is the conditional mean of $R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})$ given $\underline{\boldsymbol{Z}}$ (see (1) or Appendices C and E.4), so

$$\hat{R}^{\text{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) = \mathrm{E}_{p(\boldsymbol{y}|\underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}})}\big[R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big|\underline{\boldsymbol{Z}}; \underline{\boldsymbol{\psi}}\big] \tag{35}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}_{p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)}[L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)|\boldsymbol{Z}_i; \boldsymbol{\psi}_i] \tag{36}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) p(y_i|\boldsymbol{Z}_i; \boldsymbol{\psi}_i). \tag{37}$$

The samples are independent, so $R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})$ is approximately normally distributed by the CLT. The MMSE estimator guesses the conditional mean of this normal RV given $\underline{\boldsymbol{Z}}$.

Given $\underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}$, the MMSE estimate of the empirical-risk RV is

$$\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}) = \mathrm{E}_{p(\boldsymbol{y}|\underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}})}\big[R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big|\underline{\boldsymbol{z}}; \underline{\boldsymbol{\psi}}\big] \tag{38}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)}[L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)|\boldsymbol{z}_i; \boldsymbol{\psi}_i] \tag{39}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{y_i \in \mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i), \tag{40}$$

where $p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$ is the *training class posterior*[13]:

$$p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i) = \frac{\pi(y_i) p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i)}{\sum_{y_i' \in \mathcal{Y}} \pi(y_i') p(\boldsymbol{z}_i|y_i'; \boldsymbol{\psi}_i)}. \tag{41}$$

Having found the MMSE estimator of the empirical-risk RV, we can now speak of *MMSE training*, which seeks $\boldsymbol{\theta}^*$ that minimizes (40), the MMSE estimate of the empirical-risk RV. Equation (28) becomes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}) + \lambda J_{\mathrm{reg}}(\boldsymbol{\theta})\big]$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N}\sum_{y_i \in \mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i) + \lambda J_{\mathrm{reg}}(\boldsymbol{\theta}).$$

MMSE training handles correctly-labeled samples in a natural way. For the i^{th} sample, if the correct label is known to be ℓ, then $p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i) = \mathbb{1}(y_i = \ell)$, and the sample contributes its usual loss-function value to the empirical-risk term.

The bottom row of Table 9 summarizes training for non-probabilistic models. Ratner et al. (2016, 2017) employed (39) for discriminative models including binary logistic regression and long short-term memory recurrent neural networks. Khetan et al. (2018, §4) proposed (40) for arbitrary loss functions, used it for joint estimation of $p(\boldsymbol{z}|y)$ and training, and provided performance guarantees for binary classification. Neither group of authors demonstrated the link to MMSE estimation, so this section provides another way to motivate and arrive at this result.

We close this section by considering the loss function for a single sample. In (36), each term in the summation corresponds to the MMSE estimator of an individual loss-function RV $L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)$, namely

$$\hat{L}^{\mathrm{MMSE}}(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), \boldsymbol{Z}_i) = \mathrm{E}_{p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)}[L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)|\boldsymbol{Z}_i; \boldsymbol{\psi}_i], \quad i = 1, \ldots, N, \tag{42}$$

$$= \sum_{y_i \in \mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) p(y_i|\boldsymbol{Z}_i; \boldsymbol{\psi}_i), \quad i = 1, \ldots, N. \tag{43}$$

If the loss function has properties such as non-negativity or convexity, then (43) preserves such properties because $p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$ is a probability distribution. Not surprisingly, the MMSE estimator of the empirical-risk RV is just the average of the MMSE estimators of the individual loss-function RVs.

13. The training class posterior differs from the testing class posterior (10), which included the predicted labels and OP parameters. It is the same as (21) in Section 2.4.2.

3.4.2 Standard Properties

Standard properties of MMSE estimators apply to $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$; see Appendix C.1. It is unbiased in the Bayesian sense:[14]

$$\mathrm{E}\big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})\big] = \mathrm{E}\big[R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big]. \tag{44}$$

Likewise, the MMSE estimator of each loss function is also unbiased:

$$\mathrm{E}\big[\hat{L}^{\mathrm{MMSE}}(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), \boldsymbol{Z}_i)\big] = \mathrm{E}[L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i)], \quad i = 1, \ldots, N. \tag{45}$$

From (44), the mean of the estimation error is zero:

$$\mathrm{E}\big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big] = 0,$$

and the variance of the estimation error equals the MSE:

$$\mathrm{var}\big(\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big) = \mathrm{E}\big[\big(\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big)^2\big] \tag{46}$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \mathrm{E}\big[\mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i\big)\big]. \tag{47}$$

Appendix E.4 derives (47) and shows that the estimation error converges to a normal RV with the above moments. Note that the MSE of $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$ *is* the MMSE. Other properties, such as the orthogonality principle, also hold but are not relevant here.

The MMSE *estimator* is a function of the RV $\underline{\boldsymbol{Z}}$, so it is also an RV, and (46) and (47) describe the MMSE considering the distribution of $\underline{\boldsymbol{Z}}$. In contrast, the MMSE *estimate* is the MMSE estimator evaluated at $\underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}$ and is not random. The MSE that was realized by (39) or (40) equals the conditional variance of the empirical-risk RV given $\underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}$:

$$\mathrm{E}\big[\big(\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big)^2 \,\big|\, \underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}\big] = \mathrm{var}\big(R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y}) \,\big|\, \underline{\boldsymbol{Z}} = \underline{\boldsymbol{z}}\big)$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i = \boldsymbol{z}_i\big).$$

3.4.3 Gradient Calculation

Training often employs some form of gradient descent to find $\boldsymbol{\theta}^*$. For example, deep neural network training uses automatic differentiation to calculate the gradient of the empirical risk. In the ideal case,

$$\frac{\partial J_{\mathrm{ideal}}}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\big[\mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y})\big] + \lambda \frac{\partial J_{\mathrm{reg}}}{\partial \theta_j}, \tag{48}$$

and from (33)

$$\frac{\partial}{\partial \theta_j}\big[\mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{y})\big] = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial \theta_j}\big[L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i)\big] = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial L}{\partial \tilde{g}}(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) \frac{\partial \tilde{g}}{\partial \theta_j}(\boldsymbol{x}_i; \boldsymbol{\theta}). \tag{49}$$

14. Specifically, $\mathrm{E}_{p(\underline{\boldsymbol{z}}; \underline{\psi})}[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})] = \mathrm{E}_{p(\underline{\boldsymbol{z}}; \underline{\psi})}\big[\mathrm{E}_{p(\boldsymbol{y} | \underline{\boldsymbol{z}}; \underline{\psi})}[R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y}) \big| \underline{\boldsymbol{Z}}; \underline{\psi}]\big] = \mathrm{E}_{p(\boldsymbol{y})}[R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})]$.

Training based on MMSE estimation replaces the partial derivative $\partial\big[\mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y})\big]/\partial\theta_j$ with $\partial\big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}})\big]/\partial\theta_j$. From (40), this term is

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}})\big] &= \frac{1}{N}\sum_{i=1}^{N}\sum_{y_i\in\mathcal{Y}} p(y_i|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\frac{\partial}{\partial\theta_j}\big[L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),y_i)\big] \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{y_i\in\mathcal{Y}} p(y_i|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\frac{\partial L}{\partial\tilde{g}}(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),y_i)\frac{\partial\tilde{g}}{\partial\theta_j}(\boldsymbol{x}_i;\boldsymbol{\theta}). \qquad (50)\end{aligned}$$

It is a convex combination of the partial derivatives of the loss function. It is also compatible with automatic differentiation methods and inexpensive to compute. For a deep neural network, calculating $\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta})$ and $\partial\tilde{g}/\partial\theta_j$ represents the vast majority of the computational burden, and these quantities must only be computed once—the same burden as in the ideal case. Calculating $\partial L/\partial\tilde{g}$ requires a negligible amount of computation for a typical loss function, so calculating it for each possible value of y_i does not substantially increase the computational burden.

An example of the partial derivatives for binary logistic regression appears in Section 5.4 and Appendix F.2.

3.4.4 Special Cases

We briefly consider two special cases at opposite extremes. First, suppose that the correct label can be perfectly recovered from the noisy labels; i.e., $p(y_i'|\boldsymbol{z}_i;\boldsymbol{\psi}_i) = \mathbb{1}(y_i' = y_i)$. Then MMSE estimation returns the correct values of the empirical risk and loss function. For example, (40) yields $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}}) = R(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) = \mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y})$, and (43) gives $\hat{L}^{\mathrm{MMSE}}(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),\boldsymbol{Z}_i) = L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),Y_i)$. Likewise, the partial derivative in (50) reduces to (49): $\frac{\partial}{\partial\theta_j}\big[\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}})\big] = \frac{\partial}{\partial\theta_j}\big[\mathrm{R}^{\mathrm{emp}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y})\big]$.

Second, suppose that the noisy labels provide no information about the correct-label RVs; i.e., $p(\boldsymbol{z}|y;\boldsymbol{\psi}) = p(\boldsymbol{z};\boldsymbol{\psi})$. Then $p(y|\boldsymbol{z};\boldsymbol{\psi})$ reduces to $\pi(y)$, and (37) becomes

$$\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{Z}}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{y_i\in\mathcal{Y}} L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),y_i)\pi(y_i) = \mathrm{E}_{\pi(y)}\big[R(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{Y})\big].$$

Regardless of the value of $\underline{\boldsymbol{Z}}$, the MMSE estimator always returns the mean of the empirical-risk RV, taken with respect to the class prior $\pi(y)$. The partial derivative in (50) behaves similarly. The MMSE estimator remains unbiased, so (44) still holds. However, the estimator is a constant, so its variance is zero. By the law of total variance, its MSE is as large as possible and equals the variance of $R(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{Y})$ (see Appendix C.1, Item 4).

3.4.5 Consistency of the MMSE Estimator

For ideal training, an important reason for minimizing the empirical risk (33) for a class of predictive models is *consistency of the ERM principle*: The ideal empirical risk converges in probability to the minimum achievable risk as $N\to\infty$, even though the true distribution of $(\boldsymbol{X}_i,Y_i)_{i=1}^N$ is unknown (Vapnik, 1991). The relationship between MMSE estimation of the empirical-risk RV and consistency of the ERM principle requires further study. In the meantime, we consider the consistency of the MMSE estimator itself.

The estimator $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$ is *consistent* if it converges in probability to $R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})$ as $N \to \infty$. In this way, it attains the true value of the empirical-risk RV. Also, an estimator is *mean-square consistent* if its MSE goes to zero as $N \to \infty$, and mean-square consistency implies consistency. Hence, if $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$ is mean-square consistent, then it is consistent.

From (46) and (47), the MSE of $\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$ is

$$\mathrm{E}\big[\big(\hat{R}^{\mathrm{MMSE}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}) - R(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y})\big)^2\big] = \frac{1}{N^2} \sum_{i=1}^{N} \mathrm{E}\big[\mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i\big)\big].$$

A sufficient condition for when the MMSE estimator is consistent is therefore:

$$\lim_{N \to \infty} \frac{1}{N^2} \sum_{i=1}^{N} \mathrm{E}\big[\mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i\big)\big] = 0. \tag{51}$$

If there exists a constant b such that

$$\mathrm{E}\big[\mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i\big)\big] < b, \quad i = 1, \ldots, N, \tag{52}$$

then $\lim_{N \to \infty} N^{-2} \sum_{i=1}^{N} \mathrm{E}\big[\mathrm{var}\big(L(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), Y_i) \,\big|\, \boldsymbol{Z}_i\big)\big] < \lim_{N \to \infty} b/N = 0$, and (51) is satisfied. Hence, if the loss function is bounded, then clearly (52) is fulfilled. If the original loss function is unbounded, then one may be able to modify it to make it bounded; for example, once the original loss exceeds some threshold, it could be transformed to asymptotically approach an upper limit.

3.5 Advantages of MMSE Training

Within the unified view of training, MMSE training for non-probabilistic models has a number of appealing benefits compared to ML or MAP training for probabilistic models. MMSE training can continue to use the original loss function and involves a simple modification to the empirical risk calculation. The MMSE estimator gains standard properties such as Bayesian unbiasedness. Gradient descent and automatic differentiation can be used with minor modifications. The MMSE estimator is a consistent estimator of the empirical-risk RV if the loss function is bounded.

In contrast, the ML or MAP training approach can require new derivations, and it may not be able to leverage existing loss functions and their gradients. The resulting expressions could complicate theoretical analysis.

These differences are evident in the example for binary logistic regression, presented in Section 5.4. Equations for the primary terms and gradients are given in Appendix F.

3.6 Alternative Training Approaches

The unified view does not cover all possible approaches to training, and here we discuss some alternatives.

3.6.1 Constructing Weak Losses for Partial Labels

An alternative Bayesian approach by Cid-Sueiro (2012) and Cid-Sueiro et al. (2014) studied theoretical properties of weak losses for partial labels.[15] Labeling used length-C binary-encoded vectors. For class y, the correct label was one-hot encoded as $\bar{y} = \bar{e}_y$, where the vector $\bar{e}_j$ equals one at element j and zero elsewhere. The partial-label vector $\bar{z}$ was similarly encoded, but multiple elements could be set to one, so the noisy labeler could indicate more than one class. The authors considered different constraints on the permitted partial-label vectors, so the number of possible partial-label vectors, B, could be an integer between C and 2^C.

Recall that $s = \tilde{g}(\boldsymbol{x}; \boldsymbol{\theta})$. The authors defined a weak loss $L^{wk}(s, \bar{z})$ to be a loss function designed to operate on partial labels rather than correct ones. They then introduced an equivalent loss $L^{eq}(s, \bar{y})$ for correct labels:

$$L^{eq}(s, \bar{y}) = E_{p(\bar{z}|\bar{y})}[L^{wk}(s, \bar{Z})|\bar{Y} = \bar{y}], \tag{53}$$

and they showed that

$$\begin{aligned} E_{p(\bar{z})}[L^{wk}(s, \bar{Z})] &= \sum_{\bar{z}} p(\bar{z}) L^{wk}(s, \bar{z}) \\ &= \sum_{\bar{z}} \sum_{\bar{y}} p(\bar{y}) p(\bar{z}|\bar{y}) L^{wk}(s, \bar{z}) \\ &= \sum_{\bar{y}} p(\bar{y}) \sum_{\bar{z}} p(\bar{z}|\bar{y}) L^{wk}(s, \bar{z}) \\ &= E_{p(\bar{y})}\big[E_{p(\bar{z}|\bar{y})}[L^{wk}(s, \bar{Z})|\bar{Y}]\big] \qquad (54) \\ &= E_{p(\bar{y})}[L^{eq}(s, \bar{Y})]. \qquad (55) \end{aligned}$$

From this relationship they reasoned that training on partial labels with $L^{wk}(s, \bar{z})$ will behave like training on correct labels with $L^{eq}(s, \bar{y})$.

Given an original loss function $L(s, y)$ intended for correct labels, the equivalent loss is $L^{eq}(s, \bar{e}_y) = L(s, y)$, and one would like to find a corresponding weak loss. To this end, the authors expanded (53) into a matrix equation:

$$\begin{bmatrix} L^{eq}(s, \bar{e}_0) \\ L^{eq}(s, \bar{e}_1) \\ \vdots \\ L^{eq}(s, \bar{e}_{C-1}) \end{bmatrix} = \begin{bmatrix} p_{\bar{Z}|\bar{Y}}(\bar{b}_0|\bar{e}_0) & p_{\bar{Z}|\bar{Y}}(\bar{b}_1|\bar{e}_0) & \cdots & p_{\bar{Z}|\bar{Y}}(\bar{b}_{B-1}|\bar{e}_0) \\ p_{\bar{Z}|\bar{Y}}(\bar{b}_0|\bar{e}_1) & p_{\bar{Z}|\bar{Y}}(\bar{b}_1|\bar{e}_1) & \cdots & p_{\bar{Z}|\bar{Y}}(\bar{b}_{B-1}|\bar{e}_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_{\bar{Z}|\bar{Y}}(\bar{b}_0|\bar{e}_{C-1}) & p_{\bar{Z}|\bar{Y}}(\bar{b}_1|\bar{e}_{C-1}) & \cdots & p_{\bar{Z}|\bar{Y}}(\bar{b}_{B-1}|\bar{e}_{C-1}) \end{bmatrix} \begin{bmatrix} L^{wk}(s, \bar{b}_0) \\ L^{wk}(s, \bar{b}_1) \\ \vdots \\ L^{wk}(s, \bar{b}_{B-1}) \end{bmatrix}, \tag{56}$$

where $\bar{b}_i$ denotes the i^{th} possible partial-label vector, $i = 0, 1, \ldots, B-1$. The matrix tabulates the conditional distribution $p(\bar{z}|\bar{y})$, so it has dimensions $C \times B$. Cid-Sueiro pointed out that, for $B > C$, an infinite number of solutions for $L^{wk}(s, \bar{z})$ exist.

The MMSE training approach proposed in Section 3.4.1 is also Bayesian, but rather than design a new loss function for noisy labels, it uses the MMSE estimator $\hat{L}^{\text{MMSE}}(s, \boldsymbol{Z})$ of an original loss function $L(s, Y)$ meant for correct labels. MMSE training assumes that

15. Portions of the work by van Rooyen and Williamson (2018) are closely related to this approach.

the noisy labels do not depend on the feature vector because the noisy-label model has the form $p(\boldsymbol{z}|y)\pi(y)$.

From (42) and (43), the MMSE estimator is the conditional mean of the original loss function given $\boldsymbol{Z}$ and is just a convex combination of the original loss function values for each possible y, weighted by $p(y|\boldsymbol{Z})$. For the approach taken by Cid-Sueiro et al., (53) suggests that the equivalent loss could be interpreted as the MMSE estimate of the weak loss given $\bar{Y} = \bar{y}$. To find a weak loss for a given equivalent loss, one must solve the matrix equation (56).

From (45), the MMSE estimator is unbiased in the Bayesian sense, a standard result obtained by iterated expectations:

$$\mathrm{E}_{p(\boldsymbol{z})}[\hat{L}^{\mathrm{MMSE}}(s, \boldsymbol{Z})] = \mathrm{E}_{p(\boldsymbol{z})}\big[\mathrm{E}_{p(y|\boldsymbol{z})}[L(s, Y)|\boldsymbol{Z}]\big] = \mathrm{E}_{p(y)}[L(s, Y)].$$

Cid-Sueiro et al. obtain a similar relationship, but in the opposite order: (54) and (55) give

$$E_{p(\bar{z})}[L^{wk}(s, \bar{Z})] = E_{p(\bar{y})}\big[E_{p(\bar{z}|\bar{y})}[L^{wk}(s, \bar{Z})|\bar{Y}]\big] = E_{p(\bar{y})}[L^{eq}(s, \bar{Y})].$$

This relationship corresponds to Bayesian unbiasedness of the MMSE estimator of the weak loss given $\bar{Y}$.

MMSE training conditions on the noisy-label RV $\boldsymbol{Z}$. *After* $\boldsymbol{Z} = \boldsymbol{z}$ is observed, it uses $p(y|\boldsymbol{z})$ to estimate the original loss function. The work of Cid-Sueiro et al. conditions on the correct-label RV $\bar{Y}$. *Before* $\bar{Z}$ is observed, they tabulate $p(\bar{z}|\bar{y})$ for all combinations of $\bar{z}$ and $\bar{y}$, and they solve (56) to calculate the weak loss for every possible realization of $\bar{Z}$ that might occur.

In summary, these two Bayesian approaches bear some similarities but address truthing issues differently. Neither one estimates the correct labels. One could be interpreted as operating in the reverse direction of the other. Cid-Sueiro et al. take the equivalent loss for correct labels and construct a weak loss for noisy labels. MMSE training takes the noisy labels and estimates the original loss function for correct labels.

3.6.2 Using Proxy Loss Functions

Another alternative, proposed by Natarajan et al. (2013, 2018), took a *classical* (i.e., frequentist) view when replacing the original loss function $L(s, y)$ of $s = \tilde{g}(\boldsymbol{x}; \boldsymbol{\theta})$ and y with a proxy loss function designed for noisy labels.[16] They considered binary classification with a single labeler and took the classical viewpoint, so y is unknown but non-random. Assuming that $p(z; y)$ is known, they devised a proxy loss function $L^{pr}(s, Z)$ for the noisy-label RV Z that is an unbiased estimator of $L(s, y)$ in the classical sense, meaning that $\mathrm{E}_{p(z;y)}[L^{pr}(s, Z)] = L(s, y)$, $\forall y \in \mathcal{Y}$ (see Papoulis 1991, §9-2; Kay 1993, §2.3). They then trained models on $\underline{\boldsymbol{z}}$ using $J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, p(z; y)) = N^{-1} \sum_{i=1}^{N} L^{pr}(\tilde{g}(\boldsymbol{x}_i; \boldsymbol{\theta}), z_i)$.

16. van Rooyen and Williamson (2018) generalized this approach to other situations where a loss function for correct labels can be modified to account for noisy labels.

The proxy loss function is the solution of a system of C linear equations in C unknowns:[17]

$$\begin{bmatrix} p_{Z;y}(0;0) & p_{Z;y}(1;0) & \cdots & p_{Z;y}(C-1;0) \\ p_{Z;y}(0;1) & p_{Z;y}(1;1) & \cdots & p_{Z;y}(C-1;1) \\ \vdots & \vdots & \ddots & \vdots \\ p_{Z;y}(0;C-1) & p_{Z;y}(1;C-1) & \cdots & p_{Z;y}(C-1;C-1) \end{bmatrix} \begin{bmatrix} L^{pr}(s,0) \\ L^{pr}(s,1) \\ \vdots \\ L^{pr}(s,C-1) \end{bmatrix} = \begin{bmatrix} L(s,0) \\ L(s,1) \\ \vdots \\ L(s,C-1) \end{bmatrix},$$

which is very similar to (56) in the approach by Cid-Sueiro (2012) and Cid-Sueiro et al. (2014). The matrix is just the conditional distribution $p(z;y)$; as long as it has full rank, a unique solution for $L^{pr}(s,Z)$ exists.

This approach exploits knowledge of $p(z;y)$ and does not estimate the correct labels. As a classical approach, it does not involve a class prior $\boldsymbol{\pi}$. Extending it to multiple labelers and varying combinations of labelers might be difficult. For T labelers, if every labeler provides a label for every sample, then the proxy loss function must satisfy $\mathrm{E}_{p(\boldsymbol{z};y)}[L^{pr}(s,\boldsymbol{Z})] = L(s,y)$, $\forall y \in \mathcal{Y}$, where $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_T)$. This requirement yields a system of C linear equations in C^T unknowns, which is underdetermined for $T > 1$. If the labelers can decline to provide a label but at least one labeler must provide a label for every sample, then the number of unknowns becomes $(C+1)^T - 1$.

In contrast, MMSE training in Section 3.4.1 is a *Bayesian* method that exploits the class prior $\boldsymbol{\pi}$ as well as $p(z|y)$. It employs the MMSE estimator of $L(s,Y)$, which from (42) is $\hat{L}^{\mathrm{MMSE}}(s,\boldsymbol{Z}) = \mathrm{E}_{p(y|\boldsymbol{z})}[L(s,Y)|\boldsymbol{Z}]$. This estimator is unbiased in the *Bayesian* sense, meaning $\mathrm{E}_{p(\boldsymbol{z})}[\hat{L}^{\mathrm{MMSE}}(s,\boldsymbol{Z})] = \mathrm{E}_{p(y)}[L(s,Y)]$ (cf. (45)). The classical and Bayesian viewpoints are fundamentally different (Kay, 1993, §10.3): The former treats y as an unknown, nonrandom quantity, and the latter treats y as an unobserved realization of the RV Y. Classical unbiasedness is thus not a paramount objective in Bayesian statistics (see Breiman 2001; Gelman et al. 2013, §4.5). Our training approach uses the original loss function, so a proxy loss function is unnecessary. Also, from (43), the MMSE estimator is a convex combination of original loss function values; no system of linear equations must be solved. Finally, MMSE training readily accommodates multiple labelers and different combinations of labelers for different samples.

