Sinkhorn Label Allocation: Semi-Supervised Classification via Annealed Self-Training

Kai Sheng Tai[1] **Peter Bailis**[1] **Gregory Valiant**[1]

Abstract

Self-training is a standard approach to semi-supervised learning where the learner's own predictions on unlabeled data are used as supervision during training. In this paper, we reinterpret this label assignment process as an optimal transportation problem between examples and classes, wherein the cost of assigning an example to a class is mediated by the current predictions of the classifier. This formulation facilitates a practical annealing strategy for label assignment and allows for the inclusion of prior knowledge on class proportions via flexible upper bound constraints. The solutions to these assignment problems can be efficiently approximated using Sinkhorn iteration, thus enabling their use in the inner loop of standard stochastic optimization algorithms. We demonstrate the effectiveness of our algorithm on the CIFAR-10, CIFAR-100, and SVHN datasets in comparison with FixMatch, a state-of-the-art self-training algorithm.

1. Introduction

In semi-supervised learning (SSL), we are given a partially-labeled training set consisting of labeled examples $\{(x_i, y_i) \mid i = 1, \ldots, n_\ell\}$ and unlabeled examples $\{x_i \mid i = n_\ell + 1, \ldots, n\}$, with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Our goal in this setting is to leverage our access to unlabeled data in order to learn a predictor $f : \mathcal{X} \to \mathcal{Y}$ that is more accurate than a predictor trained using the labeled data alone. This setup is motivated by the high cost of obtaining human annotations in practice, which results in a relative scarcity of labeled examples in comparison with the total volume of unlabeled data available for training. Consequently, we are typically interested in the regime where $n_\ell \ll n$.

This paper focuses on *self-training* for semi-supervised classification tasks. Self-training, also known as self-labeling, is an SSL method where the classifier's own predictions on unlabeled data are used as additional supervision during training. Specifically, self-training involves the following alternating process: in each iteration, the classifier's outputs are used to assign labels to unlabeled examples; these artificially labeled examples are then used as supervision to update the parameters of the classifier. This intuitive bootstrapping procedure was first studied in the signal processing and statistics communities (Scudder, 1965; McLachlan, 1975; Widrow et al.; Nowlan & Hinton, 1993) and was later adopted for natural language processing (Yarowsky, 1995; Blum & Mitchell, 1998; Riloff et al., 2003) and computer vision applications (Rosenberg et al., 2005). More recently, methods based on self-training have been used to achieve strong empirical results on semi-supervised image classification tasks (Xie et al., 2020; Sohn et al., 2020).

The label assignment step is critical to the success of self-training. Incorrect assignments during training may cause further misclassifications in subsequent iterations, resulting in a feedback loop of self-reinforcing errors that ultimately yields a low-accuracy classifier. As a result, self-training algorithms commonly incorporate various heuristics for mitigating label noise. For instance, the state-of-the-art FixMatch algorithm (Sohn et al., 2020) uses a *confidence thresholding* rule wherein gradient updates only involve examples that are classified with a model probability above a user-defined threshold.

Our main contribution is a new label assignment method, Sinkhorn Label Allocation (SLA), that models the task of matching unlabeled examples to labels as a convex optimization problem. More precisely: in a classification problem where $\mathcal{Y} = \{1, \ldots, k\}$, we seek an assignment $Q \in \mathbb{R}^{n \times k}$ of n examples to k classes that minimizes the total assignment cost $\sum_{ij} Q_{ij} C_{ij}(\theta)$, where the cost $C_{ij}(\theta)$ of assigning example i to class j is given by the corresponding negative log probability under the model distribution p_θ:

$$C_{ij}(\theta) = -\log p_\theta(j \mid x_i). \tag{1}$$

This formulation is desirable for several reasons. First, we are able to subsume several commonly used label assignment heuristics within a single, principled optimization

[1] Stanford University, Stanford, CA, USA. Correspondence to: Kai Sheng Tai <kst@cs.stanford.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

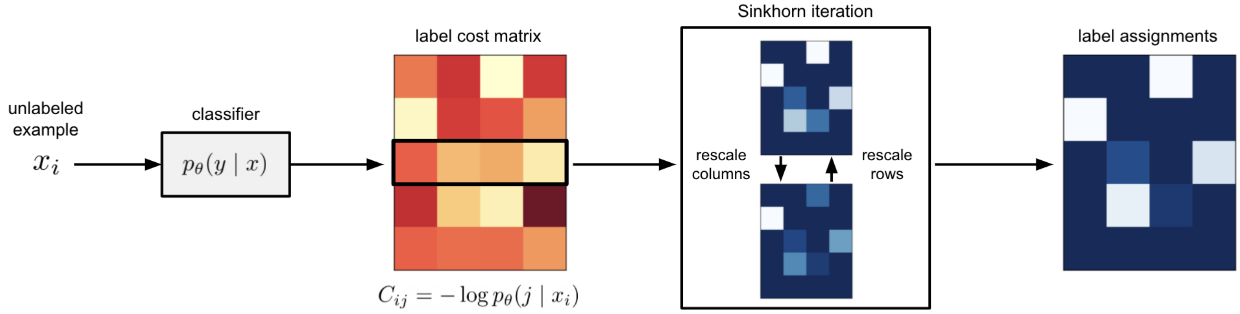

Figure 1. **A schematic representation of the label assignment process in Sinkhorn Label Allocation (SLA).** We model the label assignment task as an optimal transport problem between n examples and k classes, where the entries of the $n \times k$ assignment cost matrix are determined by the predictions of the classifier on unlabeled examples. In the figure, lighter shades correspond to lower costs and higher label assignment weights. By approximating the solution to the optimization problem using Sinkhorn iteration, we derive soft labels that can be used within a self-training algorithm. SLA allows for additional control over the label assignment process through the use of constraints on class proportions and on the total mass of allocated labels.

framework through our choice of constraints on the label assignment matrix Q. In addition to the aforementioned confidence thresholding heuristic, SLA is also able to simulate *label annealing* strategies where the labeled set is slowly grown over time (e.g., Blum & Mitchell (1998)), as well as *class balancing* heuristics that constrain the artificial label distribution to be similar to the empirical distribution of the labeled set (e.g., Joachims (1999); Berthelot et al. (2020); Xie et al. (2020)). Second, we can efficiently find approximate solutions for the resulting family of optimization problems using the Sinkhorn-Knopp algorithm (Cuturi, 2013). Consequently, we are able to run SLA within the inner loop of standard stochastic optimization algorithms while incurring only a small computational overhead.

We demonstrate the practical utility of SLA through an evaluation on standard semi-supervised image classification benchmarks. On CIFAR-10 with 4 labeled examples per class, self-training with SLA and consistency regularization achieved a mean test accuracy of 94.83% (std. dev. 0.32%) over 5 trials. This improves on the previous state-of-the-art algorithm on this task, FixMatch, which achieved a mean test accuracy of 90.10% (std. dev. 3.00%) with the same labeled/unlabeled splits, and is comparable to the mean test accuracy of FixMatch when trained with 25 labels per class.

The remainder of this paper is structured as follows. In the following section, we describe the SLA algorithm alongside a complete self-training procedure that uses SLA in its label assignment step. After a review of related work, we present our empirical findings, which include benchmark results on the CIFAR-10, CIFAR-100, and SVHN datasets, as well as an analysis of the learning dynamics induced by SLA. We conclude with a discussion of limitations and future work. Our code is available at `https://github.com/stanford-futuredata/sinkhorn-label-allocation`.

Notation. We denote the number of labeled examples by n_ℓ, the total number of labeled and unlabeled examples by n, and the number of classes by k. Let $\mathbb{R}_+$ denote the set of nonnegative real numbers, and let $\mathbf{0}_d$ and $\mathbf{1}_d$ be the zero and all-ones vectors of dimension d respectively. Let $[m]$ denote the set of integers $\{1, \ldots, m\}$, and let $\Delta_d := \{x \in \mathbb{R}_+^d \mid x^T \mathbf{1}_d = 1\}$ denote the d-simplex. Define $x_+ := \max(0, x)$, and $x_- := \min(0, x)$ as the positive and negative parts of x. For probability distributions $p, q \in \Delta_d$, define the entropy of p as $H(p) := -\sum_{i=1}^d p_i \log p_i$, the cross-entropy between p and q as $H(p, q) := -\sum_{i=1}^d p_i \log q_i$, and the Kullback-Leibler divergence between p and q as $D_{\mathrm{KL}}(p\|q) := \sum_{i=1}^d p_i \log(p_i/q_i)$. We use $\langle X, Y \rangle := \sum_{ij} X_{ij} Y_{ij}$ to denote the Frobenius inner product between matrices of equal dimension.

2. Sinkhorn Label Allocation (SLA)

We begin this section by describing the SLA optimization problem and its derivation from standard principles in semi-supervised learning. We then show how SLA can be applied within a self-training procedure in combination with consistency regularization.

2.1. Label Assignment

Soft labels. As with any label assignment procedure, the goal of SLA is to produce a label vector $q \in \mathbb{R}^k$ for a corresponding example x. SLA is a "soft" label assignment algorithm since it generates label vectors in the set $\{q \in \mathbb{R}_+^k \mid q^T \mathbf{1}_k \leq 1\}$. While the constraint $q^T \mathbf{1}_k \leq 1$ may appear to be somewhat unusual since soft labels are typically defined to be elements of Δ_k, we note that the soft labels returned by SLA can be written as the product $q = \eta \tilde{q}$ of a distribution $\tilde{q} \in \Delta_k$ and a scalar weight $\eta \in [0, 1]$. Thus, when we plug q into the standard cross-entropy loss $H(q, p)$, we obtain $H(q,p) := -\sum_{i=1}^{k} q_i \log p_i = -\eta \sum_{i=1}^{k} \tilde{q}_i \log p_i$. An SLA soft label therefore yields a *weighted* cross-entropy loss when directly used as the target “distribution” during training.

Optimization problem. SLA derives its label assignments from the solution to the following linear program (LP):

$$\begin{aligned} \underset{Q\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \langle Q, C\rangle && (2)\\ \text{s.t.} \quad & Q_{ij} \geq 0, && \\ & Q\mathbf{1}_k \preceq \mathbf{1}_n, && \\ & Q^T\mathbf{1}_n \preceq \mathbf{1}_k + nb, && (3)\\ & \mathbf{1}_n^T Q \mathbf{1}_k \geq n(\rho - \mu_+) - 1, && (4)\end{aligned}$$

where C is the non-negative cost matrix derived from the model predictions (Eq. 1), $b \in \mathbb{R}_+^k$ is a vector of upper bounds on the fraction of labels that can be allocated to each class, $\rho \in [0,1]$ is the total fraction of labels to be allocated, and $\mu := 1 - b^T\mathbf{1}_k$. We subtract μ_+ from ρ in the mass constraint (4) to ensure that the problem is feasible. We also introduce some slack to the constraints by adding 1 to each of the column constraints (3) and subtracting 1 from the mass constraint (4) to ensure strict feasibility in order to avoid numerical instability in the final implementation.

We can derive the upper bound constraints from one of several sources. Most directly, we may have prior knowledge of the label distribution, for example in settings where we have access to aggregate group-level statistics but not instance-level labels (Kuck & de Freitas, 2005). Under the assumption that the labeled examples are drawn i.i.d. from the same distribution as the unlabeled examples, we may estimate upper bounds using confidence intervals for binomial proportions, e.g., the Wilson score interval (Wilson, 1927). In settings where the unlabeled examples are sampled from a different distribution, we can estimate label proportions using methods from the domain adaptation literature (Lipton et al., 2018; Azizzadenesheli et al., 2019).