3.6.3 Predicting the Noisy Labels

In another alternative training approach, Sukhbaatar et al. (2015) and Jindal et al. (2016) trained a composite neural network, which consists of a base network followed by an additional layer, to predict the noisy labels $\underline{\boldsymbol{z}}$ from a single labeler. The loss function is unchanged, but $\boldsymbol{y}$ is replaced by $\underline{\boldsymbol{z}}$. During training, the authors regularized both components of the composite network so that the base network learned $p(y|\boldsymbol{x})$ and the additional layer learned $p(z|y)$. Following training, the base network can be extracted and used to predict correct labels.

Let $nn_{\mathrm{b}}(\underline{\boldsymbol{x}};\boldsymbol{\theta})$ denote the base network, and $\ell_{\mathrm{a}}(nn_{\mathrm{b}}(\underline{\boldsymbol{x}};\boldsymbol{\theta});\boldsymbol{\psi})$ denote the additional layer, where $\boldsymbol{\psi}$ is a $C \times C$ matrix representing the layer's estimate of $p(z|y)$. Then the primary term for this approach is $J_{\mathrm{pri}}(\boldsymbol{\theta},\boldsymbol{\psi};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}}) = \frac{1}{N}\sum_{i=1}^{N} L\big(\ell_{\mathrm{a}}(nn_{\mathrm{b}}(\underline{\boldsymbol{x}};\boldsymbol{\theta});\boldsymbol{\psi}), z_i\big)$.

This approach does not estimate the correct labels, and it jointly estimates both $p(y|\boldsymbol{x})$ and $p(z|y)$. It does not consider the class prior $\boldsymbol{\pi}$. For a single labeler, no changes to the

17. Natarajan et al. only considered binary classification; we offer the formulation for arbitrary C.

loss function are necessary, but the loss function would have to be modified to account for multiple labelers or different combinations of labelers.

3.6.4 Predicting the Noisy Labels with Trace Regularization

For multiple independent labelers, Tanno et al. (2019) proposed jointly training a CNN and estimating the labelers' confusion matrices. They used the confusion matrices to adjust the softmax output of the CNN to obtain predicted scores for the noisy labels $\underline{\boldsymbol{z}}$. The training objective combined a cross-entropy loss term and a trace-regularization term. The cross-entropy loss was applied to the adjusted CNN outputs and noisy labels $\underline{\boldsymbol{z}}$, which amounts to predicting the noisy labels like Sukhbaatar et al. (2015). Trace regularization was introduced because, as the authors showed, doing so will drive the estimated confusion matrices to their actual values, under certain conditions.[18] Consequently, as the labelers' confusion matrices are estimated, the CNN learns to predict the correct labels.

This method does not estimate the correct labels. The estimated confusion matrices can be used to obtain estimates of $p(z_t, y)$ and $p(z_t|y)$ for each labeler, as well as $\boldsymbol{\pi}$. Tanno et al. also described some computational advantages of their method compared to the EM-based methods of Raykar et al. (2010) and Khetan et al. (2018).

3.7 Suboptimal, Infrastructure-Compatible Training

There exists significant existing infrastructure, like software packages or machine-learning frameworks, that expects correct—or assumed-to-be-correct—labels, and modifying it for one of the preceding methods might be impractical or costly. Below, we describe some training methods that are suboptimal but compatible with such infrastructure.

First, *label estimation* produces an estimate $\check{\boldsymbol{y}}$ of the correct-label RVs $\boldsymbol{Y}$ and trains on $\{\underline{\boldsymbol{x}}, \check{\boldsymbol{y}}\}$ instead of $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$. To estimate $\boldsymbol{Y}$, one can use the related work or apply the MPE criterion as in Section 2.6.1. The MPE estimator is given by $h_i^{\text{MPE}} = \arg\min_{h_i} \mathrm{E}[\mathbb{1}(h_i(\boldsymbol{Z}_i) \neq Y_i)]$. The standard result is that the solution is the MAP estimator (see (2) or Appendix D), so $\check{y}_i = \arg\max_{y \in \mathcal{Y}} p(y|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$, $i = 1, \ldots, N$. Label estimation is suboptimal (cf. Section 1.4) because training should estimate the primary term—e.g., the empirical-risk RV, as in Section 3.4.1—rather than $\boldsymbol{Y}$.

Second, *voting* trains T classifiers, where the t^{th} classifier g_t is trained on the samples and noisy labels—treated as if correct—from the t^{th} labeler only, namely $\{(\boldsymbol{x}_i, z_{i,t}) : z_{i,t} \neq \emptyset, i = 1, \ldots, N\}$. Given an unlabeled sample $\boldsymbol{x}$, the predicted label $\hat{y}$ is chosen by a majority vote among $\{g_t(\boldsymbol{x}; \boldsymbol{\theta}_t)\}_{t=1}^T$. This technique ignores estimation theory so it is suboptimal, and it multiplies training and deployment complexity by T.

Third, *sample replication*, which was suggested by Raykar et al. (2010), copies each $\boldsymbol{x}_i$ several times, assigns labels to its copies based on $p(y_i|\boldsymbol{z}_i; \boldsymbol{\psi}_i)$, and trains on the copies and their labels. For example, in binary classification, if $p_{Y_i|\boldsymbol{Z}_i}(0|\boldsymbol{z}_i; \boldsymbol{\psi}_i) = 0.2$, then $\boldsymbol{x}_i$ is replicated 5 times, with one copy assigned label 0 and four copies assigned label 1. Direct implementation could multiply the storage and computation requirements for training by the number of copies made.

18. Sukhbaatar et al. (2015) mentioned this result but applied different regularization for implementational reasons.

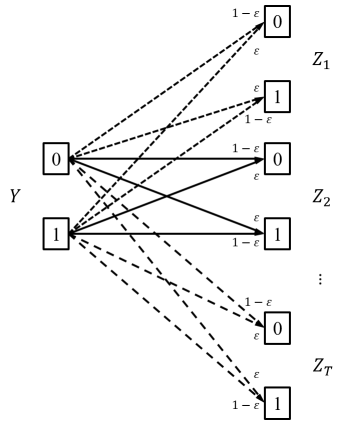


Figure 5: Binary symmetric broadcast channel.

These methods offer different compromises between accounting for truthing issues and modifying existing infrastructure. Label estimation makes a hard decision about $\boldsymbol{Y}$ before training commences, which could introduce mistaken labels into training but requires no other modifications. Voting considers different hard decisions about $\boldsymbol{Y}$ from the individual labelers, multiplies deployment complexity by T, and requires a simple voting mechanism. Sample replication never makes a hard decision about the correct-label RVs $\boldsymbol{Y}$, offers a quantized approximation of the approaches given in Sections 3.2, 3.3, and 3.4, and it multiplies training resource requirements by the number of copies, but it does not affect deployment complexity.

4. Comparing Combinations of Labelers: An Information-Theoretic View

The truthing issues have a simple information-theoretic interpretation. For a single sample, the correct-label RV Y can be viewed as the input to a channel, and the noisy-label RVs $\boldsymbol{Z} = (Z_1, \ldots, Z_T)$ can be viewed as the outputs from the channel.[19] Then the *mutual information* $I(\boldsymbol{Z}; Y) = \sum_{\boldsymbol{z},y} p(\boldsymbol{z}, y) \log_2 \big(p(\boldsymbol{z}, y)/p(\boldsymbol{z})\pi(y)\big)$ quantifies the amount of information that $\boldsymbol{Z}$ conveys about Y and vice versa (see Cover and Thomas, 1991).

Here, we show that the interpretation allows us to compare different combinations of labelers in terms of mutual information, at least in theory. In Section 5.5, we demonstrate that the training and testing techniques of Sections 2 and 3 enable us to exploit that information in practice.

4.1 Binary Symmetric Broadcast Channel

We illustrate this viewpoint for binary classification. For simplicity, we assume that the labelers or noisy-label RVs are conditionally independent, although the interpretation applies for conditionally dependent labelers as well. We also assume that all labelers have the same conditional distribution and that each labeler assigns a noisy label to every sample, so it suffices to consider a single sample.

19. In a theoretical context, Lugosi (1992) presented such a channel interpretation for a single labeler.

We can now introduce a *binary symmetric broadcast channel* (BSBC) (see Cover and Thomas, 1991, §14.6), shown in Figure 5 and parameterized by the *labeling-error probability* $\varepsilon \in [0,1]$. For $t \in \mathcal{T}$ and $y, z_t \in \{0,1\}$,

$$p(z_t|y;\varepsilon) = \begin{cases} 1-\varepsilon, & \text{if } z_t = y; \\ \varepsilon, & \text{if } z_t \neq y. \end{cases} \tag{57}$$

For $\varepsilon \in [0, 1/2]$, this model is equivalent to setting $C = 2$, $\delta_i \equiv 0$, $\forall i$, and $\phi_t \equiv 2\varepsilon$ in (59) and (60).

The binomial theorem can be used to obtain $I(\boldsymbol{Z}; Y|T, \varepsilon)$, the mutual information for T labelers, each with error probability ε:

$$\begin{aligned} I(\boldsymbol{Z}; Y|T, \varepsilon) = \pi(0) \sum_{m=0}^{T} \binom{T}{m} \varepsilon^m (1-\varepsilon)^{T-m} \log_2 \left(\varepsilon^m (1-\varepsilon)^{T-m}\right) \\ - \sum_{m=0}^{T} \binom{T}{m} \left(\pi(0)\varepsilon^m (1-\varepsilon)^{T-m} + \pi(1)(1-\varepsilon)^m \varepsilon^{T-m}\right) \\ \cdot \log_2 \left(\pi(0)\varepsilon^m (1-\varepsilon)^{T-m} + \pi(1)(1-\varepsilon)^m \varepsilon^{T-m}\right). \end{aligned}$$

For a fixed value of $\pi(1)$, $I(\boldsymbol{Z}; Y|T, \varepsilon)$ is maximized if $\varepsilon \in \{0,1\}$, in which case it equals $-\pi(0)\log_2 \pi(0) - \pi(1)\log_2 \pi(1) = H(Y)$, the entropy of Y. If $\varepsilon = 1/2$, then the labelers assign labels equiprobably, providing no information about Y, and $I(\boldsymbol{Z}; Y|T, \varepsilon) = 0$. If $\varepsilon \in (0, 1/2) \cup (1/2, 1)$, then $\lim_{T\to\infty} I(\boldsymbol{Z}; Y|T, \varepsilon) = H(Y)$, and all information about Y becomes available from $\boldsymbol{Z}$.

Figure 6 shows $I(\boldsymbol{Z}; Y|T, \varepsilon)$ for different values of T and ε when $\pi(1) = 0.1$. The plot is only shown for $0 \leq \varepsilon \leq 1/2$. As ε decreases from $1/2$ to zero, $I(\boldsymbol{Z}; Y|T, \varepsilon)$ increases from zero to its maximum as one would expect. For $\varepsilon \neq 0$, $I(\boldsymbol{Z}; Y|T, \varepsilon)$ increases with T because more noisy-label observations are available for estimating Y.

4.2 Equivalent Single Labeler

We can use $I(\boldsymbol{Z}_1; Y|T_1, \varepsilon_1)$ and $I(\boldsymbol{Z}_2; Y|T_2, \varepsilon_2)$ to compare the information provided by two different groups of labelers, but it can be helpful to think in terms of a single labeler. Let $I(Z; Y|\varepsilon')$ denote the mutual information for a single labeler with error probability ε', which corresponds to the standard (single-input, single-output) *binary symmetric channel* (BSC) (see Cover and Thomas, 1991, §8.1). Given $\boldsymbol{\pi}$, T, and ε, we can use binary search to find ε' such that $I(Z; Y|\varepsilon') = I(\boldsymbol{Z}; Y|T, \varepsilon)$ and express the T labelers as an *equivalent single labeler* with error probability ε'. Thus, we can compare different numbers of labelers with different error probabilities in terms of their equivalent single-labeler mutual information.

Figure 7 shows the relationship between $\boldsymbol{\pi}$, ε', and $I(Z; Y|\varepsilon')$ for the standard BSC. The mutual information is symmetric about $\pi(1) = 1/2$. For a fixed value of ε', it decreases as $\pi(1)$ decreases from $1/2$, which indicates that handling truthing issues becomes more challenging as the classes become more imbalanced.

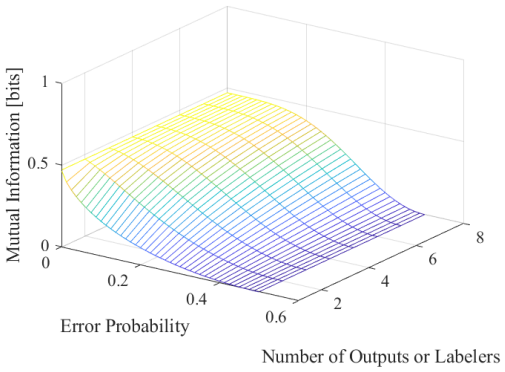


Figure 6: Graph of the mutual information $I(\boldsymbol{Z};Y|T,\varepsilon)$ of BSBC for $\pi(1) = 0.1$ as a function of T and ε.

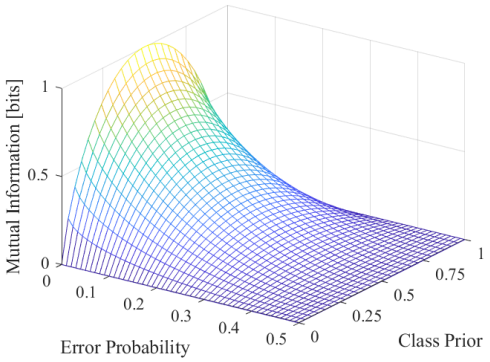


Figure 7: Mutual information $I(Z;Y|\varepsilon')$ of standard BSC as a function of class prior $\pi(1)$ and error probability ε'.

4.3 Multiple Mediocre Labelers or Single Expert Labeler

Figure 6 also indicates that, for $0 < \varepsilon' < \varepsilon < 1/2$, $I(\boldsymbol{Z};Y|T,\varepsilon)$ can equal or exceed $I(Z;Y|\varepsilon')$ if T is sufficiently large. Hence, *multiple mediocre labelers can be as informative as—or more informative than—a single expert labeler*. This result helps justify crowdsourcing and explain its successes. We can again use binary search to find the minimum value of T needed to satisfy $I(\boldsymbol{Z};Y|T,\varepsilon) \geq I(Z;Y|\varepsilon')$.

Figure 8 shows example curves for $\varepsilon' \in \{0.01, 0.02, 0.05, 0.10\}$ and $\pi(1) = 0.4$. The curves rise steeply for small values of T before diminishing returns set in; as $T \to \infty$, ε approaches an asymptote at $1/2$. Hence, a few mediocre labelers may suffice to obtain a small equivalent error probability ε'. Alternatively, one might desire a very small value of ε' that no single expert can achieve, but a few human—not superhuman—labelers might be able to attain it together.

This possibility is reminiscent of boosting (see Schapire, 1990; Freund and Schapire, 1997), in which multiple weak learners are leveraged to achieve the performance of a strong learner. We have not explored this relationship further but make some brief remarks. In boosting, the weak learners perform slightly better than random guessing, but the correct labels are known, and this information is exploited to improve performance. With truthing issues, the noisy labels might also be quite inaccurate and the correct labels are unobserved, but the conditional distribution $p(\boldsymbol{z}|y;\boldsymbol{\psi})$ is known, and it provides information about the correct labels for training and testing.

The information-theoretic implication of equivalent information is intriguing, but it does not explain *how* to exploit the information that Z or $\boldsymbol{Z}$ conveys about Y. Fortunately, the methods developed in Sections 2 and 3 provide a means to do so. They should produce better estimates as T grows because the variance of an optimal estimator decreases as more

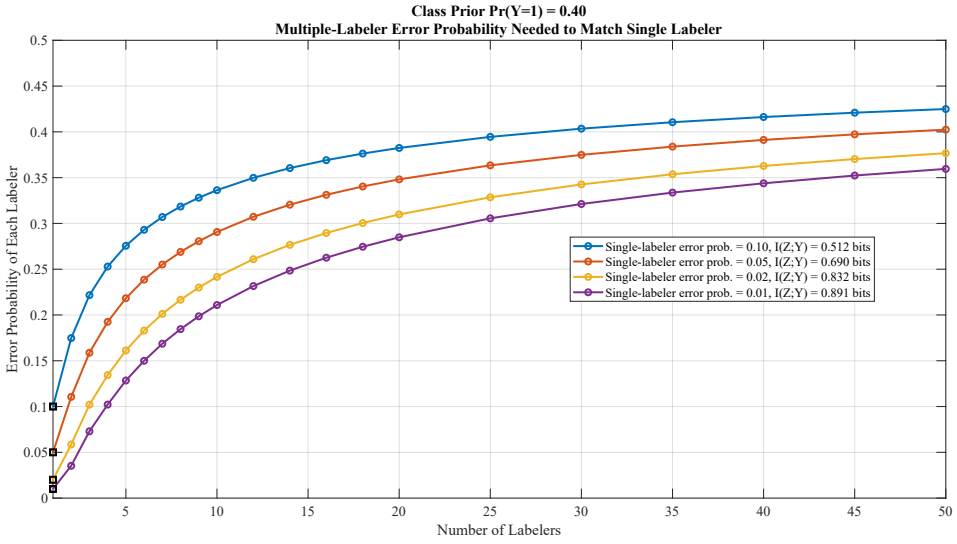


Figure 8: Number of labelers needed to achieve the same mutual information as a single labeler for $\pi(1) = 0.4$.

(independent) noisy observations become available.[20] For T sufficiently large, the estimates for the mediocre labelers should become comparable to those for the single expert labeler. Section 5.5 presents a couple of experiments that verify the implication.

5. Experiments

We conducted a number of experiments to see how the different testing and training methods performed and to check the implication of equivalent mutual information for different combinations of labelers.

5.1 Simulation

To exercise the testing methods, we need correct, noisy, and predicted labels. To study the training methods, we only need correct and noisy labels. These can all be generated via simulation. Of course, the methods do not use the correct labels, which would not be available in the intended applications, but simulation lets us compare the results with the ideal case.

5.1.1 Simulating Correct and Predicted Labels

Simulation for a desired class prior $\boldsymbol{\pi}$ is simple. For binary classification, the N correct-label realizations $\boldsymbol{y}^{(0)}$ are drawn independently $\mathrm{B}(\pi(1))$. For multi-class classification, they are

20. This behavior relates to T, the number of noisy labels per sample, rather than N, the number of samples. Increasing T means more noisy observations of Y are available to reduce the uncertainty about the correct label for a sample. Merely increasing N does not provide any more observations for the previous samples.

drawn independently from a categorical distribution with prior $\boldsymbol{\pi}$ and categories 0, 1, ..., $C-1$. Denote such a distribution as $\mathrm{Cat}\big(\boldsymbol{\pi}, (0, 1, \ldots, C-1)\big)$.

Experiments on the testing approaches of Section 2 require simulated predicted labels. To simulate a binary classifier with desired operating point $\big(p_{\mathrm{D}}^{\mathrm{des}},\ p_{\mathrm{FA}}^{\mathrm{des}}\big)$, the predicted-label realizations $\hat{\boldsymbol{y}}$ are drawn independently, with $\hat{y}_i$ drawn $\mathrm{B}(p_{\mathrm{FA}}^{\mathrm{des}})$ if $y_i^{(0)} = 0$, and $\hat{y}_i$ drawn $\mathrm{B}(p_{\mathrm{D}}^{\mathrm{des}})$ if $y_i^{(0)} = 1$. Likewise, the performance of a multi-class classifier can be specified by a desired conditional confusion matrix $\mathbf{K}^{\mathrm{des}}$, where matrix element $\mathbf{K}^{\mathrm{des}}_{n|\ell}$ contains $p(\hat{y} = n | y = \ell)$. Thus, each predicted-label realization $\hat{y}_i$ is drawn independently $\mathrm{Cat}\big((\mathbf{K}^{\mathrm{des}}_{0|\hat{y}_i}, \mathbf{K}^{\mathrm{des}}_{1|\hat{y}_i}, \ldots, \mathbf{K}^{\mathrm{des}}_{C-1|\hat{y}_i}), (0, 1, \ldots, C-1)\big)$.

5.1.2 Modeling and Simulating Noisy Labels

Since the samples are independent, we temporarily drop the sample index i while describing the modeling and simulation of the noisy-label RVs. Until this point, $p(\boldsymbol{z}|y; \boldsymbol{\psi})$ has remained completely general, so it allows for conditionally dependent labelers; that is, dependencies between different noisy-label RVs Z_t and $Z_{t'}$ for the same sample. We have also left $\boldsymbol{\psi}$ unspecified. For simulation, we need a more concrete model.

First, we assume that the noisy-label RVs for a sample are conditionally independent:

$$p(\boldsymbol{z}|y; \boldsymbol{\psi}) = \prod_{t: z_t \neq \emptyset} p(z_t|y; \psi_t), \tag{58}$$

where $\boldsymbol{\psi} = (\psi_t)_{t \in \mathcal{T}}$, and ψ_t contains parameters that determine how the t^{th} labeler labels the sample. This assumption is primarily for implementational convenience; we could have used a model with dependent labelers like one of the models in Holodnak et al. (2018).

Second, we let $\psi_t = (\delta, \phi_t)$, where $\delta \in [0, 1]$ is the *sample difficulty*, and $\phi_t \in [0, 1]$ is the *labeler fallibility*. The sample difficulty represents the ambiguity about the correct label inherent to the sample itself. For example, in image recognition, it could indicate the amount of blur or noise. The labeler fallibility is an attribute of the t^{th} labeler; it could reflect an analyst's experience or a crowdsourcing participant's trustworthiness.

The model is then

$$p(z|y; \delta, \phi_t) = \begin{cases} 1 - \varepsilon(\delta, \phi_t), & \text{if } z = y \in \mathcal{Y}; \\ \frac{\varepsilon(\delta, \phi_t)}{C-1}, & \text{if } z \neq y \in \mathcal{Y},\ z \in \mathcal{Y}; \\ 0, & \text{if } z \notin \mathcal{Y}; \end{cases} \tag{59}$$

where the *labeling-error probability* $\varepsilon(\delta, \phi_t)$ is a bilinear function:

$$\varepsilon(\delta, \phi_t) = (\delta - \delta\phi_t + \phi_t)\frac{C-1}{C}, \quad 0 \leq \delta \leq 1, 0 \leq \phi_t \leq 1. \tag{60}$$

The labeler chooses Z_t to be the correct label with probability $1 - \varepsilon(\delta, \phi_t)$ and makes a mistake with probability $\varepsilon(\delta, \phi_t) \in [0, 1]$. If a mistake occurs, then Z_t is equiprobably distributed over the $C-1$ possible incorrect labels. If $\delta = \phi_t = 0$, then $Z_t \equiv y$, but if either parameter is non-zero, then $\varepsilon(\delta, \phi_t)$ increases with both δ and ϕ_t. If either $\delta = 1$ or $\phi_t = 1$, then Z_t is equiprobably distributed over $\mathcal{Y}$.