Derivation. The LP formulation used in SLA (2) can be derived from standard principles in SSL. We start by considering the following simplified label assignment problem over label distributions $Q_i \in \Delta_k$:

$$\underset{Q_i\in\Delta_k}{\text{minimize}} \quad \sum_{i=1}^{n} D_{\mathrm{KL}}(Q_i \parallel P_i) + H(Q_i). \qquad (5)$$

This objective balances two terms: the KL-divergence term captures the requirement that the assigned labels are close to the model predictions P_i, while the entropy term represents the assumption that an optimal classifier should be able to unambiguously assign a class to all the unlabeled examples. The latter implements the standard *cluster assumption* that typifies many SSL algorithms, namely that the decision boundary of the classifier should only pass through low-density regions of the data distribution (Joachims, 1999; 2003; Sindhwani et al., 2006). The entropic penalty can also be seen to be an instance of the entropy minimization criterion in SSL (Grandvalet & Bengio, 2005).

Algorithm 1 Sinkhorn Label Allocation (SLA)

```
**Input:** label cost matrix $C \in \mathbb{R}_+^{n\times k}$, upper bounds $b \in \mathbb{R}_+^k$, allocation fraction $\rho \in [0,1]$, Sinkhorn regularization parameter $\gamma > 0$, tolerance $\epsilon > 0$
**Output:** scaling variables $\alpha, \beta$
$\alpha \leftarrow \mathbf{0}_{n+1}$, $\beta \leftarrow \mathbf{0}_{k+1}$
$M \leftarrow \begin{bmatrix} e^{-\gamma C} & \mathbf{1}_n \\ \mathbf{1}_k^T & 1 \end{bmatrix}$
// Set target row sums $r$ and column sums $c$
$\mu \leftarrow 1 - b^T\mathbf{1}_k$
$r \leftarrow \begin{bmatrix} \mathbf{1}_n^T & 1 + k + n(1 - \rho - \mu_-) \end{bmatrix}^T$
$c \leftarrow \begin{bmatrix} (\mathbf{1}_k + nb)^T & 1 + n(1 - \rho + \mu_+) \end{bmatrix}^T$
// Run Sinkhorn iteration
**while** $\|c - M^T e^\alpha\|_1 > \epsilon$ **do**
  $\beta \leftarrow \log c - \log M^T e^\alpha$
  $\alpha \leftarrow \log r - \log M e^\beta$
**end while**
**return** $\alpha, \beta$
```

Using the definition of the KL-divergence, we can rewrite the objective in (5) as follows:

$$\begin{aligned} &\sum_{i=1}^{n} D_{\mathrm{KL}}(Q_i \parallel P_i) + H(Q_i) \\ &\quad = -\sum_{i=1}^{n}\sum_{j=1}^{k} Q_{ij} \log P_{ij} = \langle Q, C\rangle, \end{aligned}$$

with $C_{ij} := -\log P_{ij}$. By relaxing the constraint $Q_i \in \Delta_k$ to allow partial label allocations and adding the class upper bound and total mass constraints, we obtain the LP formulation used for label assignment with SLA (2).

Generality. This LP encodes several defining characteristics of existing label assignment procedures for self-training. For example, suppose that we set $b = \mathbf{1}_k$ (such that constraint (3) is vacuous), and we replace the mass constraint with $\mathbf{1}_n^T Q \mathbf{1}_k \geq n$ to ensure full allocation. Then a solution to the LP is to set $Q_{ij} = 1$ iff $j = \arg\min_{j'} C_{ij'}$; this is the assignment scheme used in *pseudo-labeling* (Lee, 2013). If instead we have $\rho = 0.1$ in the mass constraint, then we have $Q_{ij} = 1$ iff $j = \arg\min_{j'} C_{ij'}$ and x_i is among the 10% most confidently classified examples. The resulting allocation strategy is therefore similar to both confidence thresholding and label annealing heuristics. Likewise, the column constraints (3) can be used to represent class balancing heuristics frequently used in SSL.

We may additionally elect to simulate several other label assignment heuristics, e.g.: (1) allocation upper bounds on subsets of classes instead of individual classes; (2) time-varying column upper bounds to introduce new classes over time; and (3) time-varying row upper bounds to simulate curriculum learning (Bengio et al., 2009), given *a priori* knowledge on the difficulty of individual examples. For simplicity, we restrict our attention in this work to the combination of label annealing and class balancing.

While the label allocation LP can be used to simulate several existing heuristics, a distinguishing property of this formulation is that it aims to optimize the label assignment globally over the entire set of unlabeled examples—this is necessary since active mass and column constraints will, in general, introduce dependencies between assignments to individual examples.

Fast approximation. General-purpose LP solvers are too slow for use for label assignment within self-training due to their impractical time complexity of $O(n^{3.5})$ (Renegar, 1988). Fortunately, it is possible to transform the LP in (2) to a more tractable form that is amenable to fast approximation algorithms. We can rewrite the problem in the following equivalent form (see the Appendix for the full derivation):

$$\begin{aligned} \underset{Q,u,v,w}{\text{minimize}} \quad & \langle Q, C \rangle \qquad (6) \\ \text{s.t.} \quad & Q_{ij} \geq 0,\ u \succeq 0,\ v \succeq 0,\ \tau \geq 0, \\ & Q\mathbf{1}_k + u = \mathbf{1}_n, \\ & Q^T\mathbf{1}_n + v = \mathbf{1}_k + nb, \\ & u^T\mathbf{1}_n + \tau = 1 + n(1 - \rho + \mu_+), \\ & v^T\mathbf{1}_k + \tau = 1 + k + n(1 - \rho - \mu_-), \end{aligned}$$

where we have introduced additional variables $u \in \mathbb{R}^n$, $v \in \mathbb{R}^k$, and $\tau \in \mathbb{R}$. For conciseness, we will use

$$\tilde{Q} := \begin{bmatrix} Q & u \\ v^T & \tau \end{bmatrix}, \quad \tilde{C} := \begin{bmatrix} C & \mathbf{0}_n \\ \mathbf{0}_k^T & 0 \end{bmatrix}$$

to denote the optimization variables and corresponding cost matrix in the problem.

By inspection, the above LP has the form of an optimal transportation problem. Its solution can therefore be efficiently approximated using the Sinkhorn-Knopp algorithm (Cuturi, 2013; Altschuler et al., 2017). Given a regularization parameter $\gamma > 0$, the Sinkhorn-Knopp algorithm is an alternating projection procedure that outputs an approximate solution of the form

$$\tilde{Q} = \operatorname{diag}\left(e^{\alpha}\right) e^{-\gamma\tilde{C}} \operatorname{diag}\left(e^{\beta}\right),$$

where $\alpha \in \mathbb{R}^{n+1}$ and $\beta \in \mathbb{R}^{k+1}$, and exponentiation is performed elementwise. The algorithm iteratively updates the variables α and β such that the row and column marginals of $\tilde{Q}$ equal their target values. As $\gamma \to \infty$, the solution approaches the optimum of the LP, but the alternating projection process will in turn require more iterations to converge.

Algorithm 2 Self-training with Sinkhorn Label Allocation and consistency regularization

```
Input: examples $\{x_i \mid i \in [n]\}$, labels $\{y_i \mid i \in [n_\ell]\}$, data augmentation distributions $P_x$, unlabeled loss weight $\lambda \geq 0$, parameter update procedure ModelUpdate, allocation upper bounds $b \in \mathbb{R}^k_+$, allocation fractions $\rho_t \in [0, 1]$, Sinkhorn regularization parameter $\gamma > 0$, tolerance $\epsilon > 0$, iterations $T$
Output: classifier $p_\theta(y \mid x)$
Initialize model parameters $\theta_0$
// Initialize scaling variables and cost matrix
$\beta \leftarrow \mathbf{0}_{k+1}$
$C_{ij} \leftarrow \log k$ for $i \in [n], j \in [k]$
for $t = 1, 2, \ldots, T$ do
  Sample labeled batch $\{(x_i, y_i) \mid i \in \mathcal{B}_\ell \subset [n_\ell]\}$
  Sample unlabeled batch $\{x_i \mid i \in \mathcal{B}_u \subset [n]\}$
  Sample augmented pairs $(\tilde{x}_i, \tilde{x}'_i)$ from $P_{x_i}$
  // Compute soft labels
  for $i \in \mathcal{B}_u$ do
    $p_i \leftarrow p_{\theta_{t-1}}(y \mid \tilde{x}_i)$
    $q_i \leftarrow [p_{i1}^{\gamma} e^{\beta_1}, \ldots, p_{ik}^{\gamma} e^{\beta_k}, e^{\beta_{k+1}}]$
    $q_i \leftarrow q_i / (q_i^T \mathbf{1}_{k+1})$
  end for
  // Compute losses and update model
  $L_\ell(\theta) \leftarrow -\frac{1}{|\mathcal{B}_\ell|} \sum_{i \in \mathcal{B}_\ell} \log p_\theta(y_i \mid \tilde{x}_i)$
  $L_u(\theta) \leftarrow -\frac{1}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} \sum_{j=1}^{k} q_{ij} \log p_\theta(j \mid \tilde{x}'_i)$
  $L(\theta) \leftarrow L_\ell(\theta) + \lambda L_u(\theta)$
  $\theta_t \leftarrow$ ModelUpdate$(\theta_{t-1}, \nabla_\theta L)$
  // Update label allocation
  $C_i \leftarrow -\log p_i$ for $i \in \mathcal{B}_u$
  $(\alpha, \beta) \leftarrow \text{SLA}(C, b, \rho_t, \gamma, \epsilon)$    (Algorithm 1)
end for
return $p_{\theta_T}(y \mid x)$
```

Algorithm 1 summarizes the SLA label assignment process.

2.2. Self-Training Algorithm

We can now use SLA label assignment within a self-training algorithm to instantiate a SSL procedure. Algorithm 2 uses SLA in combination with *consistency regularization* (Bachman et al., 2014; Sajjadi et al., 2016; Laine & Aila, 2017), which can be seen as a recent variant of earlier multi-view SSL approaches (Blum & Mitchell, 1998) that penalize deviations between model predictions on perturbed instances of training examples.

In particular, Algorithm 2 incorporates the form of consistency regularization used in FixMatch (Sohn et al., 2020).

This approach samples a pair $(\tilde{x}, \tilde{x}')$ of augmented instances of an example x: $\tilde{x}$ is a "weakly augmented" view of x, while $\tilde{x}'$ is a "strongly augmented" view corresponding to small and large perturbations of the base point respectively. For example, a weakly augmented image may be perturbed with a small random translation, while a strongly augmented image may additionally be subject to large distortions in color. Since we derive the soft labels q solely from the weakly augmented instances $\tilde{x}$, the unlabeled loss term L_u encourages predictions on the strongly augmented views to match the labels allocated to the weakly augmented views.

Algorithm 2 maintains an $n \times k$ cost matrix C where each row corresponds to an unlabeled example. We update the entries of C with the negative log probabilities assigned to each class by the current model (Eq. 1). To avoid incurring the computational cost of evaluating the model on the full set of examples in each iteration, we only update the rows of C corresponding to the current unlabeled minibatch.

In each iteration, we derive the soft label q for a given unlabeled example x by rescaling the predicted label distribution using the scaling variable β obtained from SLA:

$$q_j = \frac{p_\theta(j \mid x)^\gamma e^{\beta_j}}{e^{\beta_{k+1}} + \sum_{j'=1}^{k} p_\theta(j' \mid x)^\gamma e^{\beta_{j'}}}. \tag{7}$$

This rescaling is identical to that used in the Sinkhorn-Knopp algorithm. We can interpret the additional $e^{\beta_{k+1}}$ term in the normalizer as a *soft threshold*: if $e^{\beta_{k+1}} \gg p_\theta(j \mid x)^\gamma e^{\beta_j}$ for $j \in [k]$, then q is close to 0. In such a case, we are *abstaining* from assigning x to a class.

The allocation schedule ρ_t controls the fraction of examples that are assigned labels in each iteration. In our experiments, we generally use a simple linear ramp from no allocation to full allocation, $\rho_t = (t-1)/(T-1)$. In our ablation studies, we evaluate the performance of our label allocation algorithm in the absence of this ramping strategy.