Our model resembles the one by Whitehill et al. (2009), but our formulation is slightly different, and our purpose is significantly different. Their model is for binary classification, whereas our model applies to arbitrary C. Their model allows for adversarial labelers who tend to choose the opposite of the correct binary label, but when $C > 2$, the "opposite" of the correct label is not obvious. More important, Whitehill et al. concentrated on estimating the model parameters, while we focus on exploiting the model when its parameters are known. In the latter case, an adversarial labeler who is known to frequently choose the opposite of the correct label is actually more informative than a sloppy labeler who assigns labels equiprobably at random.[21]

Finally, we again consider all samples and let $\boldsymbol{\delta} = (\delta_i)_{i=1}^N$ and $\boldsymbol{\phi} = (\phi_t)_{t=1}^T$. Then $\underline{\boldsymbol{\psi}} = (\boldsymbol{\delta}, \boldsymbol{\phi})$, and $\boldsymbol{\psi}_i = (\delta_i, \boldsymbol{\phi})$, so

$$\begin{aligned} p(\underline{\boldsymbol{z}}|\boldsymbol{y}; \underline{\boldsymbol{\psi}}) &= \prod_{i=1}^N p(\boldsymbol{z}_i|y_i; \delta_i, \boldsymbol{\phi}) \\ &\overset{(a)}{=} \prod_{i=1}^N \prod_{t:z_{i,t}\neq\emptyset} p(z_{i,t}|y; \delta_i, \phi_t), \end{aligned} \tag{61}$$

where (a) is from (58).

During simulation, we draw δ_i, $i = 1, \ldots, N$, and $\boldsymbol{\phi}_t$, $t = 1, \ldots, T$, independently from beta or uniform distributions. We also draw a vector $\boldsymbol{\eta} = (\eta_t)_{t=1}^T$, where η_t is the approximate probability that the t^{th} labeler provides a label for a sample. Each η_t is drawn independently $\mathcal{U}(0, 1)$, where $\mathcal{U}(a, b)$ denotes a uniform distribution over (a, b). For the i^{th} sample, we repeatedly draw $\boldsymbol{\eta}_i \sim \big(\mathrm{B}(\eta_t)\big)_{t=1}^T$ with independent Bernoulli-distributed elements until at least one element of $\boldsymbol{\eta}_i$ equals unity. We then simulate $z_{i,t}$ in accord with (59) and (60) for $t \in \{t' : \eta_{i,t'} = 1\}$, and we set $z_{i,t} = \emptyset$ for $t \in \{t' : \eta_{i,t'} = 0\}$.

5.2 Examples of Testing: Binary Classification

A variety of examples that illustrate aspects of the testing approaches for binary classification appear here. We emphasize that, except for the ideal case, *no correct labels were used during testing.*

5.2.1 Main Testing Example

A simulation was conducted with $N = 10^3$, $T = 5$, $\pi(1) = 0.2$, $\big(p_{\mathrm{D}}^{\mathrm{des}}, p_{\mathrm{FA}}^{\mathrm{des}}\big) = (0.8, 0.3)$, $\delta_i \sim \mathrm{Beta}(1, 5)$, $\forall i$, and $\phi_t \sim \mathcal{U}(0, 0.4)$, $\forall t$. The simulator produced $\boldsymbol{\delta} = (0.098,\ 0.046,\ 0.347,\ \ldots,\ 0.199,\ 0.221)$, $\boldsymbol{\phi} = (0.249,\ 0.030,\ 0.387,\ 0.244,\ 0.154)$, and $\boldsymbol{\eta} = (0.727,\ 0.873,\ 0.286,\ 0.657,\ 0.232)$. The number of samples labeled by each labeler was 727, 879, 272, 667, 229, respectively; on average, there were about 2.8 noisy labels per sample.

Figure 9 shows the progression of the estimates $\tilde{p}_{\mathrm{D}}^{(j)}$ and $\tilde{p}_{\mathrm{FA}}^{(j)}$ for the iterative methods. The dotted lines show the ideal values of p_{D} and p_{FA} if the correct labels were known; the ideal values differ slightly from $\big(p_{\mathrm{D}}^{\mathrm{des}}, p_{\mathrm{FA}}^{\mathrm{des}}\big)$ because they are the result of sampling and simulation. For MMSE testing, both empirical Bayes methods (Algorithms 1 and 2), the

21. Ipeirotis et al. (2010, §3) made a similar observation in the context of assessing the cost of a labeler.

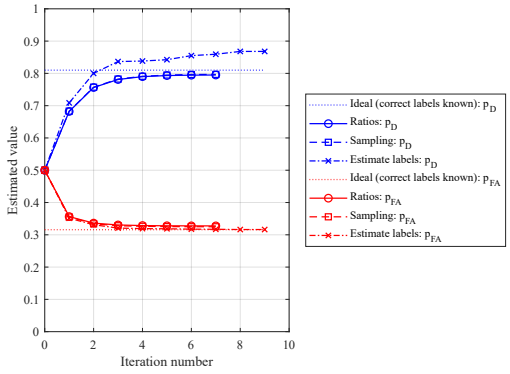


Figure 9: Main testing example: Progression of $(\tilde{p}_{\mathrm{D}}^{(j)}, \tilde{p}_{\mathrm{FA}}^{(j)})$ in iterative estimation methods.

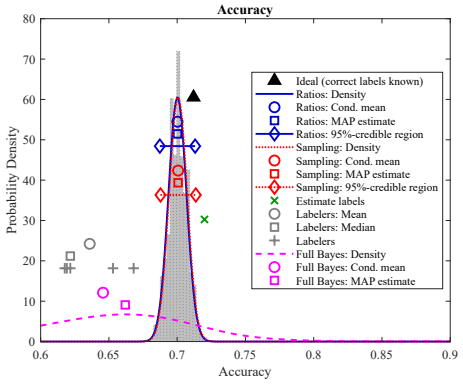


Figure 10: Main testing example: Accuracy estimates by all methods. Markers are spaced vertically for easier readability.

estimates improved on each iteration, and both converged to nearly the same final OP parameters after seven iterations. The suboptimal method of estimating the correct labels (Algorithm 3) converged after nine iterations; its final OP parameter $\tilde{p}_{\mathrm{FA}}$ was slightly more accurate than those of the empirical Bayes methods, but its final OP parameter of $\tilde{p}_{\mathrm{D}}$ was considerably less accurate than theirs.

Table 10 summarizes the estimates of all metrics by the techniques presented in Section 2, and Figure 10 shows the results for accuracy. Results for the suboptimal methods that do not fully exploit the predicted labels appear in Section 5.2.2. For MMSE testing with the empirical Bayes methods ("Ratios" and "Sampling"), the figure displays the estimated posterior density, conditional mean, MAP estimate, and 95%-credible region for ACC. The figure also displays a gray histogram, which was created by taking the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ from Algorithm 2, generating 5000 sample realizations of $\boldsymbol{Y}$ from (10), and computing the empirical accuracy for each realization. The optimal estimates are fairly close to the ideal accuracy, and the credible regions contain the ideal accuracy. The results for the two methods are very similar; this behavior was observed for all examples, so subsequent figures do not include results for Algorithm 2 beyond the histograms.

The figure also shows the estimated accuracy for the suboptimal method in Algorithm 3 ("Estimate labels"). This estimate is reasonably good, but it does not provide any sense of uncertainty like a credible region does. Also, this algorithm produced inaccurate estimates of REC or P_{D} (shown next in Figures 11 and 12).

Likewise, the figure displays the T instances of accuracy for each individual labeler ("Labelers"). We also computed the mean and median of these instances; we applied the algorithm by Vardi and Zhang (2000) to calculate the multi-dimensional median. The instances are fairly inaccurate, so their mean and median also yield poor estimates.

The last set of results in the figure are for the fully Bayesian approach. Its estimated density is very spread out as a result of marginalization over $(\tilde{P}_{\mathrm{D}}, \tilde{P}_{\mathrm{FA}})$, and its conditional

Estimate	Scalar Metrics					Joint Metrics	
	ACC	$PREC$	P_{D} or REC	P_{FA}	F_1	$(PREC, REC)$	$(P_{\mathrm{D}}, P_{\mathrm{FA}})$
Ideal (correct labels known)							
Ideal	0.712	0.421	0.810	0.316	0.554	(0.421, 0.810)	(0.810, 0.316)
MMSE Testing: Empirical Bayes estimation via ratios of jointly normal RVs (Alg. 1)							
Conditional mean	0.700[a]	0.397[a]	0.795	0.328	0.534	(0.397, 0.795)[b]	(0.795, 0.328)[b]
MAP estimate			0.795	0.325	0.529	(0.397, 0.795)[c]	(0.796, 0.325)[c]
Credible (lower)	0.687	0.373	0.764	0.317	0.506	—	—
region (upper)	0.713	0.420	0.828	0.335	0.553	—	—
MMSE Testing: Empirical Bayes estimation via sampling (Alg. 2)							
Conditional mean	0.701[a]	0.397[a]	0.796	0.327	0.535	(0.397, 0.796)[b]	(0.796, 0.327)[b]
MAP estimate			0.796	0.325	0.530	(0.397, 0.796)[c]	(0.796, 0.325)[c]
Credible (lower)	0.688	0.374	0.765	0.316	0.506	—	—
region (upper)	0.714	0.421	0.829	0.334	0.554	—	—
Estimation of correct labels (Alg. 3)							
Estimate labels	0.720	0.402	0.868	0.316	0.550	(0.402, 0.868)	(0.868, 0.316)
Combine metrics from individual labelers							
Mean	0.636	0.431	0.615	0.354	0.506	(0.431, 0.615)	(0.615, 0.354)
Median	0.622	0.421	0.605	0.354	0.504	(0.421, 0.605)	(0.605, 0.354)
Labeler index / No. of labels	Metrics from individual labelers						
1 / 727	0.622	0.450	0.574	0.354	0.505	(0.450, 0.574)	(0.574, 0.354)
2 / 879	0.653	0.414	0.643	0.343	0.504	(0.414, 0.643)	(0.643, 0.343)
3 / 272	0.618	0.405	0.605	0.377	0.485	(0.405, 0.605)	(0.605, 0.377)
4 / 667	0.619	0.421	0.587	0.366	0.490	(0.421, 0.587)	(0.587, 0.366)
5 / 229	0.668	0.465	0.667	0.331	0.548	(0.465, 0.667)	(0.667, 0.331)
Fully Bayesian estimation							
Conditional mean	0.646	0.345	0.664	0.360	0.456	(0.345, 0.664)[b]	(0.664, 0.360)[b]
MAP estimate	0.662	0.342	0.714	0.349	0.464	(0.347, 0.690)[c]	(0.663, 0.360)[c]

[a] For accuracy or precision, the empirical Bayes conditional mean and MAP estimate are identical.
[b] The conditional means for the joint metrics are the same as those for the corresponding scalar metrics.
[c] The MAP estimate of a joint metric can differ from the MAP estimates of its individual components.

Table 10: Testing metrics for main testing example.

mean and MAP estimate are not very accurate. This behavior occurred for other metrics, so we do not show the fully Bayesian method in the subsequent figures.

Next, Figure 11 displays results for the other scalar metrics. MMSE testing using the empirical Bayes method of Algorithm 1 produced estimates within about 0.025 of each metric. For REC or P_{D}, the credible region contains the ideal metric, and for the other metrics, it lies outside the credible region by just 0.001. Table 10 indicates that the credible regions of Algorithm 2 contained each ideal metric. The suboptimal method of estimating the correct labels (Algorithm 3) was quite accurate for several metrics but off by 0.058 for REC or P_{D}. The use of the labelers' labels often produced very inaccurate estimates.

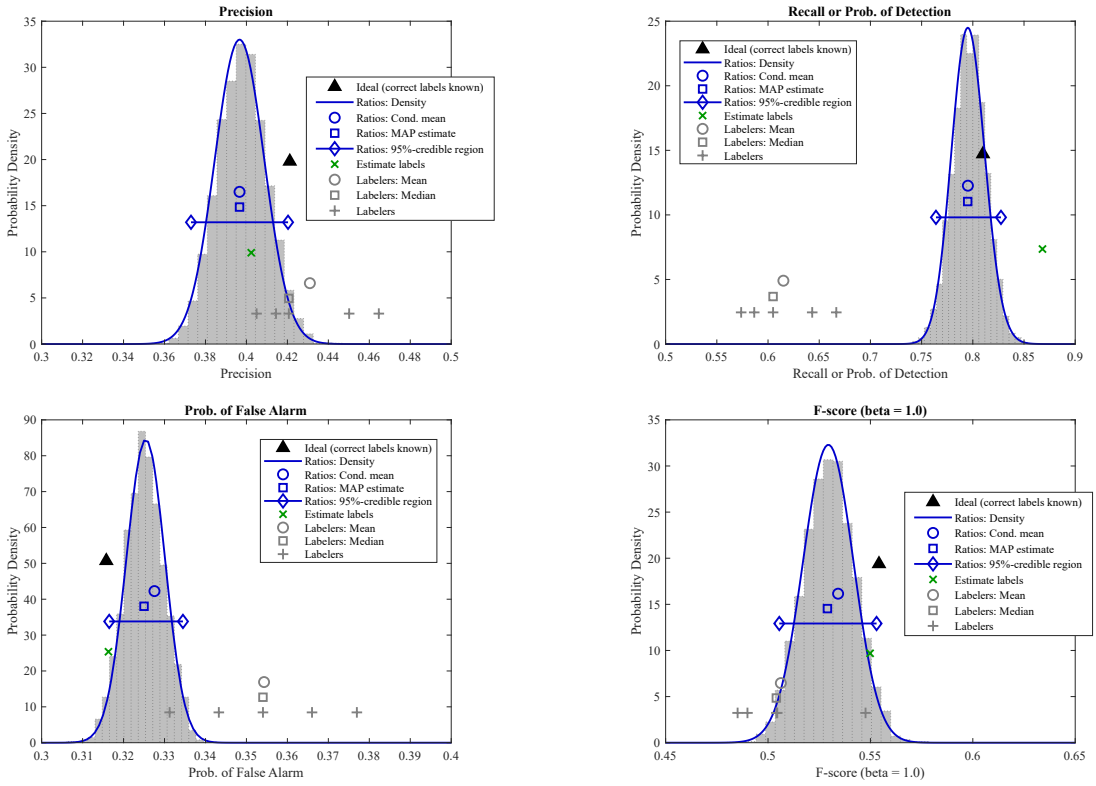


Figure 11: Main testing example: Estimates of scalar metric RVs.

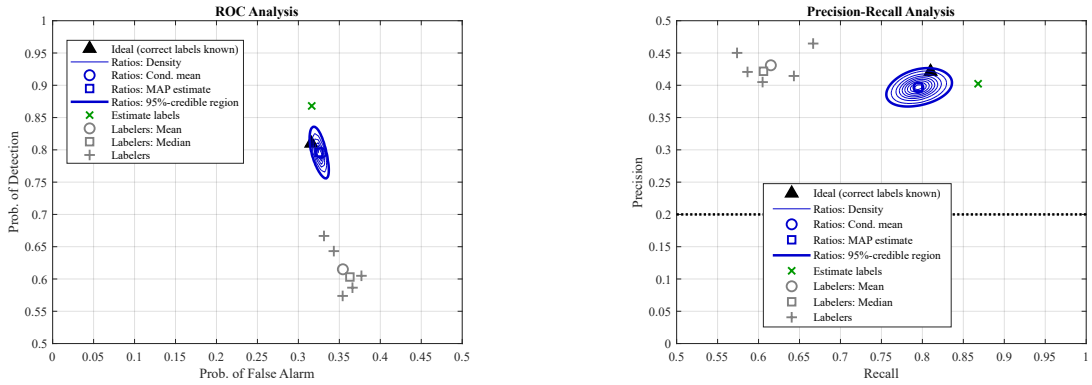


Figure 12: Main testing example: Estimates of joint metric RVs for ROC and P-R analysis.

Figure 12 presents results for P-R and ROC analysis. MMSE testing with empirical Bayes estimation produced the best joint estimates; its point estimates are closest to the ideal operating points, and its 95%-credible regions contain the ideal operating points. Estimating the correct labels yielded less accurate joint estimates, and it continues to provide no sense of uncertainty. Using the labelers' labels gave very poor estimates.

5.2.2 Exploitation of Predicted Labels

Full exploitation of the predicted labels $\hat{\boldsymbol{y}}$ and OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ is an important characteristic of the most successful testing methods. Figure 13 shows results for the simulation in Section 5.2.1 if the predicted labels are *not* fully exploited, which corresponds to the third and fourth suboptimal testing approaches in Section 2.6.1. They use $p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)$ rather than the testing model (7) with $p(\hat{y}_i|y_i;\tilde{p}_{\mathrm{D}},\tilde{p}_{\mathrm{FA}})p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)$, which amounts to setting $j_{\max}=0$ in Algorithm 1, 2, or 3. Compared to Figures 10 and 12, the estimated posterior and the estimates are quite inaccurate, and the credible regions do not contain the ideal metrics. These results illustrate the importance of including the predicted labels and OP parameters during estimation.

5.2.3 Convergence Experiments for Iterative Algorithms

Section 2.4.4 speculated that, when estimating $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ during MMSE testing, the empirical Bayes methods (Algorithms 1 and 2) will converge to the global optimum regardless of the initial OP parameters. To check this possibility, we conducted two experiments, which also included the suboptimal correct-label estimation method (Algorithm 3).

First, we used the same simulation as in Section 5.2.1 but varied the *initial OP parameters* $\left(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\right)$ over the 10×10 grid $\{0.05, 0.15, \ldots, 0.95\}\times\{0.05, 0.15, \ldots, 0.95\}$. Table 11 summarizes the results. The empirical Bayes methods converged to nearly the same final OP parameters every time, with maximum absolute errors of about 0.015. In addition, the MAC was always satisfied in Algorithm 1. The suboptimal method of Algorithm 3 had larger p_{D} errors but slightly smaller p_{FA} errors than the empirical Bayes methods. For this

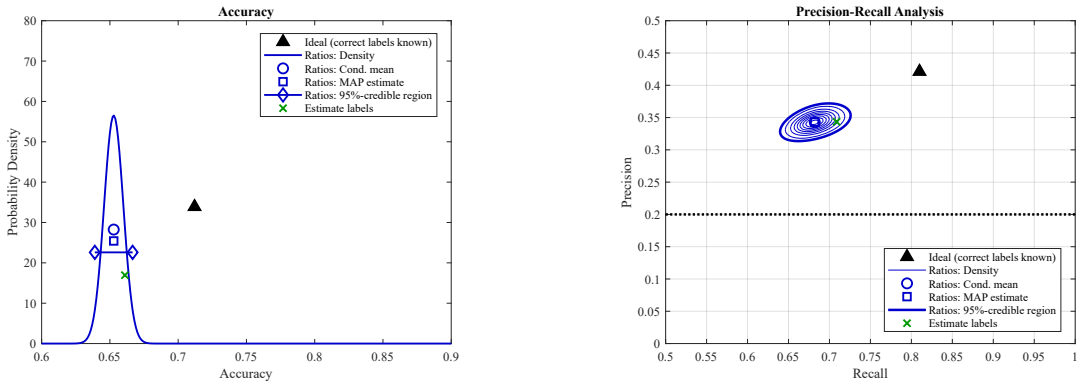


Figure 13: Examples of estimates of metric RVs if the predicted labels $\hat{\boldsymbol{y}}$ and OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ are *not* fully exploited. Compare with Figures 10 and 12. The plots do not show the labelers' results because they are unchanged from those figures.

Same Ideal Operating Point, Different Initial OP Parameters		Iterative Estimation Method		
		Ratios	Sampling	Est. Labels
Error $p_{\mathrm{D}} - \tilde{p}_{\mathrm{D}}$	Mean	0.0149	0.0131	-0.0473
	Std. dev.	3.20×10^{-4}	4.15×10^{-4}	1.61×10^{-2}
	Max abs.	0.0153	0.0142	0.0715
Error $p_{\mathrm{FA}} - \tilde{p}_{\mathrm{FA}}$	Mean	-0.0118	-0.0092	-0.0015
	Std. dev.	6.34×10^{-5}	8.56×10^{-5}	1.49×10^{-3}
	Max abs.	0.0119	0.0095	0.0033
Number of Iterations	Mean	6.6	6.6	6.2
	Std. dev.	1.08	1.05	2.35
	Max	8	8	10

Table 11: Estimation error statistics for the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ for iterative estimation methods. For the same ideal operating point $(p_{\mathrm{D}}, p_{\mathrm{FA}}) = (0.810, 0.316)$, 100 different initial OP parameters were used.

ideal operating point, all algorithms consistently converged to nearly the same final OP parameters.

Second, we ran another set of simulations that always initialized the iterative algorithms with the default initial OP parameters $\left(\tilde{p}_{\mathrm{D}}^{(0)}, \tilde{p}_{\mathrm{FA}}^{(0)}\right) = (1/2, 1/2)$, but we varied the *desired operating point* $\left(p_{\mathrm{D}}^{\mathrm{des}}, p_{\mathrm{FA}}^{\mathrm{des}}\right)$ over the same 10×10 grid as above, which produced 100 different ideal operating points. These simulations used $\pi(1) = 0.5$, $\delta_i \sim \mathcal{U}(0,1)$, $\forall i$, $\phi_t \sim \mathcal{U}(0, 0.5)$, and $\eta_t \sim \mathcal{U}(0,1)$, $\forall t$. Results appear in Table 12. The empirical Bayes methods converged every time, with average errors near 0.012 and a maximum absolute error below 0.045. Again, the MAC was always satisfied in Algorithm 1. The suboptimal method of Algorithm 3 performed less well. Although its average errors are no more than 0.021, its standard deviations are on the order of 0.200 and its maximum absolute error exceeds 0.333,

Different Ideal Operating Points, Same Default Initial OP parameters		Iterative Estimation Method		
		Ratios	Sampling	Est. Labels
Error $p_{\mathrm{D}} - \tilde{p}_{\mathrm{D}}$	Mean	-0.0113	-0.0114	-0.0141
	Std. dev.	1.04×10^{-2}	9.56×10^{-3}	1.95×10^{-1}
	Max abs.	0.0314	0.0310	0.3590
Error $p_{\mathrm{FA}} - \tilde{p}_{\mathrm{FA}}$	Mean	0.0120	0.0122	0.0210
	Std. dev.	1.36×10^{-2}	1.14×10^{-2}	2.03×10^{-1}
	Max abs.	0.0442	0.0381	0.3369
Number of Iterations	Mean	10.9	11.2	10.4
	Std. dev.	3.15	3.46	3.39
	Max	17	18	21

Table 12: Estimation error statistics for the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ for iterative estimation methods. For 100 different ideal operating points, the same default initial OP parameters was used.

indicating that it often became trapped near a local optimum. These results demonstrate the substantial benefit of the empirical Bayes methods over estimating the correct labels.

5.2.4 Estimation Performance for Different Operating Points

The convergence experiments in the previous section only examine the estimation error of the final OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$. For the second set of simulations in that section, we also compiled statistics on the estimation errors of the final estimates of the scalar and joint metrics over the 10×10 grid $\{0.05, 0.15, \ldots, 0.95\} \times \{0.05, 0.15, \ldots, 0.95\}$ of $\left(p_{\mathrm{D}}^{\mathrm{des}}, p_{\mathrm{FA}}^{\mathrm{des}}\right)$; i.e., over 100 different ideal operating points.

Figure 14 summarizes the error statistics for the estimated scalar metrics by the different testing approaches. For MMSE testing, the point estimates from the empirical Bayes methods (Algorithms 1 and 2) have average errors of about 0.012, with standard deviations near 0.011. *The scale changes between MMSE testing and the other methods.* The other methods have average errors between 0.001 and 0.043, but their standard deviations range from about 0.100 to over 0.200, an order of magnitude greater than those for the empirical Bayes methods.

Figures 15 and 16 display joint error statistics for P-R and ROC analysis. Results for MMSE testing were similar for both the conditional mean and MAP estimate, so the figures only show estimation errors for the conditional mean of Algorithm 1 and the MAP estimate of Algorithm 2. The empirical Bayes methods have average errors near 0.015 in each dimension. The square roots of the eigenvalues of the error covariance matrix fall between 0.005 and 0.025. *The scale changes between MMSE testing and the other methods.* The latter methods have average errors between 0.005 and 0.043, but the eigenvalues' square roots range from about 0.100 to over 0.430.

These results demonstrate the superior estimation performance of the MMSE testing methods, whose estimates typically lie within 0.010 to 0.040 of the ideal metrics over a wide range of ideal operating points. They significantly outperform the other methods,

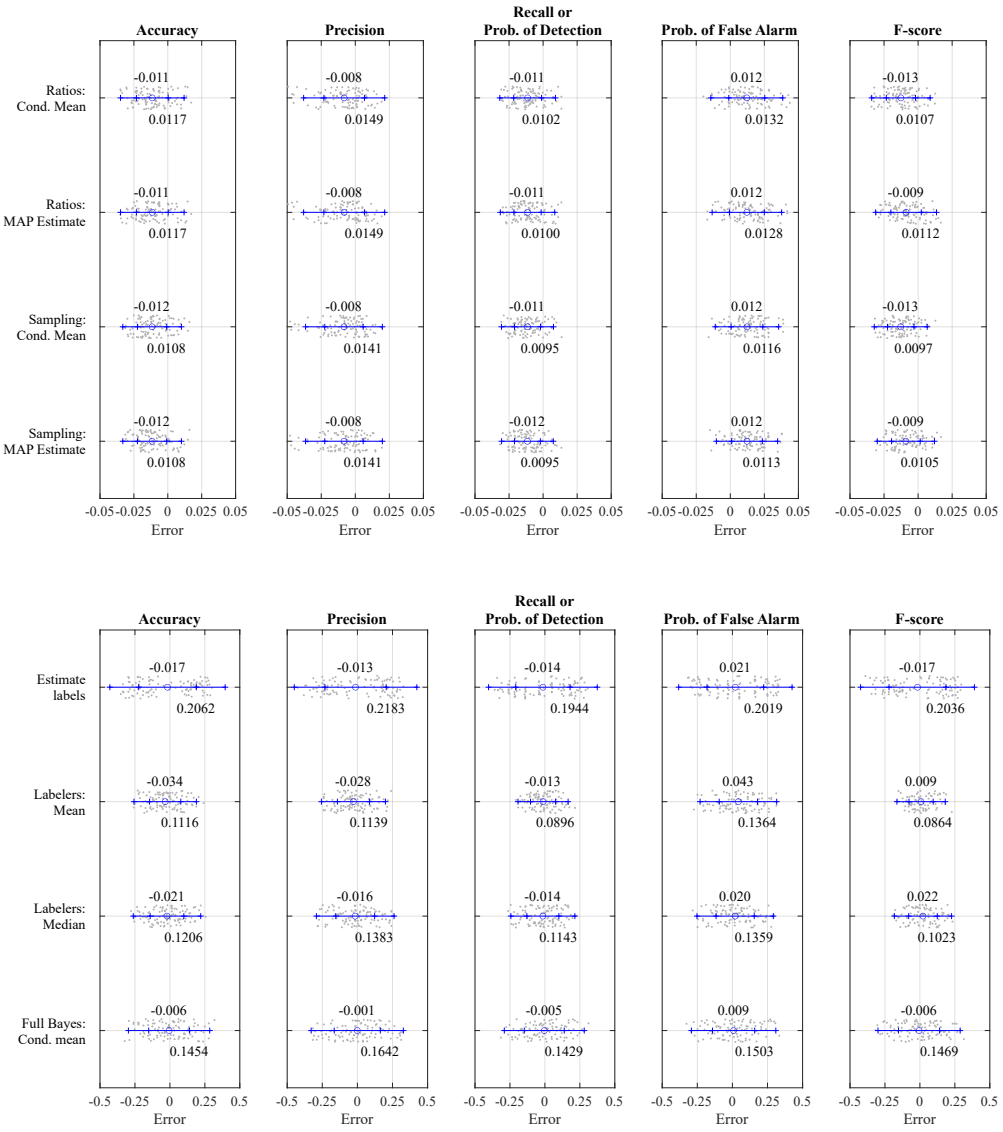


Figure 14: Scalar metric estimation errors for different testing approaches over 100 different ideal operating points. *The axis limits differ for MMSE testing with empirical Bayes* (upper plots: -0.05 to $+0.05$) *and the other approaches* (lower plots: -0.5 to $+0.5$). Miniature scatterplots of the estimation errors appear as gray dots. The average error is marked with a circle and as text above each circle. Multiples of ± 1 and ± 2 times the standard deviation of the errors appear as crosses, and text below the first cross gives the standard deviation.

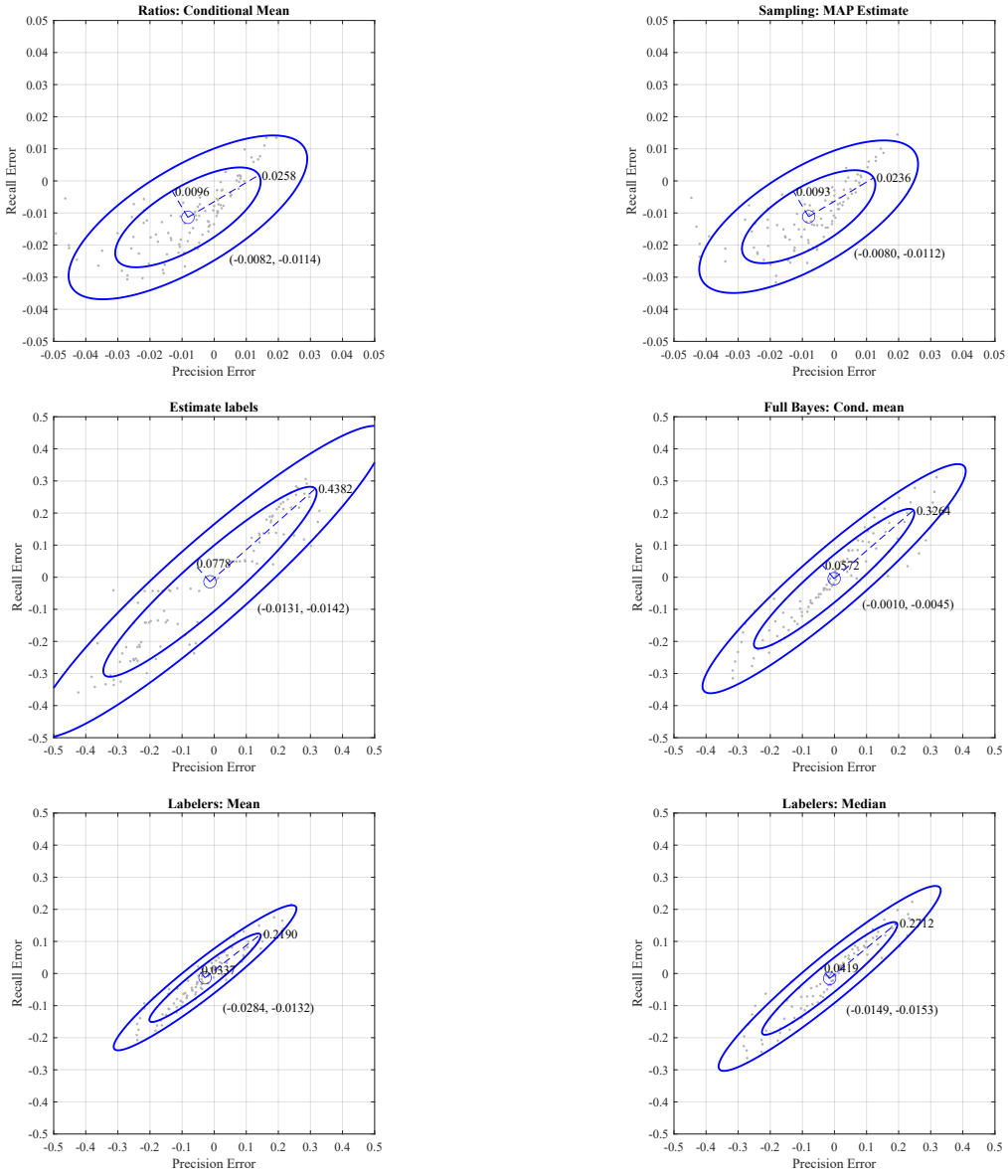


Figure 15: P-R analysis estimation errors for different testing approaches over 100 different ideal operating points. *The axis limits differ for MMSE testing with empirical Bayes* (top plots: -0.05 to $+0.05$) *and the other approaches* (middle and lower plots: -0.5 to $+0.5$). Miniature scatterplots of the estimation errors appear as gray dots. The average error is marked with a circle and listed as an ordered pair. Ellipses denote areas that account for 0.6827 and 0.9545 of the density of a bivariate normal distribution fitted to the errors, and text adjacent to the semi-major and semi-minor axes gives the square roots of the eigenvalues of the error covariance matrix.

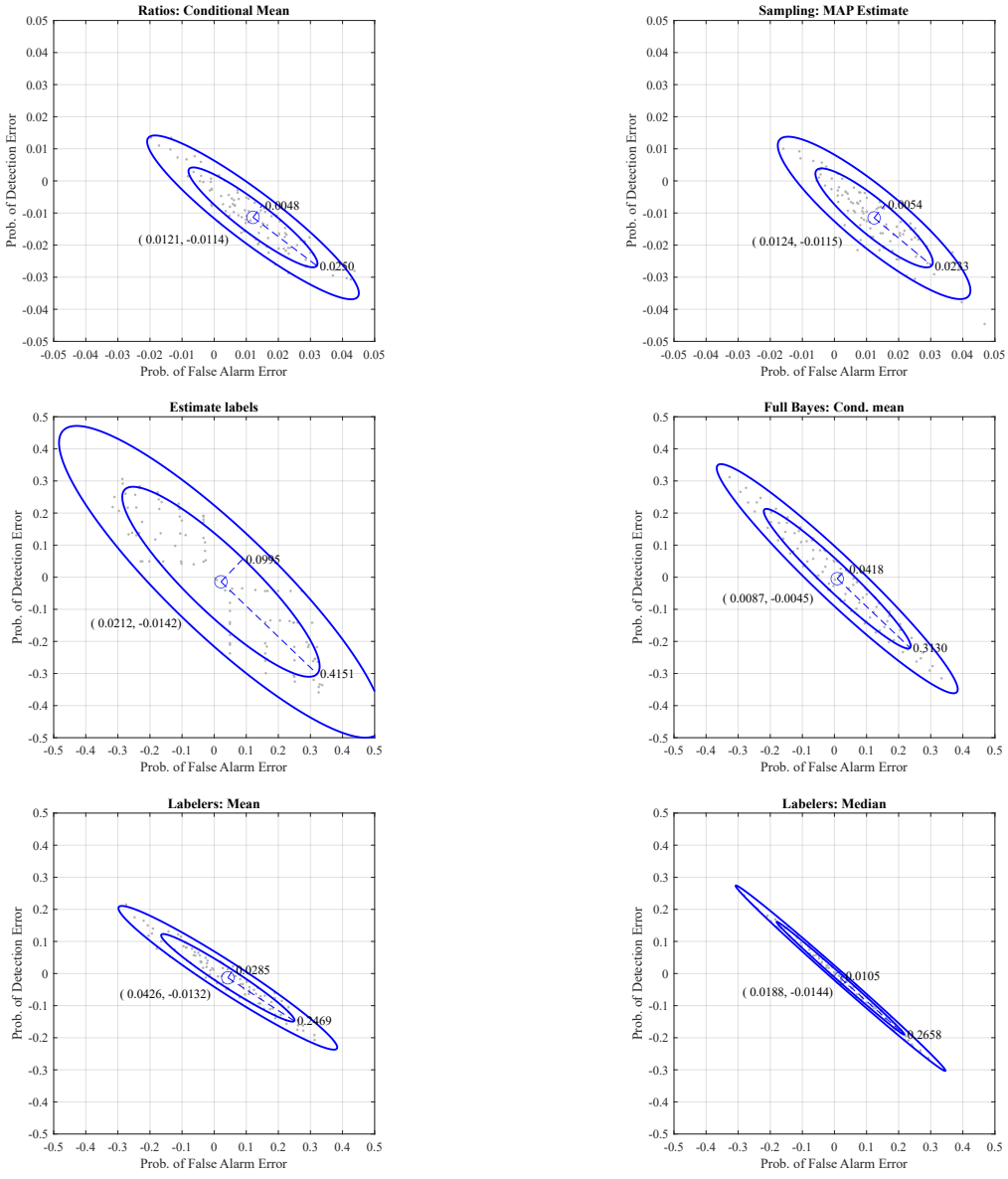


Figure 16: ROC analysis estimation errors for different testing approaches over 100 different ideal operating points. *The axis limits differ for MMSE testing with empirical Bayes* (top plots: -0.05 to $+0.05$) *and the other approaches* (middle and lower plots: -0.5 to $+0.5$). Miniature scatterplots of the estimation errors appear as gray dots. The average error is marked with a circle and listed as an ordered pair. Ellipses denote areas that account for 0.6827 and 0.9545 of the density of a bivariate normal distribution fitted to the errors, and text adjacent to the semi-major and semi-minor axes gives the square roots of the eigenvalues of the error covariance matrix.

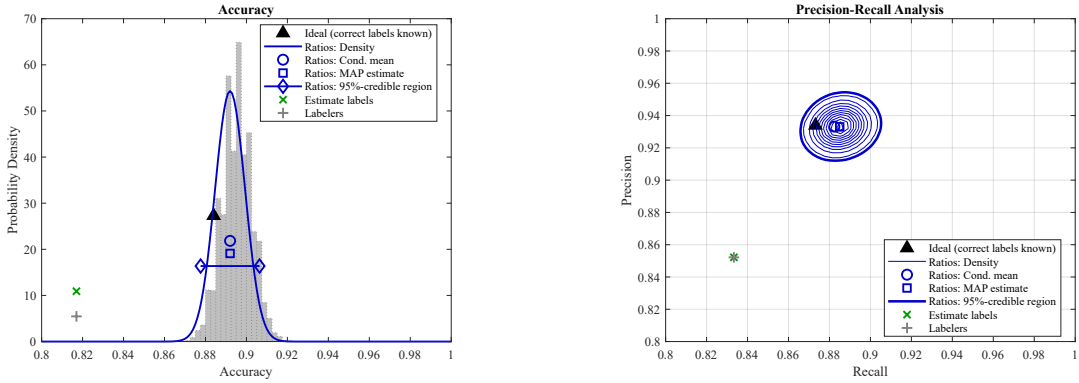


Figure 17: Testing example for a single labeler ($T = 1$) when a constant labeling-error probability $\varepsilon_0 = 0.1$ is assumed for all samples. The mean and median of the labelers' metrics are not shown because $T = 1$.

which frequently yield errors in excess of 0.100, 0.200, or more—unacceptable given that the metrics lie in $[0, 1]$. The empirical Bayes approach is clearly more appropriate than the fully Bayesian one.

5.2.5 Single Labeler and Constant Labeling-Error Probability

The testing methods can also be useful if there is a single labeler ($T = 1$), and one merely wants to know how performance would be affected by some worst-case labeling-error probability ε_0. One can simply set $\delta_i \equiv 0$ and $\phi_t \equiv C\varepsilon_0/(C-1)$ in (60) and apply the methods.

Figure 17 shows an example for $T = 1$ and $\varepsilon_0 = 0.1$; other simulation settings were $N = 10^3$, $\pi(1) = 0.6$, $\left(p_{\mathrm{D}}^{\mathrm{des}},\ p_{\mathrm{FA}}^{\mathrm{des}}\right) = (0.90, 0.10)$, $\delta_i \equiv 0$, and $\phi_t \equiv 0.2$. MMSE testing produces accurate estimates, and its estimated posteriors and credible regions allow one to understand the possible variability caused by the assumed labeling-error probability. For $0 < \varepsilon_0 < 1/2$, the estimated correct label $\check{y}_i$ is identically equal to the noisy label $z_{i,1}$, so the markers for the estimated correct labels and the labelers' labels lie in the same location. These suboptimal estimates are much less accurate than those from the empirical Bayes method.

5.2.6 Small Sample Size

The approximations behind MMSE testing are driven by the CLT, so they should hold for small N as long as $\hat{N}_1$ and $N - \hat{N}_1$ are greater than or equal to thirty. We conducted another simulation with $N = 70$, which produced $\hat{N}_1 = 37$ and $N - \hat{N}_1 = 33$; other simulation settings were $T = 3$, $\pi(1) = 0.5$, $\left(p_{\mathrm{D}}^{\mathrm{des}},\ p_{\mathrm{FA}}^{\mathrm{des}}\right) = (0.85, 0.20)$, $\delta_i \sim \mathrm{Beta}(1, 2)$, $\forall i$, and $\phi_t \sim \mathcal{U}(0.2, 0.5)$, $\forall t$.

Figure 18 displays results for P_{FA} and $(P_{\mathrm{D}}, P_{\mathrm{FA}})$. The posteriors are clearly non-Gaussian, and the 95%-credible regions are large because of the small sample size, but they contain the ideal metrics. The suboptimal methods again provide less accurate estimates.

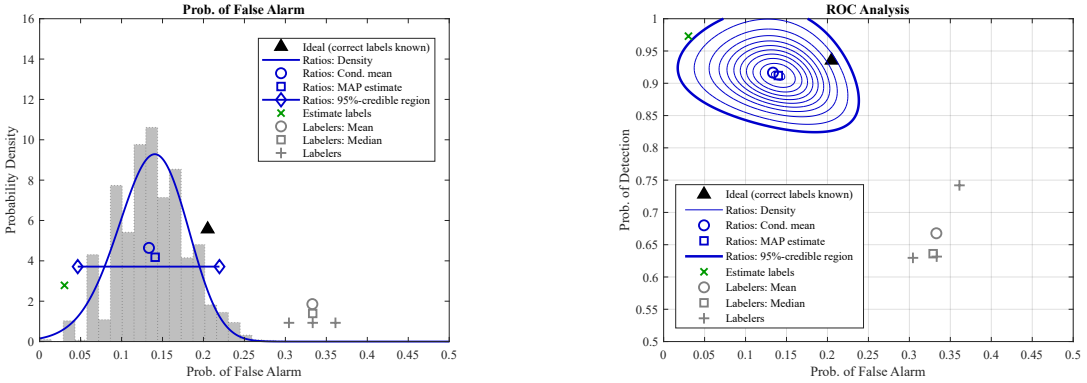


Figure 18: Testing example for small sample size ($N = 70$), with $\hat{N}_1 = 37$.

5.3 Example of Testing: Multi-Class Classification

For the multi-class testing methods of Section 2.7, we present an example with $C = 4$, $N = 2000$, $\boldsymbol{\pi} = (0.2, 0.3, 0.1, 0.4)$,

$$\mathbf{K}^{\text{des}} = \begin{bmatrix} 0.75 & 0.08 & 0.10 & 0.07 \\ 0.10 & 0.65 & 0.12 & 0.13 \\ 0.04 & 0.06 & 0.80 & 0.10 \\ 0.10 & 0.05 & 0.05 & 0.80 \end{bmatrix},$$

$T = 5$, $\delta_i \equiv 0$, $\forall i$, and $\phi_t \sim \mathcal{U}(0, 0.4)$, $\forall t$. MMSE testing with the empirical Bayes sampling method (Algorithm 4) used $M = 2500 \times C = 10^4$. We also extended the method that estimates the correct labels (Algorithm 3) to multi-class classification. Both algorithms converged after six iterations. For the individual labelers, we computed their confusion matrices, scaled the t^{th} labeler's confusion matrix by $N/(\text{no. of labels from } t^{\text{th}} \text{ labeler})$, and calculated the mean and multi-dimensional median of the scaled matrices. Apart from the ideal case, *testing did not involve the correct labels.*

First, Table 13 compares the estimates of the accuracy RV for the different estimation methods. MMSE testing produced the most accurate estimates, and the ideal accuracy value lies squarely inside the 95%-credible region. Estimating the correct labels yielded a reasonably good estimate of accuracy but again without a sense of uncertainty. Using the individual labelers' labels gave accuracies of 0.6586, 0.5667, 0.6397, 0.6698, and 0.5504; taking the mean or median of these values produces highly inaccurate estimates of the accuracy.

Second, Table 14 shows the ideal confusion matrix and the estimated confusion matrices from the different techniques. Ten matrix elements from MMSE testing are closest to the corresponding elements in the ideal confusion matrix, six from the estimation of correct labels are closest, one from the labelers' mean is closest, and one from the labelers' median is closest. (There were two ties, so these numbers sum to eighteen rather than sixteen.)

Finally, Table 15 shows the 95%-credible regions of the confusion matrix elements from MMSE testing. Every credible region contains the corresponding element in the ideal confusion matrix.

Estimation Method		ACC
Ideal (correct labels known)		0.7350
MMSE testing: Estimate conditional confusion matrix via sampling	Conditional mean	**0.7355**
	MAP estimate	**0.7355**
	95%-credible region	**(0.7282, 0.7427)**
Estimate correct labels		0.7390
Combine metrics from each labeler	Mean	0.6170
	Median	0.6397

Table 13: Estimated accuracy for 4-class classification example. Boldface indicates an estimate that was closest to the ideal accuracy or a credible region that contained the ideal accuracy.

Ideal (Correct) Labels Known)		Predicted Label			
		0	1	2	3
Correct Label	0	290	40	40	39
	1	58	391	68	68
	2	8	10	168	27
	3	93	43	36	621

MMSE Testing		Predicted Label			
		0	1	2	3
Correct Label	0	284.8	**41.3**	**36.6**	33.9
	1	63.7	**391.7**	67.0	67.6
	2	**7.5**	**7.3**	**169.8**	**29.0**
	3	**93.0**	**43.6**	38.7	**624.6**

Est. Correct Labels		Predicted Label			
		0	1	2	3
Correct Labels	0	**288**	43	35	**37**
	1	61	395	70	**68**
	2	7	4	170	**25**
	3	**93**	42	**37**	625

Mean of Labelers		Predicted Label			
		0	1	2	3
Correct Label	0	238.9	60.7	65.3	54.2
	1	**60.7**	335.5	66.3	86.3
	2	36.7	54.2	130.1	64.8
	3	76.6	91.3	48.8	529.6

Median of Labelers		Predicted Label			
		0	1	2	3
Correct Label	0	254.1	57.4	46.8	56.2
	1	71.1	343.4	**67.2**	86.9
	2	26.1	35.9	145.9	62.3
	3	90.6	65.8	46.7	543.6

Table 14: Ideal and estimated confusion matrices for 4-class classification example. Boldface indicates that the element was closest to the corresponding element in the ideal confusion matrix.