3. Related Work

Annealing and homotopy methods. Over the course of a training run where the label allocation parameter ρ is swept from 0 to 1, SLA prioritizes the highest-confidence predictions in its label assignments. This assignment strategy is reminiscent of curriculum learning (Bengio et al., 2009) and self-paced learning (Kumar et al., 2010), where "easy" examples are used early in training and more "difficult" examples are gradually introduced over time. As with these other methods, self-training with SLA can be interpreted as a *homotopy* or *continuation method* for nonconvex optimization (Allgower & Georg, 1990), which iteratively solve a sequence of relaxed problem instances that eventually converges to the original optimization problem. In the context of SSL, Sindhwani et al. (2006) propose a homotopy strategy for training semi-supervised SVMs that gradually anneals the entropy of soft labels assigned to the unlabeled examples—this strategy differs from our approach since it involves an assignment of labels to *all* unlabeled examples in each iteration.

The confidence thresholding heuristic used in FixMatch (Sohn et al., 2020) also induces an annealing schedule: as model predictions become more confident over the course of training, unlabeled examples are more frequently assigned labels and thus more frequently contribute to model updates.[1] However, it is generally unclear how the confidence threshold should be set since the predictions of many modern neural network architectures are known to not be calibrated without additional post-processing (Hendrycks & Gimpel, 2017; Guo et al., 2017). Our use of an allocation schedule in SLA obviates the need to manually select a confidence threshold parameter for training.

Robust estimation. The bootstrapping process in self-training is essentially a problem of learning with noisy labels where the source of label noise is the inaccuracy of the classifier during training, in contrast to the typical assumptions of random or adversarial label corruption. We can view the label annealing component of SLA as a means of mitigating label noise—from this perspective, the SLA label assignment process is similar to robust learning methods such as iterative trimmed loss minimization (Shen & Sanghavi, 2019), which computes model updates using only a preset fraction of low-loss training examples.

Class balancing. The use of class balancing criteria has long been commonplace in SSL algorithms in order to avoid imbalanced label assignments. The original co-training algorithm (Blum & Mitchell, 1998) grows the training set by adding artificially labeled examples in proportion to the class ratio in the labeled set, while the Transductive SVM (Joachims, 1999) fixes the number of positive labels to be assigned to the unlabeled data. Variants of class balancing have since appeared in many other works (Zhu & Ghahramani, 2002; Sindhwani et al., 2006; Chapelle et al., 2008). A recent example is the ReMixMatch algorithm, which employs a variant of class balancing called "distribution alignment" (Berthelot et al., 2020). In self-supervised learning, Sinkhorn iteration has been used to ensure an even assignment of examples to clusters (Asano et al., 2020; Caron et al., 2020). A distinguishing feature of SLA is its use of upper bounds instead of exact equality constraints, which allows for additional flexibility in the label assignment process.

Our class proportion constraints are also similar to prior work on learning from label proportions (Kuck & de Freitas, 2005; Musicant et al., 2007; Dulac-Arnold et al., 2019),

[1] We document this effect empirically in Sec. 4.2.

where the goal is to learn a classifier given the label distributions of several subsets of examples. Our setting involves a single global set of constraints on the class distribution of the unlabeled set, in contrast to the LLP setting which concerns large sets of small bags of data.

Additionally, class proportion constraints are also conceptually related to methods for learning with constraints on the model posterior, e.g., constraint driven learning (Chang et al., 2007), generalized expectation criteria (Mann & McCallum, 2007; 2008), and posterior regularization (Ganchev et al., 2010). These methods aim to guide learning by constraining posterior expectations of user-defined features that encode prior knowledge about the desired solution.

Expectation Maximization. Finally, we remark that the alternating minimization process in Algorithm 2 that iterates between label updates and model updates is similar to applications of the EM algorithm in SSL (Nigam et al., 2000). Our algorithmic approach differs since we do not use label expectations with respect to a probabilistic model.

4. Experiments

In this empirical study, we investigate (1) the accuracy of classifiers trained with SLA, (2) the training dynamics induced by the SLA label assignment process, and (3) the effect the hyperparameters introduced by SLA. Our main baseline for comparison is the FixMatch algorithm (Sohn et al., 2020) since it is a state-of-the-art method for semi-supervised image classification. For each configuration, we report the mean and standard deviation of the error rate across 5 independent trials.

Datasets and labeled splits. We used the CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and SVHN (Netzer et al., 2011) image classification datasets with their standard train/test splits. In each trial, we independently sampled a labeled set without replacement from the training split, and we used the same labeled/unlabeled splits across runs of different methods. We used labeled set sizes of $\{10, 20, 40, 80, 250\}$ for CIFAR-10, $\{400, 800, 2500\}$ for CIFAR-100, and $\{20, 40, 80\}$ for SVHN.

Following the experimental protocol in recent work (Berthelot et al., 2020; Sohn et al., 2020), we chose the label distribution of the labeled set such that it is as close as possible to the true label distribution of the training set in total variation distance, subject to the constraint that there is at least one example sampled for each class. We observe that this setup implies that the empirical label distributions of the labeled sets for CIFAR-10/100 are always *well-specified*, in the sense that they are equal to the true distribution of labels in the training set.[2] In contrast, the empirical label distributions for SVHN are *misspecified* since the training label distribution is non-uniform.[3] Since the well-specified setting is arguably somewhat unrealistic for real-world SSL applications, we additionally report the results of CIFAR-10 experiments in the misspecified case where the labeled sets are sampled uniformly without replacement from the training split, conditioned on there being at least one example per class.