95%-Credible Regions		Predicted Label			
		0	1	2	3
Correct Label	0	(277.72, 291.95)	(35.80, 46.79)	(32.27, 40.88)	(28.54, 39.22)
	1	(57.75, 69.56)	(384.72, 398.77)	(61.51, 72.46)	(60.74, 74.43)
	2	(4.79, 10.23)	(3.97, 10.67)	(163.73, 175.83)	(24.03, 33.93)
	3	(86.74, 99.25)	(38.47, 48.82)	(34.34, 42.99)	(615.90, 633.21)

Table 15: 95%-credible regions for individual elements of the confusion matrix estimated by MMSE testing (Algorithm 4) for 4-class classification example.

5.4 Example of Training: Logistic Regression

This section uses logistic regression to illustrate the training approaches in Section 3. We use the Ionosphere binary-classification data set from the UCI Machine Learning Repository (see Dua and Graff, 2017), which contains $N = 351$ samples, each consisting of 34 real-valued features. Class 0 corresponds to a good radar return and class 1 to a bad radar return. Ionosphere contains 126 bad radar returns, so $\pi(1) = 0.359$. We employ 75%–25% stratified hold-out validation since multi-fold cross-validation produced cluttered plots that were too difficult to read. In practice, one could use cross-validation, of course.

Training and testing were conducted for $T = 1$, 5, 9, and 13. The data set provides the correct labels; noisy labels were simulated as in Section 5.1.2; hence, the noisy-label RVs are conditionally independent as in (58), $\psi_{i,t} = (\delta_i, \phi_t)$, and (27) reduces to (61). The settings were $\delta_i \sim \text{Beta}(1, 5)$, $\forall i$; $\phi_t \sim \mathcal{U}(0, 0.5)$, $\forall t$; $\eta_1 \equiv 1$ to force the first labeler to label every sample; and $\eta_t \sim \mathcal{U}(0.33, 1)$, $t \in \mathcal{T} \setminus \{1\}$. The sample-difficulty realization was $\boldsymbol{\delta} = (0.367, 0.524, 0.115, 0.181, 0.021, \ldots, 0.154)$, and the labeler-fallibility realization for $T = 13$ was $\boldsymbol{\phi} = (0.079,\ 0.440,\ 0.137,\ 0.207,\ 0.148,\ 0.314,\ 0.290,\ 0.300,\ 0.133,\ 0.142,\ 0.127,\ 0.164,\ 0.072)$. For each value of T, $\underline{\boldsymbol{z}}$ consisted of the noisy labels for labeler indexes 1 through T. In fact, the introductory example in Table 1 is an excerpt of $\underline{\boldsymbol{z}}$ for $T = 5$.

We consider five classifiers. The *ideal* classifier performs conventional ML training with $\{\underline{\boldsymbol{x}}, \boldsymbol{y}\}$ and is included for reference. The *ML-optimal* classifier performs ML training given $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$ according to part 1a of the unified view in Section 3.2, and primary term (32) from Section 3.3.1; details appear in Appendix F.1. The resulting objective function may no longer be convex, but gradient descent can still be used to find a local optimum. The *MMSE-optimal* classifier uses MMSE training according to part 2 of the unified view, primary term (40) from Section 3.4.1, and gradient components in (50) of Section 3.4.3; details are in Appendix F.2. The suboptimal, infrastructure-compatible *label-estimation* and *voting* classifiers described in Section 3.7 are also included. We omitted sample replication since it is essentially a quantized form of training with (32). For all classifier training, we used L_2 regularization and the standard Broyden-Fletcher-Goldfarb-Shanno method.

All testing metrics were calculated using the empirical Bayes method of Algorithm 1, so, except for the ideal classifier, *no correct labels were used during training or testing.* Although training could be optimal or suboptimal, testing always applied the same technique. For each training method, the regularization weight λ was swept over $\{0.5, 1.0, \ldots, 10.0\}$, producing twenty trained models. For the ideal classifier, we selected the model with the

largest ideal area under the ROC curve on the held-out testing set. For the other classifiers, we estimated the ROC curve on the testing set (explained shortly in Section 5.4.2) and selected the model with the largest estimated area under the ROC curve.

5.4.1 Fixed Decision Threshold

A trained logistic regression model computes $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) \in [0,1]$ as its estimate of $p(y|\boldsymbol{x};\boldsymbol{\theta})$ and compares it against a threshold τ; the predicted label $\hat{y}$ is 1 if $\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) > \tau$ and 0 otherwise. This section presents results for the single default threshold $\tau = 1/2$.

Figure 19 displays P-R analysis plots for the held-out testing set as T increases. For the classifiers trained with noisy labels, contours show the estimated joint posteriors of $(PREC, REC)$, circles show the conditional means, and solid lines indicate the 95%-credible regions. Normally, the correct labels $\boldsymbol{y}$ would not be available, but since we have the luxury of knowing them, inverted triangles mark each classifier's actual performance. The upright, solid black triangle shows the ideal classifier's operating point, which is identical for all values of T.

With $T = 1$, $\underline{\boldsymbol{z}}$ conveys little information about $\boldsymbol{y}$, so the posteriors are spread out, and the credible regions are quite large. For each training method, the conditional mean and actual operating point are not very close together, but the actual operating point lies within the credible region. The suboptimal methods outperform the MMSE-optimal and ML-optimal classifiers for this case, but the credible regions overlap so much that one could not make this conclusion without access to the correct labels. This plot also demonstrates that our training and testing approaches are applicable even for a single, imperfect labeler.

When $T = 5$, more information about $\boldsymbol{y}$ is available from $\underline{\boldsymbol{z}}$, so the credible regions become smaller, and the conditional means become much closer to the actual operating points for all classifiers. The ML-optimal classifier outperforms the ones that used suboptimal training, and from the posteriors, one could reasonably expect it to do so. It also happens to outperform the ideal classifier. The MMSE-optimal classifier performs comparably to the label-estimation classifier.

For $T = 9$, the training methods provide similar estimated precisions, with the ML-optimal classifier achieving greater recall. However, the posteriors overlap substantially, so one could not claim this without access to the correct labels. The ML-optimal classifier again slightly outperforms the ideal one. The MMSE-optimal and voting-trained classifier have identical performance.

By the time $T = 13$, the posteriors and credible regions are much tighter, and the conditional means give quite accurate estimates of the actual operating points. The ML-optimal classifier and the classifier trained with label estimation slightly outperform voting training, and their actual operating points coincide with that of the ideal classifier. For this choice of the threshold τ, the MMSE-optimal classifier has the lowest recall. In the next sections, the threshold is varied, and the MMSE-optimal classifier is seen to have competitive performance.

5.4.2 Performance Curves

It is also common practice to sweep the threshold τ over its range of possible values to obtain ROC or P-R *curves*. Figure 20 displays *estimated* ROC curves, which are the conditional

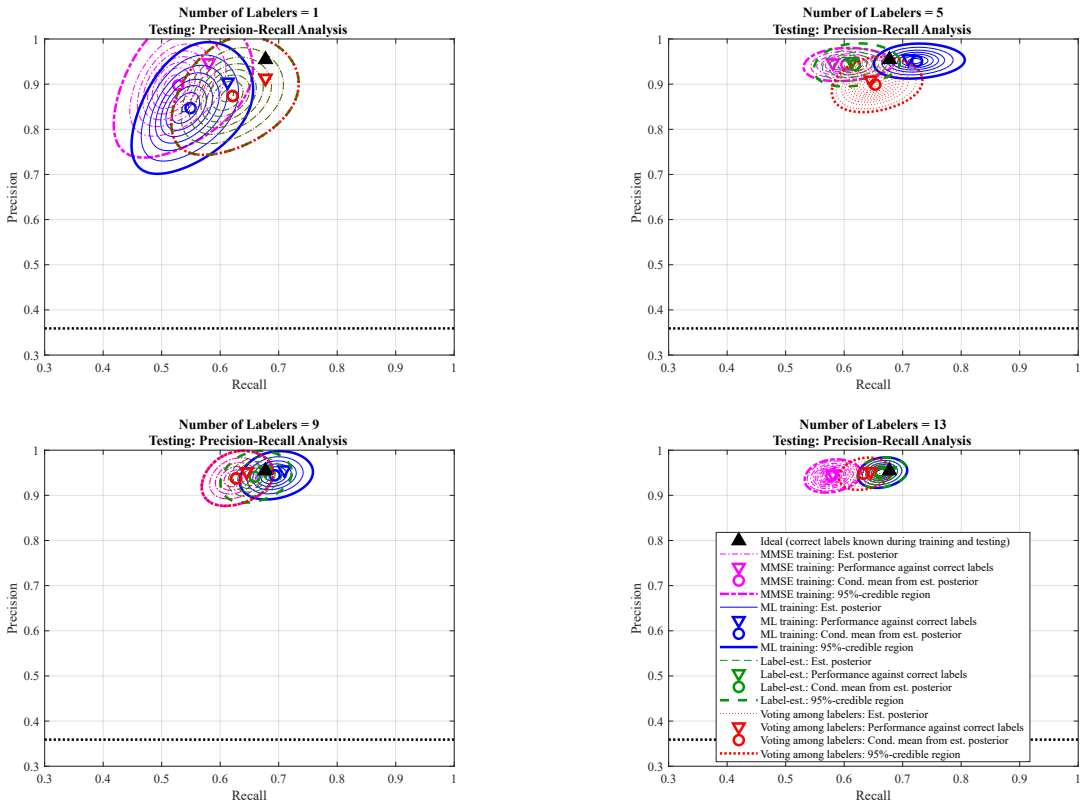


Figure 19: Training example with different numbers of labelers T: Estimated testing results from MMSE testing with Algorithm 1 on the held-out testing set. The chance line appears as a black dotted line.

means of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ from MMSE testing using Algorithm 1, for the different training methods and numbers of labelers. The estimated performance improves as T increases, although it does not change much beyond $T = 5$. When $T = 1$, it is difficult to estimate performance, which results in the jagged estimated ROC curves. As T increases, the estimated ROC curves begin to display the stepwise shape of the ideal ROC curve.

For each value of T, all of the training methods perform comparably. This behavior suggests that the regularized suboptimal training methods offer practical alternatives to training methods that are optimal according to the unified view. This observation differs from the experiments on testing from Sections 5.2.1, 5.2.4, and 5.2.5, where MMSE testing (Algorithms 1 and 2) outperformed the method of estimating the correct labels (Algorithm 3). We posit that this difference reflects the inherently different goals of training and testing and the presence or absence of regularization. The goal of training is to learn a predictive model that generalizes well to *out-of-sample* data beyond the training set $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. Therefore, training includes regularization, which helps a suboptimal training method compensate for its inferior estimation ability. In contrast, the goal of testing is to obtain the *in-sample* metrics for the testing set $\{\hat{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. Regularization is not called for in this case, and MMSE testing can provide much better estimation performance over suboptimal testing methods.

Figure 21 displays the actual ROC curves calculated against the correct labels. These curves would not be available in practice if truthing issues are present. They show that the training methods can achieve performance similar to the ideal case, and a comparison with Figure 20 shows that the estimated ROC curves are reasonably good even when $T = 1$, and they are very good for $T = 9$ or 13.

5.4.3 Performance Curve Posteriors

MMSE testing allows us to estimate the joint posterior of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ or $(PREC, REC)$. We can average the posteriors over all threshold values to obtain the posterior of a ROC or P-R curve—an important capability when truthing issues are present. Figure 22 shows the posteriors of the ROC curves for the ML-optimal classifier as T varies. Posterior values over the range $[10^{-3}, 10^{4}]$ are shown using heat maps with a base-10 logarithmic color scale. The figure also displays the estimated and actual ROC curves for this training method. We could also compute credible regions for the curves, but we do not show them to avoid cluttering the figure. Figure 19 already demonstrated the good containment of the credible regions.

Similarly, Figure 23 shows P-R curve posteriors for the MMSE-optimal classifier. When $\tau > 1$, the classifier predicts $\hat{y}_i = 0$, $\forall i$, so recall is zero but precision is undefined, and none of the curves show a point when $rec = 0$.

The figures also reveal another benefit of MMSE testing. Although actual performance is similar to the ideal case when $T = 1$ or 5, the estimated curves display some large deviations from the actual ones, and the posteriors have broad support, which indicates substantial uncertainty about performance. For $T = 9$ or 13, the estimated curves coincide closely with the actual ones, and the posteriors have more concentrated support, which reflects less uncertainty about performance. In the presence of truthing issues, the actual

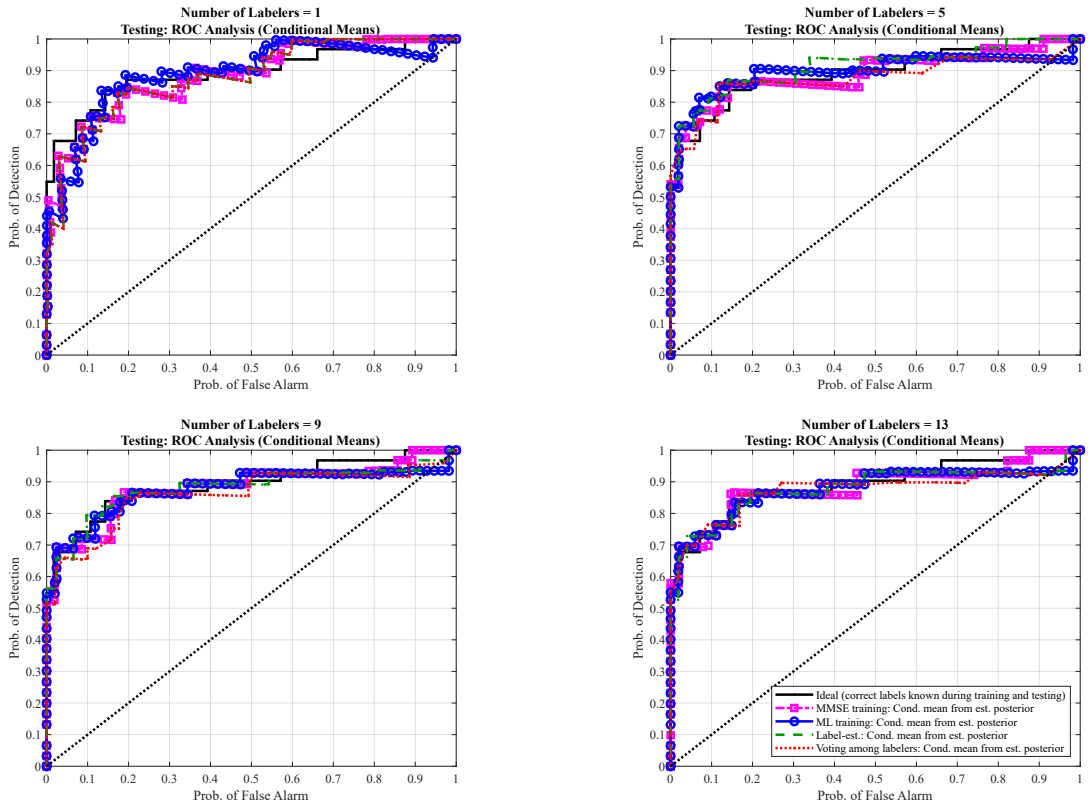


Figure 20: Training example with different numbers of labelers T: Estimated ROC curves from MMSE testing with Algorithm 1 on the held-out testing set. The chance line appears as a black dotted line.

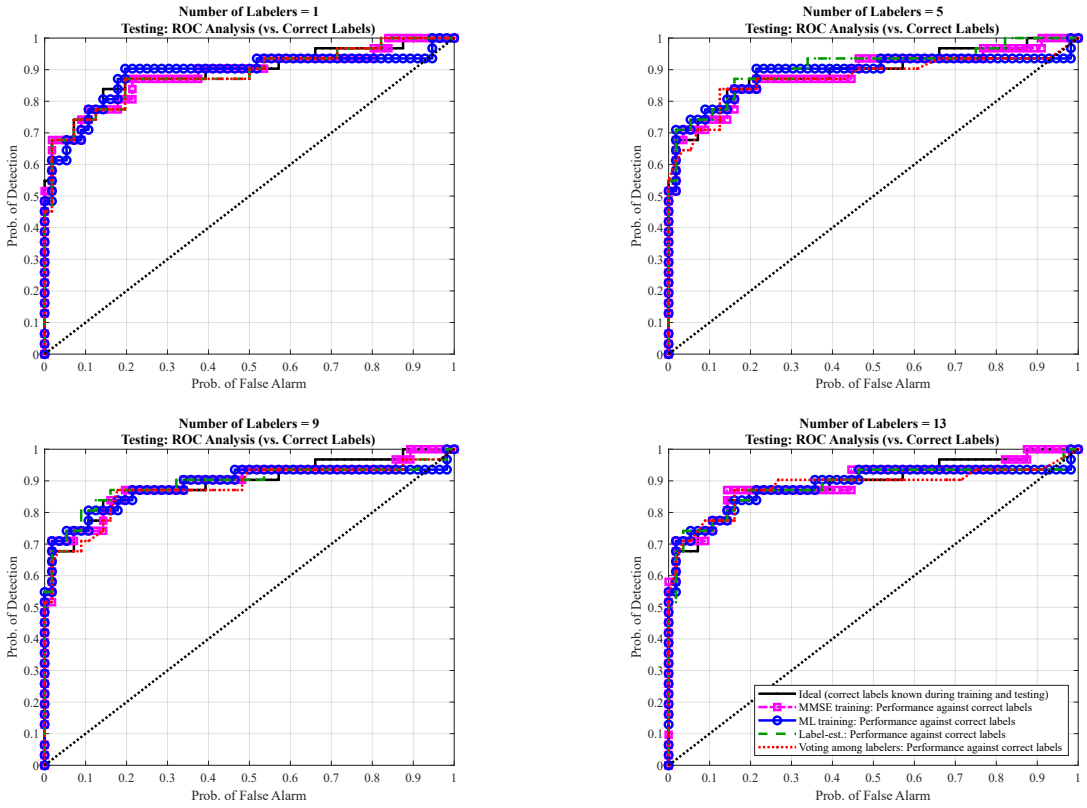


Figure 21: Training example with different numbers of labelers T: Actual ROC curves on the held-out testing set. The chance line appears as a black dotted line.

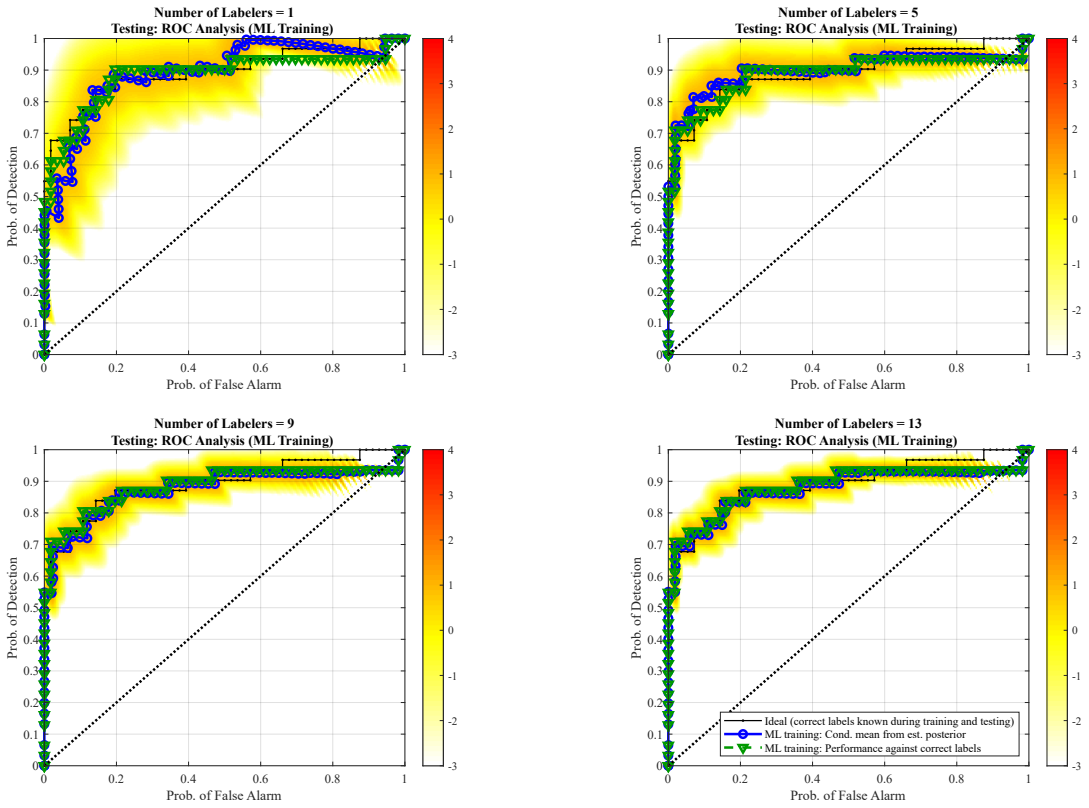


Figure 22: Training example with different numbers of labelers T for ML-trained classifiers: Estimated ROC curve posteriors from Algorithm 1 on the held-out testing set. The posteriors appear as heat maps with a base-10 logarithmic color scale; the color bar ticks correspond to the exponents of the scale. The chance line appears as a black dotted line.

performance curve will not be available, but the support of the performance curve posterior can help one understand the potential variability in performance that might occur.

5.5 Examples of Equivalent Mutual Information

In Section 4, the use of mutual information as a basis for comparing different combinations of labelers implied that multiple mediocre labelers could be as informative as a single expert labeler. To check the implication, we revisited the Ionosphere data set and simulated noisy labels with the model of (57), both for a single good labeler with $\varepsilon' = 0.05$ and for nine mediocre labelers, each with $\varepsilon = 0.25$. Every labeler provided a noisy label for every sample. Then $I(Z;Y|\varepsilon' = 0.05) = 0.667$ bits, and $I(\boldsymbol{Z};Y|T = 9, \varepsilon = 0.25) = 0.758$ bits. As in Section 5.4, we trained ML-optimal and MMSE-optimal classifiers using regularized logistic regression and estimated the testing metric RVs with Algorithm 1 for MMSE testing.

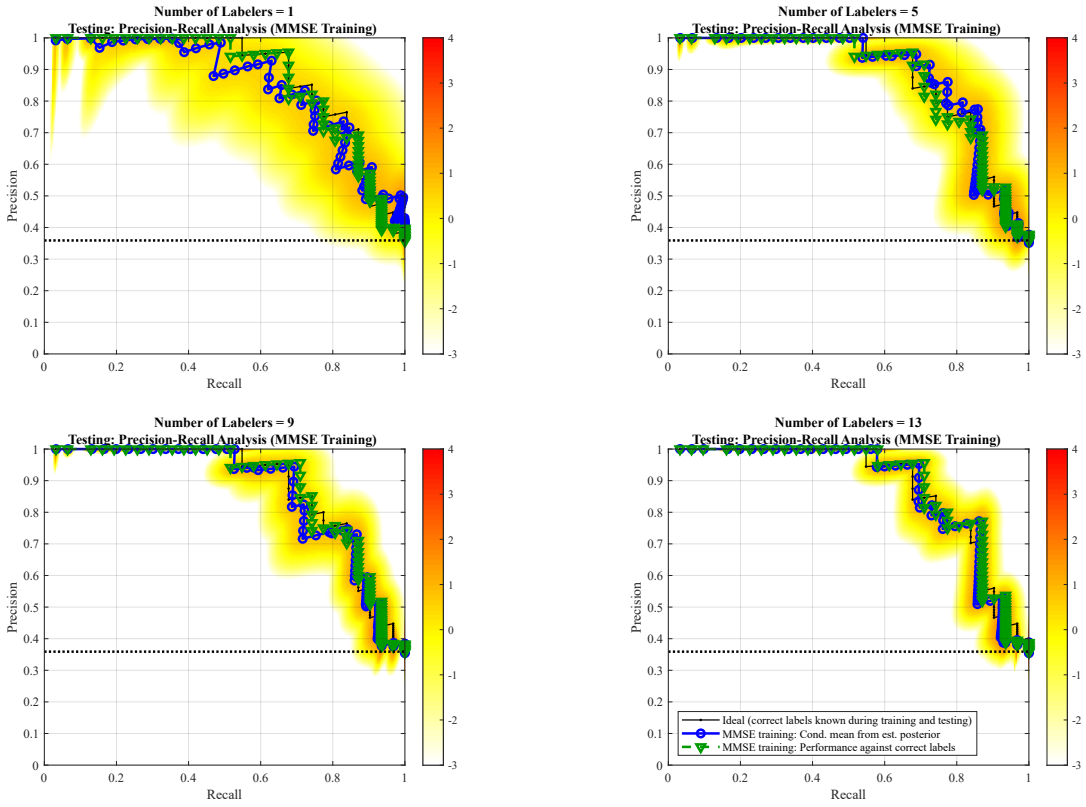


Figure 23: Training example with different numbers of labelers T for MMSE-trained classifiers: Estimated P-R curve posteriors from Algorithm 1 on the held-out testing set. The posteriors appear as heat maps with a base-10 logarithmic color scale; the color bar ticks correspond to the exponents of the scale. The chance line appears as a black dotted line.