Hyperparameters. Our experiments used the same experimental setup as in the evaluation of FixMatch where applicable. We optimized our classifiers using the stochastic Nesterov accelerated gradient method with a momentum parameter of 0.9 and a cosine learning rate schedule given by $0.03 \cos(7\pi t/16T)$, where t is the current iteration and $T = 2^{20}$ is the total number of iterations.[4] We used a labeled batch size of 64, an unlabeled batch size of 448, weight decay of 5×10^{-4} on all parameters except biases and batch normalization weights, and unlabeled loss weight $\lambda = 1$. For CIFAR-10 and SVHN, we used the Wide ResNet-28-2 architecture (Zagoruyko & Komodakis, 2016), whereas for CIFAR-100, we used the Wide ResNet-28-8 architecture (with a weight decay of 10^{-3}). When evaluating on the test set, we used an exponential moving average of the model parameters (Tarvainen & Valpola, 2017) with a decay parameter of 0.999. We used a confidence threshold of 0.95 for our FixMatch baselines.

For hyperparameters specific to SLA, we used an Sinkhorn regularization parameter of $\gamma = 100$ and tolerance parameter $\epsilon_t = 0.01\|c_t\|_1$ for Sinkhorn iteration, where c_t is the target column sum at iteration t. Unless otherwise specified, we increased the allocation parameter ρ linearly from 0 to 1 over the course of training. For CIFAR-10/100, we used the empirical label distribution of the labeled examples as the class proportion upper bounds b. For SVHN, we used upper bounds given by the 80% Wilson score interval (Wilson, 1927) since the empirical label distribution only approximates the true label distribution.

Data augmentation. We ran both SLA self-training and the FixMatch baselines with the same data augmentation distributions. For consistency regularization, our weak augmentation policy consisted of random translations of up to 4 pixels (for all datasets) and random horizontal flips with probability 0.5 (for CIFAR-10/100, but not SVHN). Our strong augmentation policy consisted of the weak augmentation policy composed with RandAugment (Cubuk et al., 2020), followed by 16×16 Cutout augmentations (DeVries & Taylor, 2017).

Computational cost. In our runs, SLA incurred an average

[2] This is due to our choices of labeled set sizes, and that CIFAR-10/100 are balanced datasets.

[3] The TV distances for SVHN with 20, 40, and 80 labels are 0.068, 0.034, and 0.018 respectively.

[4] This schedule anneals the learning rate from 0.03 to ≈ 0.006.

Table 1. A test error comparison (mean and standard deviation over 5 runs) on CIFAR-10 and CIFAR-100 with varying labeled set sizes. We obtained the FixMatch results using our own reimplementation, while the results for MixMatch (Berthelot et al., 2019), UDA (Xie et al., 2019), and ReMixMatch (Berthelot et al., 2020) are as reported in (Sohn et al., 2020). SLA improves on the mean accuracy of FixMatch on CIFAR-10 and CIFAR-100 for all labeled set sizes, except for the 2500 label runs on CIFAR-100.

	CIFAR-10					**CIFAR-100**		
Method	10 labels	20 labels	40 labels	80 labels	250 labels	400 labels	800 labels	2500 labels
MixMatch	-	-	47.54 ± 11.50	-	11.05 ± 0.86	67.61 ± 1.32	-	39.94 ± 0.37
UDA	-	-	29.05 ± 5.93	-	8.82 ± 1.08	59.28 ± 0.88	-	33.13 ± 0.22
ReMixMatch	-	-	19.10 ± 9.64	-	5.44 ± 0.05	44.28 ± 2.06	-	27.43 ± 0.31
FixMatch	37.02 ± 8.35	20.53 ± 8.90	9.90 ± 3.00	6.42 ± 0.21	5.09 ± 0.61	43.42 ± 2.41	35.53 ± 1.00	27.99 ± 0.42
SLA	34.13 ± 10.83	18.09 ± 6.77	5.17 ± 0.32	5.02 ± 0.28	4.89 ± 0.27	41.44 ± 1.41	34.31 ± 1.09	28.73 ± 0.44

Table 2. A test error comparison on SVHN with varying labeled set sizes. The results for MixMatch, UDA, and ReMixMatch are as reported in (Sohn et al., 2020). SLA improves on FixMatch on average, except with 20 labeled examples where the class upper bounds are poor estimates of the true label distribution.

	SVHN		
Method	20 labels	40 labels	80 labels
MixMatch	-	42.55 ± 14.53	-
UDA	-	52.63 ± 20.51	-
ReMixMatch	-	3.34 ± 0.20	-
FixMatch	14.92 ± 7.82	4.74 ± 3.28	2.98 ± 1.31
SLA	22.85 ± 9.84	3.63 ± 2.91	2.48 ± 0.18

Table 3. A test error comparison on CIFAR-10 with 40 labels distributed evenly between the classes (Uniform) and with 40 labels sampled uniformly from the training set, conditioned on at least one label being drawn for each class (Multinomial). Accuracy degrades for all methods in the more challenging multinomial setting.

Method	Uniform	Multinomial
FixMatch	9.90 ± 3.00	11.23 ± 3.56
FixMatch (with DA)	5.70 ± 1.63	18.64 ± 11.29
SLA (without upper bounds)	9.71 ± 5.95	13.40 ± 6.41
SLA	5.17 ± 0.32	14.95 ± 7.12

21.1% overhead in total training time for CIFAR-10 and a 23.2% overhead for CIFAR-100.

4.1. Classification Benchmarks

Tables 1 and 2 summarize the test error rates achieved by self-training with FixMatch and SLA on CIFAR-10, CIFAR-100 and SVHN. We observe an improvement in mean accuracy over FixMatch on the CIFAR-10 dataset across all configurations, on CIFAR-100 with 400 and 800 labels, and on SVHN with 40 and 80 labels. In particular, the accuracy of SLA on CIFAR-10 with 40 labels (94.83%) was comparable to the accuracy of FixMatch on 250 labels (94.91%).

SLA often yielded more consistent results across runs; for example, the standard deviation for CIFAR-10 with 40 labels was reduced by 2.7%, and for SVHN with 80 labels by 1.1%. This can be attributed to the use of the upper bound constraints, which help prevent convergence to poor local minima due to the overrepresentation of certain classes during training.