Figure 24 shows ROC analysis plots for the held-out testing set. The results show that the classifier trained with the mediocre labelers' noisy labels performs better than the one trained with the single good labeler's noisy labels, which confirms the implication.

As an extreme example, we simulated a single expert labeler with $\varepsilon' = 0.01$ and 399 poor labelers, each with $\varepsilon = 0.45$, so $I(Z;Y|\varepsilon' = 0.01) = 0.863$ bits and $I(\boldsymbol{Z};Y|T = 399, \varepsilon = 0.45) = 0.859$ bits. The estimated P-R curve posteriors appear in Figure 25; when $\tau > 1$, both classifiers predict $\hat{y}_i = 0$, $\forall i$, so recall is zero but precision is undefined, and no points for $rec = 0$ are plotted. The curves indicate comparable performance and again confirm the implication.

6. Summary, Conclusions, and Future Directions

In supervised classification, a number of truthing issues may arise: noisy labels; missing labels; multiple, conflicting labels for the same sample; and different combinations of labelers for different samples. This situation involves three components, each of which requires a model: truthing warrants a noisy-label model, training learns a predictive model, and testing calls for a testing model. We did not study the problem of formulating and learning a noisy-label model, which is the subject of much of the related work. Instead, we concentrated on testing and training, and we began by assuming that a good noisy-label model $p(\boldsymbol{z}|y, \boldsymbol{\psi})\pi(y)$ was available, which makes our work compatible with and complementary to the related work. Our methods support models with dependent labelers.

6.1 Summary and Conclusions

By applying principles from Bayesian estimation theory, we succeeded in obtaining some promising and insightful answers to the questions posed in the introduction.

1. *How can one test a classifier in the presence of truthing issues?*

   Given noisy labels $\underline{\boldsymbol{z}}$, predicted labels $\hat{\boldsymbol{y}}$, noisy-label model parameters $\underline{\boldsymbol{\psi}}$, and class prior $\boldsymbol{\pi}$, we developed testing methods that are optimal: they estimate the metric RVs rather than the correct-label RVs, they fully exploit all available information, and they optimize a well-defined criterion (MMSE). Our approach is completely separate from training and applicable beyond the realm of machine learning. For example, it could be used to reconcile diagnoses made by clinicians or categorizations assigned by scientists.

   To arrive at the methods, we proposed a novel testing model (7), and we used it to derive approximate marginal posteriors for several scalar metrics, as well as joint posteriors for ROC and P-R analysis. We then introduced *MMSE testing* and developed empirical Bayes algorithms (Algorithms 1 and 2) for iteratively finding the MMSE estimate of the testing-model parameters from $\{\underline{\boldsymbol{z}}, \hat{\boldsymbol{y}}, \underline{\boldsymbol{\psi}}, \boldsymbol{\pi}\}$. After estimating the parameters, we calculated Bayesian optimal estimates (MMSE or MAP point estimates, or credible regions) of the metric RVs. Finally, we extended the approach to multi-class classification.

   In our experiments, MMSE testing provided excellent estimates of many binary-classification metrics. Their estimation errors were an order of magnitude smaller

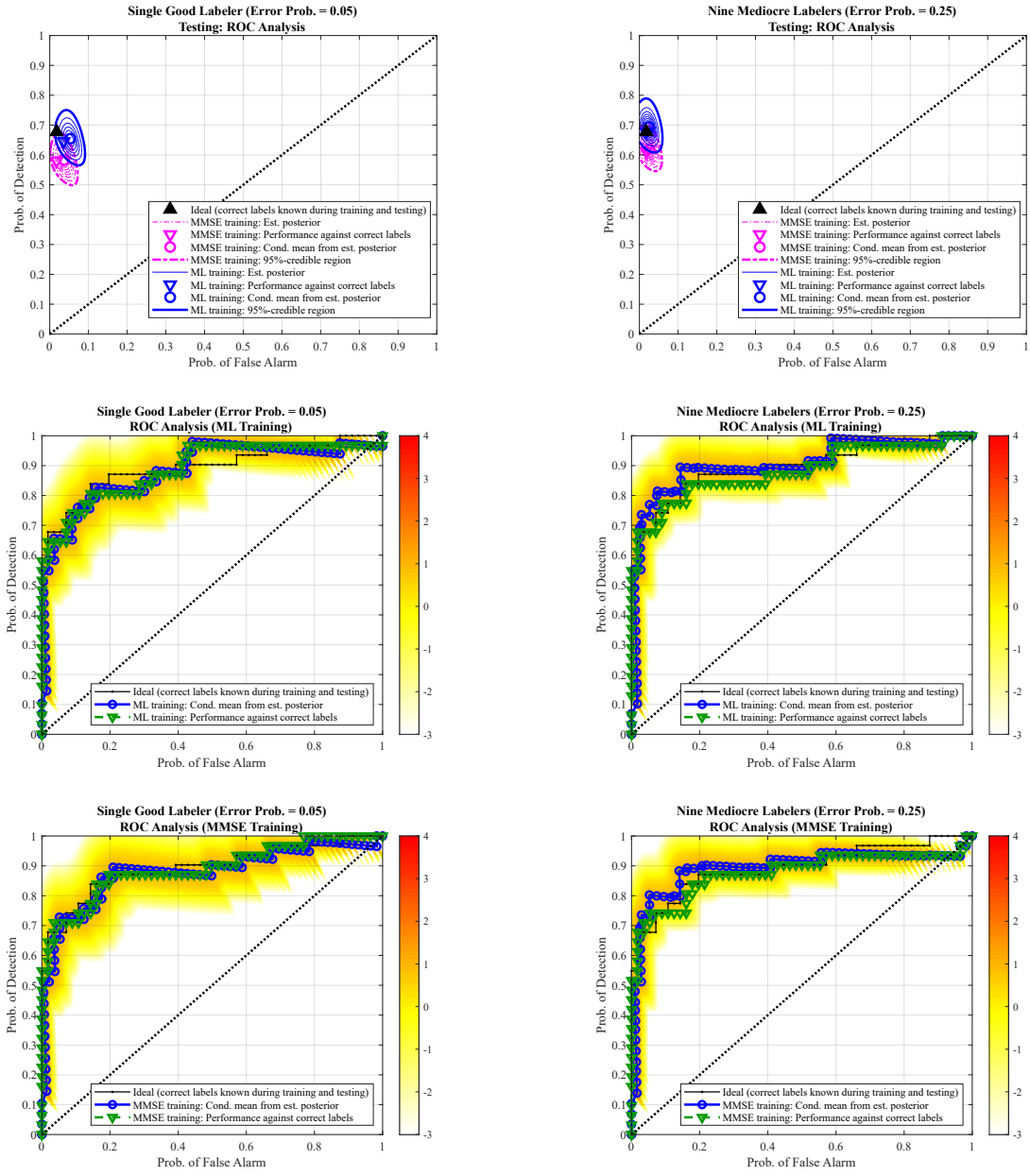


Figure 24: ROC analysis for a single good labeler (left) and multiple mediocre labelers (right). The upper graphs show performance for the single threshold $\tau = 1/2$. The middle graphs show the estimated ROC curve posteriors for ML-trained classifiers. The lower graphs show the estimated ROC curve posteriors for MMSE-trained classifiers. The posteriors appear as heat maps with a base-10 logarithmic color scale; the color bar ticks correspond to the exponents of the scale. The chance line appears as a black dotted line.

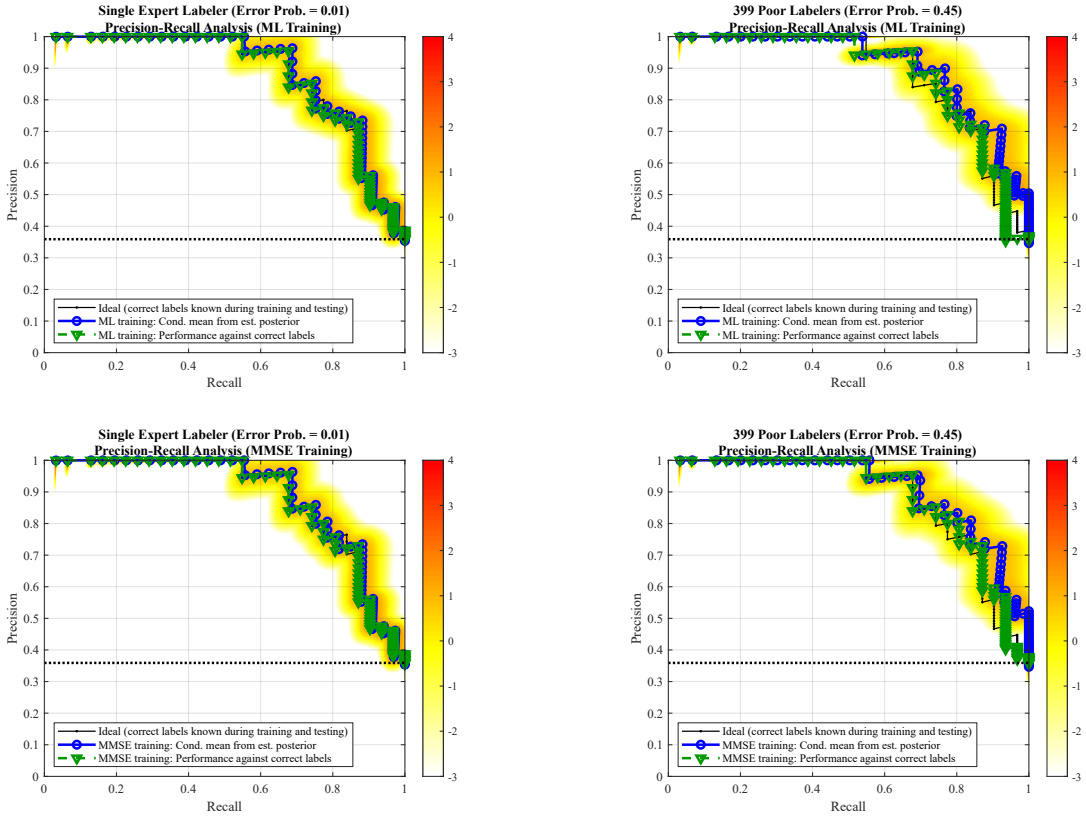


Figure 25: Estimated P-R curve posteriors for a single expert labeler (left) and many poor labelers (right). The upper graphs show results for ML-trained classifiers; the lower graphs show results for MMSE-trained classifiers. The posteriors appear as heat maps with a base-10 logarithmic color scale; the color bar ticks correspond to the exponents of the scale. The chance line appears as a black dotted line.

than those for suboptimal methods like estimating the correct labels or averaging the metrics from individual labelers. For multi-class classification, MMSE testing outperformed the suboptimal methods at estimating accuracy and individual elements of the confusion matrix.

2. *How can one train a classifier in the presence of truthing issues?*

   With $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{\psi}}$, $\boldsymbol{\pi}$, and the feature vectors $\underline{\boldsymbol{x}}$ given, we presented a unified view of training that is elegant and intuitive. The unified view explains how to train a broad range of classifiers, and it organizes some of the related work. Each one of our training approaches is optimal: it employs an appropriate likelihood function, posterior, or estimator; it fully exploits the available information; and it optimizes a well-defined penalty or utility criterion. None of the approaches estimates the correct labels.

   For probabilistic (i.e., generative or discriminative) models with parameters $\boldsymbol{\theta}$, we showed that the optimization principle from ideal training can be retained. In ML training, the likelihood function $p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta})$ or $p(\boldsymbol{y},\underline{\boldsymbol{x}};\boldsymbol{\theta})$ is replaced with $p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta})$ or $p(\underline{\boldsymbol{z}},\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta})$, respectively. In MAP training, the posterior $p(\boldsymbol{\theta}|\boldsymbol{y},\underline{\boldsymbol{x}})$ is replaced by $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}},\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}})$. For non-probabilistic models, we proposed *MMSE training*, which retains the original loss function and minimizes the in-sample MMSE estimate of the empirical-risk RV. Some related work has proposed the same form of training but did not use estimation theory to motivate it. We discussed properties of the MMSE estimator and provided a condition for when the MMSE estimator is a consistent estimator.

   Experiments using binary logistic regression demonstrated the effectiveness of our training approach, as well as the competitiveness of some suboptimal methods, like estimating the correct labels or voting among labelers, which are compatible with existing machine-learning infrastructure. We reasoned that regularization helps the suboptimal methods compensate for their estimation deficiencies. The experiments employed our testing methods, thus demonstrating the feasibility of *training and testing with noisy labels only*. Moreover, they showed that our testing methods can provide *approximate posteriors and optimal estimates of ROC and P-R curves*.

3. *How can one compare different combinations of labelers with different abilities?*

   The noisy-labeling process can be viewed as a broadcast channel, so mutual information quantifies the amount of information about the correct label conveyed by a group of labelers, and it facilitates comparison between different combinations of labelers. As another basis for comparison, any combination of labelers can be represented as an equivalent single labeler with some corresponding error probability. This observation implies that *multiple mediocre labelers can convey information greater than or equal to that from a single expert labeler*.

   The preceding statement is theoretical; it does not explain *how* to extract this information in practice. Fortunately, our training and testing methods provide a way to do so. The work in this paper culminated in training and testing experiments that confirmed the implication and showed that our methods can realize its benefits.

6.2 Expanded Workflow

Truthing issues are a reality in many applications. To address them, we advocate an expanded workflow that combines this work and the complementary related work. Training must learn *two* models: a noisy-label model and the desired predictive model. The models can be learned separately, using methods like those in Section 1.2.1 to learn the noisy-label model and then using the training techniques from Section 3 to learn the predictive model. Alternatively, the models can be learned jointly, using techniques like those by Raykar et al. (2010), Khetan et al. (2018), or Tanno et al. (2019). After both models have been learned, the predictive model can be tested using Algorithm 1, 2, or 4 from MMSE testing.

6.3 Future Directions

We close with a discussion of areas for future work.

6.3.1 Directly-Related Topics

A number of aspects of MMSE testing merit further study. First, Section 2.2 applied the CLT to the common RVs U and V, and subsequent manipulations produced the approximate posteriors of the metric RVs in Section 2.3; it would be useful to consider the case when one or both of the summations in (8) and (9) contain an insufficient number of terms to justify the CLT. Second, it would be fulfilling to obtain an expression for the joint posterior (17) of $(P_{\mathrm{D}}, P_{\mathrm{FA}})$ along the chance line $p_{\mathrm{D}} = p_{\mathrm{FA}}$. Third, closed-form expressions for the maximum of (17) and (18) would simplify MAP estimation of the ROC and P-R operating points. Fourth, Section 2.4.4 discussed convergence of the empirical Bayes methods, but more detailed study is called for. Finally, Section 2.7 examined multi-class classification, and it considered accuracy and individual elements of the confusion matrix. One could investigate other multi-class classification metrics, the joint distribution of the confusion matrix, the effect of a highly imbalanced class prior, and tactics when the number of classes is large.

Regarding training of a probabilistic predictive model, Section 3.3 showed that ML or MAP training could be modified for noisy labels. One could apply the forms given in Table 9 to adapt training from the ideal case to the case of truthing issues.

For MMSE training of non-probabilistic predictive models, Section 3.4.3 explained that its gradient is amenable to automatic differentiation (cf. (50)), so it would be exciting to see MMSE training applied to deep neural networks. Section 3.4.5 considered consistency of the MMSE estimator of the empirical-risk RV and gave the simplest of sufficient conditions. One could derive other conditions or bounds on the estimation error. One could also investigate conditions under which MMSE training preserves consistency of the ERM principle. The work of Khetan et al. (2018), Cid-Sueiro (2012), and Cid-Sueiro et al. (2014) could be useful in this regard.

The information-theoretic view was illustrated with the BSBC in Section 4.1. One could examine an asymmetric channel (i.e., a channel with different miss and false-alarm probabilities), a channel with dependent labelers, or a multi-class channel.

6.3.2 Learning and Labeler Allocation

This paper has assumed that the noisy-label model is known, but in many cases it must be learned. This requirement creates a variety of allocation problems. As an example of learning allocation, suppose that one has a small set $\{\underline{\boldsymbol{x}}', \boldsymbol{y}', \underline{\boldsymbol{z}}'\}$ with both correct and noisy labels and a much larger set $\{\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}\}$ with only noisy labels. How should one partition the sets for learning the noisy-label model, training the predictive model, and testing the learned predictive model?

Two examples of labeler allocation follow. Given a labeling budget and costs for labelers with different abilities, what is the most cost-effective way to acquire labels?[22] Similarly, given a set of samples with different labeling difficulties (e.g., images under a variety of lighting conditions), how should the labeling effort be distributed across the set?

6.3.3 Extension to Weak Supervision

The ideas in the related work and this paper could be adapted to other forms of supervised learning that involve noisy annotation, also known as weak supervision. Per Section 1.4, it would require three models: the usual predictive model, a noisy-annotation model for the imperfect annotation process, and a testing model that relates the predictions and noisy annotations. The noisy-annotation model is like the noisy-label model $p(\boldsymbol{z}|y)\pi(y)$ and provides estimated probabilities. It should generalize to unseen, out-of-sample realizations of $(\boldsymbol{Z}, Y)$, so it will be learned with machine-learning methods. The testing model is analogous to $p(\hat{y}, \boldsymbol{z}|y) = p(\hat{y}|y)p(\boldsymbol{z}|y)$ in (7); it characterizes in-sample performance on the testing set, so it will be learned with estimation-theoretic methods.

For training, many aspects of MMSE training are likely applicable to weak supervision. Section 3.4 and Appendix E.4 allow for a generic loss function and noisy-annotation model. If the annotation set $\mathcal{Y}$ is continuous rather than finite, then the summations over $\mathcal{Y}$ must be replaced by integrals (cf. (37), (40), (41), (43), (50)), and techniques for computing or approximating the integrals will be needed.

For testing of the predictive model, work tailored to the particular metrics will be required; the MMSE testing approach can provide a general strategy. Testing should follow the principle of estimating the in-sample metric RV rather than the correct annotation. First, one should identify a suitable testing model and parameters[23] (cf. Section 2.1) and express the metrics as RVs (Section 2.2). Second, one should obtain the posteriors of the parameters and metric RVs (Section 2.3); if the samples are independent, then the CLT may be helpful. Third, one should develop algorithms for estimating the parameters (Section 2.4). Finally, once the parameters have been estimated, the posteriors of the metric RVs can be used to find optimal estimates of the metrics (Section 2.5).

Acknowledgments

22. See Sheng et al. (2008), for example.
23. For binary classification, the OP parameters $(\tilde{p}_{\mathrm{D}}, \tilde{p}_{\mathrm{FA}})$ corresponded to probability of detection and probability of false alarm, which are standard metrics, but multi-class classification introduced the conditional confusion matrix, which is not commonly used.

The author gratefully thanks Rajmonda Caceres, John Holodnak, Jason Matterer, and Anna Yanchenko for their helpful comments and interest in this work, as well as Paul Monticciolo, Vijay Gadepally, Tim Dasey, and Jason Thornton for enabling this work. The author also thanks the Action Editors and anonymous reviewers for their careful attention, thoughtful questions, and helpful suggestions.

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001.

Appendix A. Metrics in Terms of Common RVs

For probability of false alarm, its RV form is

$$P_{\text{FA}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = 1 \text{ and } Y_i = 0)}{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = 0)}.$$

The numerator is

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = 1 \text{ and } Y_i = 0) &= \frac{1}{N}\sum_{i:\hat{y}_i=1} \mathbb{1}(Y_i = 0) \\
&= \frac{1}{N}\sum_{i:\hat{y}_i=1} (1 - \mathbb{1}(Y_i = 1)) \\
&= \frac{1}{N}\sum_{i:\hat{y}_i=1} 1 - \frac{1}{N}\sum_{i:\hat{y}_i=1} \mathbb{1}(Y_i = 1) \\
&= \frac{\hat{N}_1}{N} - U,
\end{aligned}$$

and the denominator is

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = 0) &= \frac{1}{N}\sum_{i=1}^{N} (1 - \mathbb{1}(Y_i = 1)) \\
&= 1 - \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = 1) \\
&= 1 - (U + V).
\end{aligned}$$

The RV form of accuracy is

$$\begin{aligned} ACC &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(Y_i = \hat{y}_i) \\ &= \frac{1}{N}\sum_{i:\hat{y}_i=0} \mathbb{1}(Y_i = 0) + \frac{1}{N}\sum_{i:\hat{y}_i=1} \mathbb{1}(Y_i = 1) \\ &= \frac{1}{N}\sum_{i:\hat{y}_i=0} (1 - \mathbb{1}(Y_i = 1)) + U \\ &= \frac{1}{N}(N - \hat{N}_1) - V + U \\ &= U - V - \hat{N}_1/N + 1. \end{aligned}$$

The RV form of precision is

$$\begin{aligned} PREC &= \frac{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = 1 \text{ and } Y_i = 1)}{\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = 1)} \\ &= \frac{\frac{1}{N}\sum_{i:\hat{y}_i=1} \mathbb{1}(Y_i = 1)}{\frac{1}{N}\hat{N}_1} \\ &= \frac{N}{\hat{N}_1} U. \end{aligned}$$

F_β is obtained by taking

$$F_\beta = (1+\beta^2)\frac{PREC \cdot REC}{\beta^2 PREC + REC},$$

substituting the expressions for $PREC$ and REC in Table 8, and performing a little algebra.

Appendix B. Ratios of Jointly Normal Random Variables

This section summarizes the procedure from Marsaglia (2006, 1965) for calculating the distribution or approximating the moments of the ratio of jointly normal RVs Z' and W'. Let $\mathrm{E}[Z'] = \mu_{Z'}$, $\mathrm{var}(Z') = \sigma^2_{Z'}$, $\mathrm{E}[W'] = \mu_{W'}$, $\mathrm{var}(W') = \sigma^2_{W'}$, and $\rho = \mathrm{cov}(Z', W')/\sigma_{Z'}\sigma_{W'}$. The density of Z'/W' is obtained as follows:

1. Let $h = \pm\sigma_{Z'}\sqrt{1-\rho^2}$; the sign of h will be determined momentarily. Also let $r = \sigma_{W'}/h$, $s = \rho\sigma_{Z'}/\sigma_{W'}$, $b = \mu_{W'}/\sigma_{W'}$, and $a = \pm(\mu_{Z'} - s\mu_{W'})/h$.

2. Choose the sign of h so that a and b have the same sign. Then the RV $T' = r(Z'/W' - s)$ can be expressed in the form $(a + X')/(b + Y')$, where X' and Y' are independent $\mathcal{N}(0,1)$ RVs. The density of T' is

$$p(t') = \frac{e^{-(a^2+b^2)/2}}{\pi(1+t'^2)}\left(1 + q(t')e^{(q(t'))^2/2}\int_0^{q(t')} e^{-x^2/2}\,dx,\right),$$

where $q(t') = (b + at')/\sqrt{(1+t'^2)}$, and the integral can be calculated using the error function $\mathrm{erf}(y) = (2/\sqrt{\pi})\int_0^y e^{-\tau^2}\,d\tau$.

Symbol	P_{D}, REC	P_{FA}	F_β
Z'	U	$(\hat{N}_1/N) - U$	$(1+\beta^2)U$
W'	$U+V$	$1-(U+V)$	$\beta^2(U+V) + \hat{N}_1/N$
$\mu_{Z'}$	μ_U	$(\hat{N}_1/N) - \mu_U$	$(1+\beta^2)\mu_U$
$\sigma^2_{Z'}$	σ^2_U	σ^2_U	$(1+\beta^2)^2\sigma^2_U$
$\mu_{W'}$	$\mu_U + \mu_V$	$1-(\mu_U+\mu_V)$	$\beta^2(\mu_U+\mu_V) + \hat{N}_1/N$
$\sigma^2_{W'}$	$\sigma^2_U + \sigma^2_V$	$\sigma^2_U + \sigma^2_V$	$\beta^4(\sigma^2_U + \sigma^2_V)$
$\mathrm{cov}(Z', W')$	σ^2_U	σ^2_U	$\beta^2(1+\beta^2)\sigma^2_U$
ρ	$\frac{\sigma_U}{\sqrt{\sigma^2_U+\sigma^2_V}}$	$\frac{\sigma_U}{\sqrt{\sigma^2_U+\sigma^2_V}}$	$\frac{\sigma_U}{\sqrt{\sigma^2_U+\sigma^2_V}}$

Table 16: Parameters for scalar metric RVs with posteriors equal to the ratio Z'/W' of jointly approximately normal RVs Z' and W'.

3. It follows that $Z'/W' = T'/r + s$, so $p_{Z'/W'}(\zeta) = |r| \cdot p_{T'}(r(\zeta - s))$.

The density $p(t')$ includes the standard Cauchy density $1/\big(\pi(1+t'^2)\big)$, so technically, the moments of T' of order greater than zero do not exist; i.e., $\int_{-\infty}^{\infty} t'^i p(t')\,dt'$ is infinite for $i \in \{1, 2, \ldots\}$. However, Marsaglia (2006, §4) points out that, in practice, one might be able to assume that the denominator $b + Y'$ approaches zero with negligible probability, enabling one to compute the moments (conditioned on this assumption).

For example, if $b = 4$, then $\Pr(b + Y' \le 0) = \Pr(Y' \le -4) \approx 3.17 \times 10^{-5}$. Marsaglia reports that, conditioned on $b > 4$ (or $Y' > -4$), the mean and variance of T' are approximately $\mu_{T'} = a/(1.01\,b - 0.2713)$ and $\sigma^2_{T'} = (a^2+1)/(b^2 + 0.108\,b - 3.795) - \mu^2_{T'}$.

Finally, Marsaglia also observes that, if $a < 2.256$ and $b > 4$, then T' can be reasonably approximated by a *normal* distribution.

Appendix C. Review of MMSE Estimation

Let $\boldsymbol{A}$ be the unobserved RV, $\boldsymbol{B}$ be the observed RV, and let $\boldsymbol{f}(\boldsymbol{A}) = \big[f_1(\boldsymbol{A}) \;\; \cdots \;\; f_D(\boldsymbol{A})\big]^{\mathrm{T}}$ be a D-dimensional vector of scalar functions of $\boldsymbol{A}$, with $\mathrm{E}[f_j^2(\boldsymbol{A})] < \infty$, $j = 1, \ldots, D$. Define an estimator of $\boldsymbol{f}(\boldsymbol{A})$ from $\boldsymbol{B}$ as $\boldsymbol{h}(\boldsymbol{B}) = \big[h_1(\boldsymbol{B}) \;\; \cdots \;\; h_D(\boldsymbol{B})\big]^{\mathrm{T}}$, and define the MSE of $\boldsymbol{h}(\boldsymbol{B})$ by $mse\big(\boldsymbol{h}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big) = \sum_{j=1}^{D} \mathrm{E}\big[\big(h_j(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)^2\big]$. The goal is to find the MMSE estimator $\boldsymbol{h}^{\mathrm{MMSE}} = \arg\min_{\boldsymbol{h}} mse\big(\boldsymbol{h}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big)$.

For Section 2.4.1, $\boldsymbol{A} = (P_{\mathrm{D}}, P_{\mathrm{FA}})$, $\boldsymbol{B} = (\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}})$, and $\boldsymbol{f}(\boldsymbol{A})$ is the identity function $\boldsymbol{f}(\boldsymbol{A}) \equiv \boldsymbol{A}$. The estimators are $h_1(\boldsymbol{B}) = h_{\mathrm{D}}\big(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}, \underline{\psi}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$ and $h_2(\boldsymbol{B}) = h_{\mathrm{FA}}\big(\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}, \underline{\psi}, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$, where $\underline{\psi}$, $\tilde{p}_{\mathrm{D}}^{(j-1)}$, and $\tilde{p}_{\mathrm{FA}}^{(j-1)}$ are non-random parameters. In Section 2.6.2 on fully Bayesian estimation of a metric RV like accuracy, $\boldsymbol{A} = ACC$, $\boldsymbol{B} = (\hat{\boldsymbol{Y}}, \underline{\boldsymbol{Z}})$, $\boldsymbol{f}(\boldsymbol{A})$ is the identity function, and $\underline{\psi}$ is a non-random parameter. For estimation of the empirical-risk RV in Section 3.4.1 and Appendix E.4, $\boldsymbol{A} = \boldsymbol{Y}$, $\boldsymbol{B} = \underline{\boldsymbol{Z}}$, $\boldsymbol{f}(\boldsymbol{A}) = J_{\mathrm{pri}}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \boldsymbol{Y}) = J(\boldsymbol{Y})$, and an estimator is $\boldsymbol{h}(\boldsymbol{B}) = \hat{J}(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}}, \underline{\psi}, \boldsymbol{\pi}) = \hat{J}(\underline{\boldsymbol{Z}})$. For estimating

the i^{th} loss-function RV in Appendix E.4, $\boldsymbol{A} = Y_i$, $\boldsymbol{B} = \boldsymbol{Z}_i$, $\boldsymbol{f}(\boldsymbol{A}) = L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}), Y_i) = \ell_i(Y_i)$, and an estimator is $\boldsymbol{h}(\boldsymbol{Z}_i) = \hat{\ell}_i(\boldsymbol{x}_i, \boldsymbol{\theta}, \boldsymbol{Z}_i) = \hat{\ell}_i(\boldsymbol{Z}_i)$.

C.1 Standard Results

1. The MMSE estimator is $\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}) = E[\boldsymbol{f}(\boldsymbol{A})|\boldsymbol{B}]$, or $h_d^{\text{MMSE}}(\boldsymbol{B}) = E[f_d(\boldsymbol{A})|\boldsymbol{B}]$, $d = 1, \ldots, D$; that is, *the MMSE estimator is the conditional mean.* The MMSE *estimator* is an RV because it is a function of $\boldsymbol{B}$. Given $\boldsymbol{B} = \boldsymbol{b}$, the MMSE *estimate* is the non-random quantity $\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{b}) = E[\boldsymbol{f}(\boldsymbol{A})|\boldsymbol{B} = \boldsymbol{b}]$, or $h_d^{\text{MMSE}}(\boldsymbol{b}) = E[f_d(\boldsymbol{A})|\boldsymbol{B} = \boldsymbol{b}]$, $d = 1, \ldots, D$. See (Van Trees, 1968, §2.4.1), (Papoulis, 1991, §7-5, §8-3), (Kay, 1993, §10.3), (Kamen and Su, 1999, Theorems 3.1, 3.4), (Oppenheim and Verghese, 2015, §8.1, §8.2).

2. The MMSE estimator is unbiased: $\text{E}[\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B})] = \text{E}[\boldsymbol{f}(\boldsymbol{A})]$; see (Papoulis, 1991, §7-4), (Kay, 1993, §11.6), (Kamen and Su, 1999, §3.3).

3. The estimation error $\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}) - \boldsymbol{f}(\boldsymbol{A})$ has the following properties.

   (a) The mean is zero: $\text{E}\big[\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}) - \boldsymbol{f}(\boldsymbol{A})\big] = \boldsymbol{0}$ from Result 2.

   (b) The total variance equals the MMSE, which *is* the MSE of the MMSE estimator: $\sum_{j=1}^{D} \text{var}\big(h_j^{\text{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big) = mse\big(\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big)$; see (Kay, 1993, §11.6).

   (c) (Orthogonality principle) The error is orthogonal to every function $\boldsymbol{w}(\boldsymbol{B}) = \begin{bmatrix} w_1(\boldsymbol{B}) & \cdots & w_D(\boldsymbol{B}) \end{bmatrix}^{\text{T}}$: $\text{E}\big[\big(\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}) - \boldsymbol{f}(\boldsymbol{A})\big)^{\text{T}} \boldsymbol{w}(\boldsymbol{B})\big] = 0$; see (Papoulis, 1991, §8-3), (Kamen and Su, 1999, Theorems 3.2, 3.3), (Oppenheim and Verghese, 2015, §8.2.1).

4. The MMSE estimator can be related to the law of total variance, which states that, for any RV G with finite variance, $\text{var}(G) = \text{E}[\text{var}(G|\boldsymbol{B})] + \text{var}(\text{E}[G|\boldsymbol{B}])$, and $\text{E}[\text{var}(G|\boldsymbol{B})]$ and $\text{var}(\text{E}[G|\boldsymbol{B}])$ are the unexplained and explained variances of G, respectively; see (Blitzstein and Hwang, 2019, §9.5).

   (a) The MMSE is equal to the total unexplained variance of $\boldsymbol{f}(\boldsymbol{A})$: $mse\big(\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big) = \sum_{j=1}^{D} \text{E}[\text{var}(f_j(\boldsymbol{A})|\boldsymbol{B})]$; see (Van Trees, 1968, §2.4), (Kay, 1993, §10.4).

   (b) The total variance of the MMSE estimator equals the total explained variance of $\boldsymbol{f}(\boldsymbol{A})$: $\sum_{j=1}^{D} \text{var}\big(h_j^{\text{MMSE}}(\boldsymbol{B})\big) = \sum_{j=1}^{D} \text{var}\big(\,\text{E}[f_j(\boldsymbol{A})|\boldsymbol{B}]\,\big)$, which means the MMSE estimator accounts for as much of the variation of $\boldsymbol{f}(\boldsymbol{A})$ as possible given $\boldsymbol{B}$.

   (c) Hence, the total variance of $\boldsymbol{f}(\boldsymbol{A})$ equals the sum of the MMSE (the unexplained variance) and the variance of the MMSE estimator (the explained variance):

$$\sum_{j=1}^{D} \text{var}(f_j(\boldsymbol{A})) = \underbrace{mse(\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A}))}_{\sum_{j=1}^{D} \text{E}[\,\text{var}(f_j(\boldsymbol{A})|\boldsymbol{B})\,]} + \underbrace{\sum_{j=1}^{D} \text{var}(h_j^{\text{MMSE}}(\boldsymbol{B}))}_{\sum_{j=1}^{D} \text{var}(\,\text{E}[f_j(\boldsymbol{A})|\boldsymbol{B}]\,)}.$$

5. Results 2 through 4 apply to the MMSE estimator. For the MMSE estimate given $\boldsymbol{B} = \boldsymbol{b}$, the MSE equals the sum of the conditional variances of the functions being estimated: $mse\big(\boldsymbol{h}^{\text{MMSE}}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big|\boldsymbol{B} = \boldsymbol{b}\big) = \sum_{j=1}^{D} \text{var}(f_j(\boldsymbol{A})|\boldsymbol{B} = \boldsymbol{b})$; see (Oppenheim and Verghese, 2015, §8.1).

C.2 Proofs and Derivations of Standard Results

The derivation of Result 1 assumes $\boldsymbol{B}$ is continuous but is easily modified if $\boldsymbol{B}$ is discrete. Expand the MSE as

$$\begin{aligned}
mse\big(\boldsymbol{h}(\boldsymbol{B}),\boldsymbol{f}(\boldsymbol{A})\big) &= \sum_{j=1}^{D}\big\{\mathrm{E}[h_j^2(\boldsymbol{B})] - 2\mathrm{E}[h_j(\boldsymbol{B})f_j(\boldsymbol{A})] + \mathrm{E}[f_j^2(\boldsymbol{A})]\big\} \\
&\overset{(a)}{=} \sum_{j=1}^{D}\big\{\mathrm{E}[h_j^2(\boldsymbol{B})] - 2\mathrm{E}\big[\,\mathrm{E}[h_j(\boldsymbol{B})f_j(\boldsymbol{A})|\boldsymbol{B}]\,\big] + \mathrm{E}[f_j^2(\boldsymbol{A})]\big\} \\
&= \sum_{j=1}^{D}\left\{\int p(\boldsymbol{b})h_j^2(\boldsymbol{b})\,d\boldsymbol{b} - 2\int p(\boldsymbol{b})E[h_j(\boldsymbol{B})f_j(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]\,d\boldsymbol{b} + \mathrm{E}[f_j^2(\boldsymbol{A})]\right\} \\
&\overset{(b)}{=} \sum_{j=1}^{D}\left\{\int p(\boldsymbol{b})h_j^2(\boldsymbol{b})\,d\boldsymbol{b} - 2\int p(\boldsymbol{b})h_j(\boldsymbol{b})E[f_j(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]\,d\boldsymbol{b} + \mathrm{E}[f_j^2(\boldsymbol{A})]\right\},
\end{aligned}$$

where (a) is from iterated expectations, and (b) is because $h_j(\boldsymbol{B})$ is non-random and equal to $h_j(\boldsymbol{b})$ when conditioned on $\boldsymbol{B}=\boldsymbol{b}$. For $d=1,\ldots,D$, take the partial derivative of the above equation with respect to h_d, set it equal to zero, and solve for h_d. Doing so gives

$$\begin{aligned}
\frac{\partial mse}{\partial h_d} &= 2\int p(\boldsymbol{b})h_d(\boldsymbol{b})\,d\boldsymbol{b} - 2\int p(\boldsymbol{b})E[f_d(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]\,d\boldsymbol{b}, \quad d=1,\ldots,D, \\
&= 2\int p(\boldsymbol{b})\big[h_d(\boldsymbol{b}) - E[f_d(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]\big]\,d\boldsymbol{b}, \quad d=1,\ldots,D.
\end{aligned}$$

Since $p(\boldsymbol{b})$ is a probability distribution, this expression is zero when $h_d(\boldsymbol{b}) = E[f_d(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]$ for any valid $\boldsymbol{b}$, which is just the definition of the conditional mean. Also, $\partial^2 mse/\partial h_d^2 = 2\int p(\boldsymbol{b})\,d\boldsymbol{b} = 2 > 0$, $d=1,\ldots,D$, so the solution is the unique minimum. Thus, the MMSE estimator is $\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}) = E[\boldsymbol{f}(\boldsymbol{A})|\boldsymbol{B}]$, or $h_d^{\mathrm{MMSE}}(\boldsymbol{B}) = E[f_d(\boldsymbol{A})|\boldsymbol{B}]$, $d=1,\ldots,D$; and the MMSE estimate given $\boldsymbol{B}=\boldsymbol{b}$ is $\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{b}) = E[\boldsymbol{f}(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]$, or $h_d^{\mathrm{MMSE}}(\boldsymbol{b}) = E[f_d(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]$, $d=1,\ldots,D$. This proves Result 1.

For Result 2, write $\mathrm{E}[\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B})] = \mathrm{E}\big[E[\boldsymbol{f}(\boldsymbol{A})|\boldsymbol{B}]\big] \overset{(a)}{=} \mathrm{E}[\boldsymbol{f}(\boldsymbol{A})]$, where (a) applies iterated expectations.

For Result 3a, unbiasedness means $\mathrm{E}\big[\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}) - \boldsymbol{f}(\boldsymbol{A})\big] = \boldsymbol{0}$. For Result 3b, the preceding relation means $\sum_{j=1}^{D}\mathrm{var}\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big) = \sum_{j=1}^{D}\mathrm{E}\big[\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)^2\big] = mse\big(\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}),\boldsymbol{f}(\boldsymbol{A})\big)$. To obtain Result 3c, for any valid j and $\boldsymbol{b}$, write $\mathrm{E}\big[(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A}))w_j(\boldsymbol{B})\big|\boldsymbol{B}=\boldsymbol{b}\big] = h_j^{\mathrm{MMSE}}(\boldsymbol{b})w_j(\boldsymbol{b}) - \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]w_j(\boldsymbol{b}) = \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]w_j(\boldsymbol{b}) - \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B}=\boldsymbol{b}]w_j(\boldsymbol{b}) = 0$. Then $\mathrm{E}\big[\big(\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}) - \boldsymbol{f}(\boldsymbol{A})\big)^{\mathrm{T}}\boldsymbol{w}(\boldsymbol{B})\big] = \sum_{j=1}^{D}\mathrm{E}\big[\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)w_j(\boldsymbol{B})\big] = \sum_{j=1}^{D}\mathrm{E}\big[\,\mathrm{E}\big[\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)w_j(\boldsymbol{B})\,\big|\,\boldsymbol{B}\big]\,\big] = \sum_{j=1}^{D}\int\mathrm{E}\big[\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)w_j(\boldsymbol{B})\big|\boldsymbol{B}=\boldsymbol{b}\big]p(\boldsymbol{b})\,d\boldsymbol{b} = \sum_{j=1}^{D}\int 0\cdot p(\boldsymbol{b})\,d\boldsymbol{b} = 0$.

For Result 4a, write $mse\big(\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}),\boldsymbol{f}(\boldsymbol{A})\big) \overset{(a)}{=} \sum_{j=1}^{D}\mathrm{E}\big[\mathrm{E}\big[\big(h_j^{\mathrm{MMSE}}(\boldsymbol{B}) - f_j(\boldsymbol{A})\big)^2\big|\boldsymbol{B}\big]\big] = \sum_{j=1}^{D}\mathrm{E}\big[\mathrm{E}\big[\big(f_j(\boldsymbol{A}) - \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B}]\big)^2\big|\boldsymbol{B}\big]\big] \overset{(b)}{=} \sum_{j=1}^{D}\mathrm{E}[\,\mathrm{var}(f_j(\boldsymbol{A})|\boldsymbol{B})\,]$, where (a) uses iterated expectations and (b) applies the definition of conditional variance. Result 4b is because

$h_j^{\mathrm{MMSE}}(\boldsymbol{B}) = E[f_j(\boldsymbol{A})|\boldsymbol{B}]$, so their variances are equal. Applying the law of total variance to each term in $\sum_{j=1}^{D} \mathrm{var}(f_j(\boldsymbol{A}))$ yields Result 4c.

For Result 5, conditioning the MSE on $\boldsymbol{B} = \boldsymbol{b}$ gives $mse\big(\boldsymbol{h}^{\mathrm{MMSE}}(\boldsymbol{B}), \boldsymbol{f}(\boldsymbol{A})\big|\boldsymbol{B} = \boldsymbol{b}\big) = \sum_{j=1}^{D} \mathrm{E}\big[\big(f_j(\boldsymbol{A}) - \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B}]\big)^2\big|\boldsymbol{B} = \boldsymbol{b}\big] = \sum_{j=1}^{D} \mathrm{E}\big[\big(f_j(\boldsymbol{A}) - \mathrm{E}[f_j(\boldsymbol{A})|\boldsymbol{B} = \boldsymbol{b}]\big)^2\big|\boldsymbol{B} = \boldsymbol{b}\big] \overset{(a)}{=} \sum_{j=1}^{D} \mathrm{var}(f_j(\boldsymbol{A})|\boldsymbol{B} = \boldsymbol{b})$, where (a) uses the definition of conditional variance given $\boldsymbol{B} = \boldsymbol{b}$.

Appendix D. Review of MPE and MAP Estimation

Additional derivations appear in Van Trees (1968, §2.4.1) and Kay (1998, Ch. 3). Let Y be a finite RV with domain $\mathcal{Y}$, and let $\boldsymbol{B}$ be the observed, discrete RV. Denote an estimator of Y given $\boldsymbol{B}$ as $h(\boldsymbol{B})$. The probability of error of $h(\boldsymbol{B})$ is $p_{\mathrm{error}}(h(\boldsymbol{B}), Y) = \Pr(h(\boldsymbol{B}) \neq Y) = \mathrm{E}_{p(y,\boldsymbol{b})}[\mathbb{1}(h(\boldsymbol{B}) \neq Y)]$. The goal is to find the MPE estimator, namely $h^{\mathrm{MPE}} = \arg\min_h p_{\mathrm{error}}(h(\boldsymbol{B}), Y)$.

For Section 2.6.1, $Y = Y_i$, and $\boldsymbol{B} = (\hat{Y}_i, \boldsymbol{Z}_i)$ with additional conditioning parameters $\big(\boldsymbol{\psi}_i, \tilde{p}_{\mathrm{D}}^{(j-1)}, \tilde{p}_{\mathrm{FA}}^{(j-1)}\big)$. For Section 3.7, $Y = Y_i$, and $\boldsymbol{B} = \boldsymbol{Z}_i$ with additional parameter $\boldsymbol{\psi}_i$.

Write $p_{\mathrm{error}}(h(\boldsymbol{B}), Y) = \mathrm{E}[1 - \mathbb{1}(h(\boldsymbol{B}) = Y)] = 1 - \mathrm{E}\big[\mathrm{E}[\mathbb{1}(h(\boldsymbol{B}) = Y) \,|\, \boldsymbol{B}]\big] = 1 - \sum_{\boldsymbol{b}} p(\boldsymbol{b}) \cdot \mathrm{E}[\mathbb{1}(h(\boldsymbol{B}) = Y) \,|\, \boldsymbol{B} = \boldsymbol{b}]$. The last form is minimized by maximizing the conditional expectation for each $\boldsymbol{b}$. Then $\mathrm{E}[\mathbb{1}(h(\boldsymbol{B}) = Y) \,|\, \boldsymbol{B} = \boldsymbol{b}] = \sum_{y \in \mathcal{Y}} p(y|\boldsymbol{b}) \mathbb{1}(h(\boldsymbol{b}) = y) = p_{Y|\boldsymbol{B}}(h(\boldsymbol{b}) \,|\, \boldsymbol{b})$, and the maximum occurs when $h(\boldsymbol{b})$ is the MAP estimate of Y given $\boldsymbol{B} = \boldsymbol{b}$; i.e., when $h(\boldsymbol{b}) = h^{\mathrm{MAP}}(\boldsymbol{b}) = \arg\max_{y \in \mathcal{Y}} p(y|\boldsymbol{b})$. Define the MAP estimator $h^{\mathrm{MAP}}(\boldsymbol{B})$ as the function of $\boldsymbol{B}$ that returns $h^{\mathrm{MAP}}(\boldsymbol{b})$ when $\boldsymbol{B} = \boldsymbol{b}$; then the MPE estimator is $h^{\mathrm{MPE}}(\boldsymbol{B}) = h^{\mathrm{MAP}}(\boldsymbol{B})$.