Table 3 compares test errors on CIFAR-10 with 40 labels, where the empirical label distribution of the labeled set is well-specified (Uniform) or misspecified (Multinomial).[5] We compare SLA with and without the class proportion upper bounds against standard FixMatch and FixMatch with the distribution alignment (DA) heuristic (Berthelot et al., 2020) that encourages the model label distribution to match the empirical label distribution. In the multinomial setting, we used 80% Wilson upper bounds for SLA. As expected, the performance of all four methods degrades in the more challenging multinomial setting. FixMatch with DA incurs a large misspecification penalty since DA essentially imposes a soft equality constraint with the empirical label distribution. In comparison, SLA incurs a smaller accuracy penalty due to its more forgiving upper bound constraints.

4.2. Training Dynamics

Figure 2 shows the total fraction of unlabeled examples that are assigned labels as a function of the training iteration count. These plots show that the FixMatch confidence thresholding criterion induces an implicit annealing schedule where the allocated fraction increases quickly early in training. In fact, FixMatch never reaches full label allocation with its fixed confidence threshold in the case of CIFAR-100 with 400 labels. We suggest that the explicit allocation schedule used in SLA is a more intuitive interface for practitioners than the fixed confidence threshold used in FixMatch.

In the bottom row of Figure 2, we observe that SLA typically achieves higher test accuracy at any fixed allocation frac-

[5] The mean TV distance to the true label distribution in the multinomial setting is ≈ 0.154.

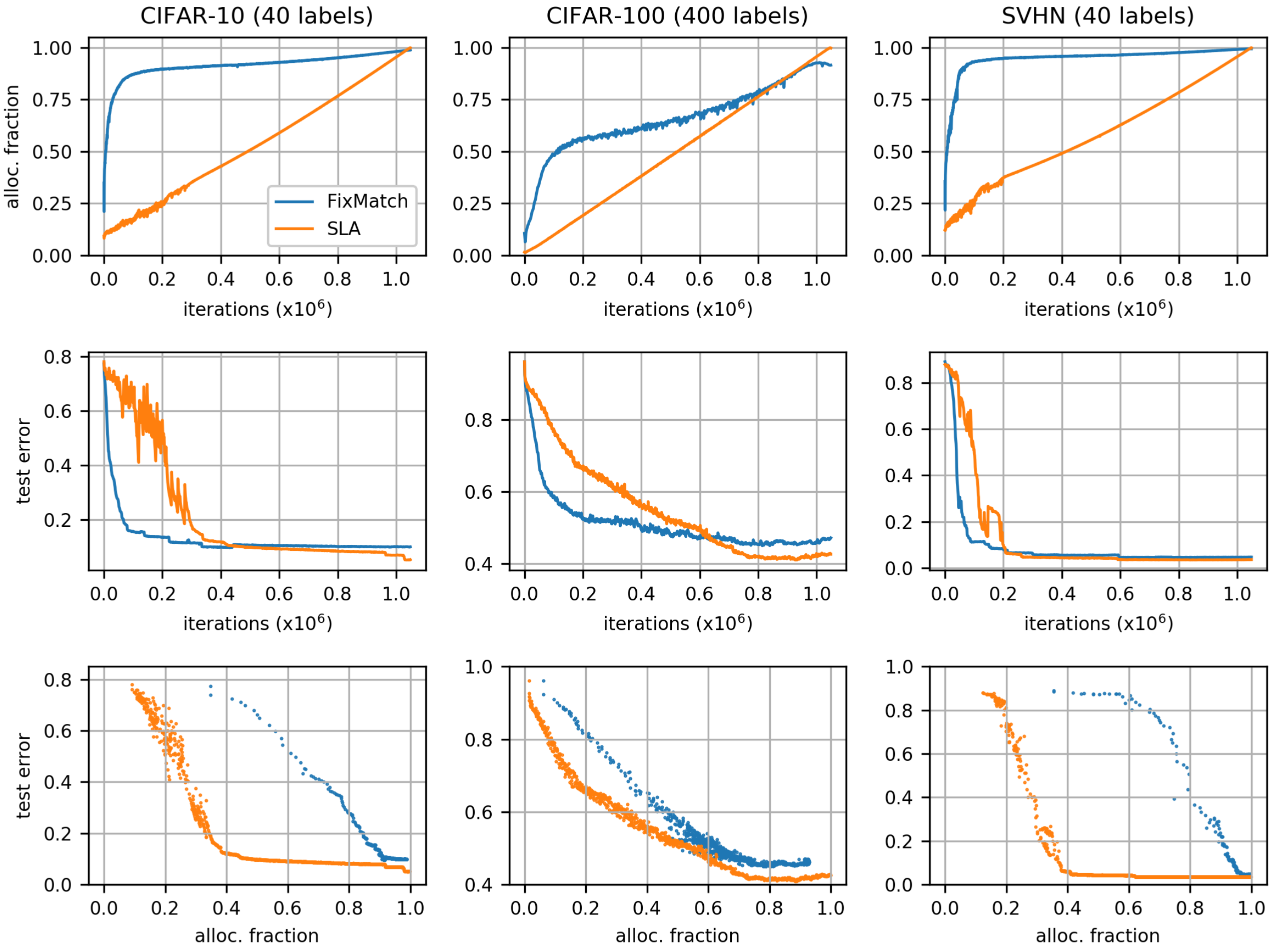

Figure 2. The fraction of unlabeled examples assigned labels over the course of training (top row), test error during training (middle row), and the relationship between the allocation fraction and the test error during training (bottom row). FixMatch induces an annealing schedule that quickly increases the allocation fraction early in training, while SLA allocation increases approximately linearly according to the ρ_t schedule (the SLA allocation is not exactly linear since the mass constraint is a lower bound). In these experiments, SLA yields lower test error on average across all allocation fractions.

tion. Further, we note that the effect of the SLA constraints is apparent in the CIFAR-10 runs, where the accuracy improves in a stepwise fashion towards the end of training as the remaining "difficult" examples are assigned labels.