Appendix E. Training with Truthing Issues

E.1 Discriminative Model, Random Parameters

Ideal case: Find $\boldsymbol{\theta}$ to maximize the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}, \underline{\boldsymbol{x}})$, given by $p(\boldsymbol{\theta}|\boldsymbol{y}, \underline{\boldsymbol{x}}) \overset{(a)}{\propto} p(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}|\underline{\boldsymbol{x}}) \overset{(b)}{=} p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$, where (a) uses Bayes' rule, and (b) is from (29).

Truthing issues: Find $\boldsymbol{\theta}$ to maximize $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}})$, and write $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}) \overset{(a)}{\propto} p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}; \underline{\boldsymbol{\psi}}) \cdot p(\boldsymbol{\theta}|\underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}) \overset{(b)}{=} p(\boldsymbol{\theta}) \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i) p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$, where (a) uses Bayes' rule, and (b) applies (31).

E.2 Generative Model, Non-Random Parameters

Ideal case: Find $\boldsymbol{\theta}$ to maximize the likelihood function $p(\boldsymbol{y}, \underline{\boldsymbol{x}}; \boldsymbol{\theta})$, which can be written as $p(\boldsymbol{y}, \underline{\boldsymbol{x}}; \boldsymbol{\theta}) = p(\underline{\boldsymbol{x}}|\boldsymbol{y}; \boldsymbol{\theta}) p(\boldsymbol{y}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|y_i; \boldsymbol{\theta}) \pi(y_i)$.

Truthing issues: Instead, find $\boldsymbol{\theta}$ to maximize $p(\underline{\boldsymbol{z}}, \underline{\boldsymbol{x}}; \underline{\boldsymbol{\psi}}, \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{z}_i, \boldsymbol{x}_i; \boldsymbol{\psi}_i, \boldsymbol{\theta}) \overset{(a)}{=} \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{z}_i, y_i, \boldsymbol{x}_i; \boldsymbol{\psi}_i, \boldsymbol{\theta}) = \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, y_i; \boldsymbol{\psi}_i, \boldsymbol{\theta}) p(\boldsymbol{z}_i \,|\, y_i; \boldsymbol{\psi}_i, \boldsymbol{\theta}) p(y_i; \boldsymbol{\psi}_i, \boldsymbol{\theta}) \overset{(b)}{=} \prod_{i=1}^{N} \sum_{y_i \in \mathcal{Y}} p(\boldsymbol{x}_i|y_i; \boldsymbol{\theta}) p(\boldsymbol{z}_i|y_i; \boldsymbol{\psi}_i) \pi(y_i)$, where (a) uses marginalization and (b) applies non-dependencies.

E.3 Generative Model, Random Parameters

Ideal case: Find $\boldsymbol{\theta}$ to maximize the posterior $p(\boldsymbol{\theta}|\boldsymbol{y},\underline{\boldsymbol{x}})$; then $p(\boldsymbol{\theta}|\boldsymbol{y},\underline{\boldsymbol{x}}) \overset{(a)}{\propto} p(\boldsymbol{y},\underline{\boldsymbol{x}}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \overset{(b)}{=} p(\boldsymbol{\theta})\prod_{i=1}^{N} p(\boldsymbol{x}_i|y_i,\boldsymbol{\theta})\pi(y_i)$, where (a) uses Bayes' rule and (b) substitutes the expression for $p(\boldsymbol{y},\underline{\boldsymbol{x}}|\boldsymbol{\theta})$ from the ideal case in Appendix E.2.

Truthing issues: Find $\boldsymbol{\theta}$ to maximize $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}},\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}})$. Write $p(\boldsymbol{\theta}|\underline{\boldsymbol{z}},\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}}) \overset{(a)}{\propto} p(\underline{\boldsymbol{z}},\underline{\boldsymbol{x}}|\boldsymbol{\theta};\underline{\boldsymbol{\psi}})p(\boldsymbol{\theta}) \overset{(b)}{=} p(\boldsymbol{\theta})\prod_{i=1}^{N}\sum_{y_i\in\mathcal{Y}} p(\boldsymbol{x}_i|y_i,\boldsymbol{\theta})p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i)\pi(y_i)$, where (a) applies Bayes' rule and (b) obtains $p(\underline{\boldsymbol{z}},\underline{\boldsymbol{x}}|\boldsymbol{\theta};\underline{\boldsymbol{\psi}})$ in the same way that Appendix E.2 obtained $p(\underline{\boldsymbol{z}},\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta})$.

E.4 Non-Probabilistic Models: MMSE Estimation of Empirical-Risk RV

As shorthand, denote the i^{th} loss-function RV by $\ell_i(Y_i) = L(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}),Y_i)$ and the empirical-risk RV by $R(\boldsymbol{Y}) = R(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{Y}) = N^{-1}\sum_{i=1}^{N}\ell_i(Y_i)$, which is (34). Also let $\hat{R}(\underline{\boldsymbol{Z}}) = \hat{R}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{Z}})$ be an estimator of $R(\boldsymbol{Y})$ with $mse(\hat{R}(\underline{\boldsymbol{Z}}),R(\boldsymbol{Y})) = \mathrm{E}[(\hat{R}(\underline{\boldsymbol{Z}}) - R(\boldsymbol{Y}))^2]$. The MMSE estimator is $J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{Z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi}) = \hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}}) = \arg\min_{\hat{R}} mse(\hat{R}(\underline{\boldsymbol{Z}}),R(\boldsymbol{Y}))$.

The standard result (see Appendix C) is that the MMSE estimator is the conditional mean of $R(\boldsymbol{Y})$ given $\underline{\boldsymbol{Z}}$: $\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}}) = \mathrm{E}[R(\boldsymbol{Y})|\underline{\boldsymbol{Z}}]$, which is (35). Then $\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}}) = N^{-1}\sum_{i=1}^{N}\mathrm{E}[\ell_i(Y_i)|\underline{\boldsymbol{Z}}] = N^{-1}\sum_{i=1}^{N}\mathrm{E}[\ell_i(Y_i)|\boldsymbol{Z}_i] = N^{-1}\sum_{i=1}^{N}\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i)$, which is (36). The last expression is because, for an estimator $\hat{\ell}_i(\boldsymbol{Z}_i) = \hat{\ell}_i(\boldsymbol{x}_i,\boldsymbol{\theta},\boldsymbol{Z}_i)$ of the i^{th} loss-function RV $\ell_i(Y_i)$, the MMSE estimator is $\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i) = \arg\min_{\hat{\ell}_i} mse(\hat{\ell}_i(\boldsymbol{Z}_i),\ell_i(Y_i)) = \arg\min_{\hat{\ell}_i}\mathrm{E}[(\hat{\ell}_i(\boldsymbol{Z}_i) - \ell_i(Y_i)^2] = \mathrm{E}[\ell_i(Y_i)|\boldsymbol{Z}_i]$. Thus, $\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}})$ equals the average of the MMSE estimators of the individual loss-function RVs.

Likewise, the estimation error is $\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}}) - R(\boldsymbol{Y}) = N^{-1}\sum_{i=1}^{N}\big(\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i) - \ell_i(Y_i)\big)$, the average of the estimation errors of the MMSE loss-function estimators. The estimation error of each of these estimators has $\mathrm{E}[\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i) - \ell_i(Y_i)] = 0$ and $\mathrm{var}(\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i) - \ell_i(Y_i)) = mse(\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i),\ell_i(Y_i)) = \mathrm{E}[\mathrm{var}(\ell_i(Y_i)|\boldsymbol{Z}_i)]$. The samples are independent, so the estimation errors of the MMSE loss-function estimators are independent, too. By the CLT, the estimation error of $\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}})$ converges in distribution to a normal RV with mean zero and variance $\mathrm{var}(\hat{R}^{\text{MMSE}}(\underline{\boldsymbol{Z}}) - R(\boldsymbol{Y})) = N^{-2}\sum_{i=1}^{N}\mathrm{var}(\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i) - \ell_i(Y_i)) = N^{-2}\sum_{i=1}^{N} mse(\hat{\ell}_i^{\text{MMSE}}(\boldsymbol{Z}_i),\ell_i(Y_i)) = N^{-2}\sum_{i=1}^{N}\mathrm{E}[\mathrm{var}(\ell_i(Y_i)|\boldsymbol{Z}_i)]$, which is (47).

Appendix F. Regularized Logistic Regression Training Equations

Let the feature dimensionality be D, so $\boldsymbol{\theta}$ has dimensionality $D+1$. Define $\tilde{\boldsymbol{x}}_i^{\mathrm{T}} = \begin{bmatrix}1 & \boldsymbol{x}_i^{\mathrm{T}}\end{bmatrix}$, and index $\tilde{\boldsymbol{x}}_i$ and $\boldsymbol{\theta}$ from zero rather than one. Logistic regression uses the model $Y_i|\boldsymbol{x}_i,\boldsymbol{\theta} \sim \mathrm{B}(\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}))$, where $\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}) = 1/(1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}})$, so the model assumes $p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}) = (1-\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta}))^{1-y_i}\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta})^{y_i}$. We use the L_2-regularization term $J_{\text{reg}}(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{j=1}^{D}\theta_j^2$; the intercept weight θ_0 is not subject to regularization and is omitted from the summation.

F.1 ML Training

From (26) and (30), ML training in the ideal case seeks $\boldsymbol{\theta}$ to minimize

$$J_{\text{ideal}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) = \underbrace{-\frac{1}{N}\log p(\boldsymbol{y}|\underline{\boldsymbol{x}};\boldsymbol{\theta})}_{J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y})} + \lambda J_{\text{reg}}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2,$$

and some algebra yields

$$J_{\text{ideal}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\boldsymbol{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left((1-y_i)\log\frac{1}{1+e^{+\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}} + y_i\log\frac{1}{1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\right) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2. \quad (62)$$

For the gradient, the partial derivatives with respect to θ_j are

$$\frac{\partial J_{\text{ideal}}}{\partial\theta_j} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}} - y_i\right)\tilde{\boldsymbol{x}}_i(j) + \frac{\lambda}{N}\theta_j\mathbb{1}(j\neq 0), \quad j=0,1,\ldots,D. \quad (63)$$

From (28) and (32), ML training with truthing issues means minimizing

$$\begin{aligned}
J_{\text{ML}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi}) &= \underbrace{-\frac{1}{N}\log p(\underline{\boldsymbol{z}}|\underline{\boldsymbol{x}};\underline{\boldsymbol{\psi}},\boldsymbol{\theta})}_{J_{\text{pri}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi})} + \lambda J_{\text{reg}}(\boldsymbol{\theta}) \\
&= -\frac{1}{N}\sum_{i=1}^{N}\log\sum_{y_i=0}^{1}p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta})p(\boldsymbol{z}_i|y_i;\boldsymbol{\psi}_i) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2 \\
&\overset{(a)}{=} -\frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{y_i=0}^{1}p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta})\prod_{t:z_{i,t}\neq\emptyset}p(z_{i,t}|y_i;\delta_i,\phi_t)\right) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2 \\
&\overset{(b)}{=} -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{1}{1+e^{+\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|0;\delta_i,\phi_t)\right. \\
&\qquad\left. + \frac{1}{1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|1;\delta_i,\phi_t)\right) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2,
\end{aligned}$$

where (a) applies $\boldsymbol{\psi}_i = (\delta_i,\boldsymbol{\phi})$ along with (61), and (b) is because $p_Y(0|\boldsymbol{x}_i;\boldsymbol{\theta}) = 1-\tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta})$ and $p_Y(1|\boldsymbol{x}_i;\boldsymbol{\theta}) = \tilde{g}(\boldsymbol{x}_i;\boldsymbol{\theta})$.

For the gradient, differentiation and some algebra yield

$$\begin{aligned}
\frac{\partial J_{\text{ML}}}{\partial\theta_j} = \frac{1}{N}\sum_{i=1}^{N}&\frac{\prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|0;\delta_i,\phi_t) - \prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|1;\delta_i,\phi_t)}{\prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|0;\delta_i,\phi_t) + e^{+\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}\prod_{t:z_{i,t}\neq\emptyset}p_{Z_{i,t}|Y_i}(z_{i,t}|1;\delta_i,\phi_t)} \\
&\cdot\frac{1}{1+e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}}}\tilde{\boldsymbol{x}}_i(j) + \frac{\lambda}{N}\theta_j\mathbb{1}(j\neq 0), \quad j=0,1,\ldots,D.
\end{aligned}$$

F.2 MMSE Training

For ideal training, rewrite the first term in (62) as the empirical risk (33) by setting $s = \tilde{g}(\boldsymbol{x};\boldsymbol{\theta}) = 1/(1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}})$ and defining $L(s,y) = -(1-y)\log(1-s) - y\log s$. Then

$$\frac{\partial}{\partial\theta_j}\big[L(\tilde{g}(\boldsymbol{x};\boldsymbol{\theta}),y)\big] = \left(\frac{1}{1+e^{-\tilde{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{\theta}}} - y\right)\tilde{\boldsymbol{x}}(j). \tag{64}$$

For the gradient, plugging this equation into (48) and (49) again yields (63).

For MMSE training with truthing issues, we apply (40) and minimize

$$J_{\mathrm{MMSE}}(\boldsymbol{\theta};\underline{\boldsymbol{x}},\underline{\boldsymbol{z}},\underline{\boldsymbol{\psi}},\boldsymbol{\pi}) = -\frac{1}{N}\sum_{i=1}^{N}\left(p_{Y|\boldsymbol{Z}}(0|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\log\frac{1}{1+e^{+\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\right.$$
$$\left. + p_{Y|\boldsymbol{Z}}(1|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\log\frac{1}{1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\right) + \frac{\lambda}{2N}\sum_{j=1}^{D}\theta_j^2.$$

For the gradient, plugging (64) into (50) and simplifying yields

$$\frac{\partial J_{\mathrm{MMSE}}}{\partial\theta_j} = \frac{1}{N}\sum_{i=1}^{N}\left(p_{Y|\boldsymbol{Z}}(0|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\frac{1}{1+e^{-\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}} - p_{Y|\boldsymbol{Z}}(1|\boldsymbol{z}_i;\boldsymbol{\psi}_i)\frac{1}{1+e^{+\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{\theta}}}\right)\tilde{\boldsymbol{x}}_i(j)$$
$$+ \frac{\lambda}{N}\theta_j\mathbb{1}(j\neq 0), \quad j = 0,1,\ldots,D.$$

References

Joseph K. Blitzstein and Jessica Hwang. *Introduction to Probability*. CRC Press, Boca Raton, FL, USA, 2nd edition, 2019. doi: 10.1201/9780429428357.

Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 6109–6118, July 2017. doi: 10.1109/CVPR.2017.647. URL `https://openaccess.thecvf.com/content_cvpr_2017/html/Branson_Lean_Crowdsourcing_Combining_CVPR_2017_paper.html`.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, August 2001. doi: 10.1214/ss/1009213726.

M. C. Burl, U. M. Fayyad, P. Perona, P. Smyth, and M. P. Burl. Automating the hunt for volcanoes on Venus. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 302–309, 1994. doi: 10.1109/CVPR.1994.323844.

Mark J. Carlotto. Effect of errors in ground truth on classification accuracy. *Intl. J. Remote Sensing*, 30(18):4831–4849, September 2009. doi: 10.1080/01431160802672864.

George Casella. Illustrating empirical Bayes methods. *Chemometrics and Intelligent Laboratory Systems*, 16(2):107–125, October 1992. doi: 10.1016/0169-7439(92)80050-E.

Jesús Cid-Sueiro. Proper losses for learning from partial labels. In *Neural Information Processing Systems (NeurIPS)*, pages 1565–1573. Curran Associates, Inc., December 2012. URL https://dl.acm.org/doi/10.5555/2999134.2999309.

Jesús Cid-Sueiro, Darío García-García, and Raúl Santos-Rodríguez. Consistency of losses for learning from weak labels. In *Machine Learning and Knowledge Discovery in Databases*, volume 8724 of *Lecture Notes in Computer Science*, pages 197–210. Springer, 2014. doi: 10.1007/978-3-662-44848-9_13.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, USA, 1st edition, 1991. doi for 2nd edition: 10.1002/047174882X.

A. Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979. doi: 10.2307/2346806.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. URL https://www.jstor.org/stable/2984875.

Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. Unsupervised supervised learning I: Estimating classification and regression errors without labels. *J. Machine Learning Research (JMLR)*, 11(44):1323–1351, April 2010. URL https://www.jmlr.org/papers/v11/donmez10a.html.

Dheeru Dua and Casey Graff. UCI machine learning repository. URL https://archive.ics.uci.edu/ml, 2017. URL visited Feb. 1, 2017.

Mark Everingham et al. The 2005 PASCAL visual object classes challenge. In *Machine Learning Challenges Workshop (MLCW)*, volume 3944 of *Lecture Notes in Computer Science*, pages 117–176. Springer, 2006. doi: 10.1007/11736790_8.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Networks and Learning Systems*, 25(5):845–869, 2014. doi: 10.1109/TNNLS.2013.2292894.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, 55(1):119–139, August 1997. doi: 10.1006/jcss.1997.1504.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall-CRC, Boca Raton, FL, USA, 3rd edition, 2013. doi: 10.1201/b16018.

John T. Holodnak, Jason T. Matterer, and William W. Streilein. Estimating classifier accuracy using noisy expert labels. Technical Report 1225, MIT Lincoln Laboratory, Lexington, MA, USA, January 2018.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on Amazon Mechanical Turk. In *ACM Intl. Conf. Knowledge Discovery and Data Mining (SIGKDD)*, Workshop on Human Computation (HCOMP), pages 64–67, 2010. doi: 10.1145/1837885.1837906.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *IEEE Intl. Conf. Data Mining (ICDM)*, pages 967–972. IEEE, 2016. doi: 10.1109/ICDM.2016.0121.

Edward W. Kamen and Jonathan K. Su. *Introduction to Optimal Estimation*. Springer-Verlag, London, UK, 1999. doi: 10.1007/978-1-4471-0417-9.

David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, February 2014. doi: 10.1287/opre.2013.1235.

Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1993.

Steven M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. In *Intl. Conf. Learning Representations (ICLR)*, December 2018. doi: 10.48550/arXiv.1712.04577. URL `https://openreview.net/forum?id=H1sUHgb0Z`.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, USA, 2009.

Chuck P. Lam and David G. Stork. Evaluating classifiers by means of test data with noisy labels. In *Intl. Joint Conf. Artificial Intelligence (IJCAI)*, pages 513–518. Morgan Kaufmann, 2003. URL `http://ijcai.org/Proceedings/03/Papers/076.pdf`.

Gábor Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25(1):79–87, January 1992. doi: 10.1016/0031-3203(92)90008-7.

George Marsaglia. Ratios of normal variables and ratios of sums of uniform variables. *J. Amer. Statistical Assoc.*, 60(309):193–204, March 1965. doi: 10.2307/2283145.

George Marsaglia. Ratios of normal variables. *J. Statistical Software*, 16(4):1–10, May 2006. doi: 10.18637/jss.v016.i04.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Neural Information Processing Systems (NeurIPS)*, pages 1196–1204. Curran Associates Inc., December 2013. URL `https://dl.acm.org/doi/10.5555/2999611.2999745`.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Cost-sensitive learning with noisy labels. *J. Machine Learning Research (JMLR)*, 18(155):1–33, April 2018. URL `https://jmlr.org/papers/v18/15-226.html`.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Neural Information Processing Systems (NeurIPS)*, pages 841–848. MIT Press, January 2001. URL https://dl.acm.org/doi/10.5555/2980539.2980648.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *J. Artificial Intelligence Research (JAIR)*, 70:1373—1411, May 2021a. ISSN 1076-9757. URL https://doi.org/10.1613/jair.1.12125.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*. Curran Associates, Inc., December 2021b. URL https://doi.org/10.48550/arXiv.2103.14749.

Alan V. Oppenheim and George C. Verghese. *Signals, Systems and Inference*. Pearson Education, London, UK, 2015.

Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, USA, 3rd edition, 1991.

Emmanouil A. Platanios, Avinava Dubey, and Tom Mitchell. Estimating accuracy from unlabeled data: A Bayesian approach. In *Intl. Conf. Machine Learning (ICML)*, pages 1416–1425, June 2016. URL https://proceedings.mlr.press/v48/platanios16.pdf.

Alexander Ratner, Stephen Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Intl. Conf. Very Large Data Bases (VLBD)*, volume 11(3), pages 269–282. VLDB Endowment, November 2017. doi: 10.14778/3157794.3157797.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Neural Information Processing Systems (NeurIPS)*, pages 3567–3575. Curran Associates, Inc., December 2016. URL https://dl.acm.org/doi/10.5555/3157382.3157497.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerado Hemosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *J. Machine Learning Research (JMLR)*, 11(43):1297–1322, April 2010. URL https://jmlr.csail.mit.edu/papers/v11/raykar10a.html.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *Public Library of Science (PLOS) ONE*, 10(3):1–21, March 2015. doi: 10.1371/journal.pone.0118432.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, June 1990. doi: 10.1007/BF00116037.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *ACM Intl. Conf. Knowledge Discovery and Data Mining (SIGKDD)*, pages 614—-622, 2008. doi: 10.1145/1401890.1401965.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of Venus images. In *Neural Information Processing Systems (NeurIPS)*, pages 1085–1092. MIT Press, January 1994. URL https://dl.acm.org/doi/10.5555/2998687.2998822.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *Intl. Conf. Learning Representations (ICLR)*, April 2015. doi: 10.48550/arXiv.1406.2080.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 11236–11245, 2019. doi: 10.1109/CVPR.2019.01150.

Grant Van Horn, Steve Branson, Scott Loarie, Serge Belongie, and Pietro Perona. Lean multiclass crowdsourcing. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2714–2723, 2018. doi: 10.1109/CVPR.2018.00287. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Van_Horn_Lean_Multiclass_Crowdsourcing_CVPR_2018_paper.pdf.

Brendan van Rooyen and Robert C. Williamson. A theory of learning with corrupted labels. *J. Machine Learning Research (JMLR)*, 18(228):1–50, July 2018. URL https://jmlr.org/papers/v18/16-315.html.

Harry L. Van Trees. *Detection, Estimation, and Modulation Theory. Part I: Detection, Estimation, and Linear Modulation Theory.* John Wiley & Sons, New York, NY, USA, 1968. doi: 10.1002/0471221082.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Neural Information Processing Systems (NeurIPS)*, pages 831–838. Morgan-Kaufmann, December 1991. URL https://dl.acm.org/doi/10.5555/2986916.2987018.

Yehuda Vardi and Cun-Hui Zhang. The multivariate L_1-median and associated data depth. *Proc. Natl. Acad. Sci. USA (PNAS)*, 97(4):1423–1426, February 2000. doi: 10.1073/pnas.97.4.1423.

Peter Welinder and Pietro Perona. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Workshop on Advancing Computer Vision with Humans in the Loop (ACVHL), pages 25–32, June 2010. doi: 10.1109/CVPRW.2010.5543189.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *Neural Information Processing Systems (NeurIPS)*, pages 2424–2432. Curran Associates, Inc., 2010. URL https://dl.acm.org/doi/10.5555/2997046.2997166.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Neural Information Processing Systems (NeurIPS)*, pages 2035–2043. Curran

Associates, Inc., December 2009. URL https://dl.acm.org/doi/10.5555/2984093.2984321.

Dengyong Zhou, Qiang Liu, John C. Platt, Christopher Meek, and Nihar B. Shah. Regularized minimax conditional entropy for crowdsourcing. URL https://arxiv.org/abs/1503.07240, March 2015.