For CIFAR-100 (middle column), we find that the test error for SLA reaches a minimum and then increases towards the end of training. This "U"-shaped test error rate suggests that in some settings, label noise due to misclassification can start to dominate as we approach full allocation. This observation indicates that partial label allocation, e.g. with a truncated schedule such as $\rho_t = \min\left(0.8, \frac{t-1}{T-1}\right)$, can be an effective strategy for certain tasks.

In Figure 3, we plot the values of the scaling values β and the label allocations corresponding to two pairs of similar classes from CIFAR-10 and CIFAR-100. These plots illustrate the role of the scaling variables in influencing the dynamics of training by promoting underrepresented classes and inhibiting overrepresented classes. Indeed, this is consistent with their interpretation as dual variables corresponding to the class balancing constraints in the optimization problem.

4.3. Ablations

We investigate the effects of SLA-specific hyperparameters through a series of ablation experiments. First, the use of a label annealing strategy is important: without any label annealing, i.e., by setting $\rho_t = 1$, we achieve a test error of $13.67 \pm 1.83\%$ on CIFAR-10 with 40 labels (vs. $5.17 \pm 0.32\%$ with the default linear ramp). The use of class proportion upper bounds has a significant positive effect when the label distribution is well-specified: removing these class constraints while retaining label annealing achieves $9.71 \pm 5.95\%$.

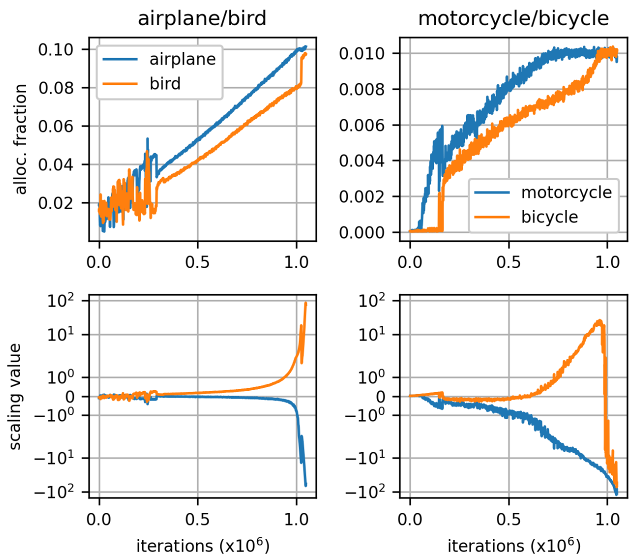

Figure 3. SLA scaling variables β and class allocations over the course of training for two similar class pairs: airplane/bird (CIFAR-10) and motorcycle/bicycle (CIFAR-100). The scaling variables promote underrepresented classes (positive values) and inhibit overrepresented classes (negative values). The *bicycle* class is initially promoted, but as it approaches the allocation constraint of 0.01, the corresponding scaling variable turns negative in order to enforce the upper bound.

Larger values of the Sinkhorn regularization parameter γ result in a better approximation to the solution of the optimal transportation problem, at the cost of additional time spent on Sinkhorn iteration. We find that the use of an overly coarse approximation has a significant negative effect on final accuracy. Specifically, $\gamma = 1, 10, 100, 1000$ achieve single-run error rates of 42.48, 5.78, 4.94, and 5.10% respectively on CIFAR-10 with 40 labels. For our set of tasks, we find that $\gamma = 100$ strikes an acceptable trade-off between approximation accuracy and speed.

5. Discussion

In this work, we motivated SLA as an optimization-based strategy for assigning labels in self-training. This framework proved to be sufficiently rich to synthesize several existing label assignment heuristics in SSL under a single formulation, while still retaining computational tractability via the use of an efficient approximation algorithm.

An attractive direction for future work is to extend the flexibility of this general optimization framework by allowing for a wider range of constraints, thus allowing for the incorporation of richer forms of prior knowledge in SSL problems. A possible extension of SLA is to replace the Sinkhorn-Knopp iteration with Dykstra's algorithm (Dykstra, 1985; Benamou et al., 2015), which performs cyclic Bregman projections onto collections of convex sets. An example usecase that would be enabled by such an extension would be *semi-supervised multi-label learning*. This setting corresponds to a simple modification of our LP constraints: we stipulate $0 \leq Q_{ij} \leq 1$ and replace the constraint $Q\mathbf{1}_k \preceq \mathbf{1}_n$ with $Q\mathbf{1}_k \preceq N\mathbf{1}_n$, where N is the maximum number of labels that can be assigned to an example. Another possible usecase is the introduction of *lower bounds* on class proportions in addition to our current upper bound constraints. The use of lower bounds would help prevent one of the failure modes we observed in our experiments, namely where a subset of classes end up with zero allocation when using loose upper bounds or partial label allocation. These lower bound constraints are not handled by our current reduction to optimal transport.

In our experiments, we found that SLA is susceptible to incurring an accuracy penalty when the constraints are misspecified. In a sense, this should not be surprising as it can be understood as a manifestation of the "no free lunch" theorems. However, we nevertheless speculate that it may be possible to extend SLA such that it is able to adaptively identify possible misspecification over the course of training. For instance, we observe empirically that infeasible or near-infeasible constraints result in a chaotic oscillation of the model parameters and scaling variables—such signals may potentially be used to dynamically tune the constraint set during training. Orthogonally, we note that methodological advancements on the problem of estimating label proportions from unlabeled data can yield immediate improvements for SLA via tighter bounds on class proportions.

Acknowledgements

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, and VMware—as well as Toyota Research Institute, Northrop Grumman, Cisco, and SAP. This work was also supported by NSF awards 1704417 and 1813049, ONR YIP award N00014-18-1-2295, and DOE award DE-SC0019205. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

Allgower, E. L. and Georg, K. *Numerical continuation methods: An introduction*. Springer, 1990.

Altschuler, J., Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via

Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, 2017.

Asano, Y. M., Rupprecht, C., and Vedaldi, A. Self-labeling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020.

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

Bachman, P., Alsharif, O., and Precup, D. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, 2014.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 2015.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *International Conference on Machine Learning*, 2009.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. MixMatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., Raffel, C., and Sohn, K. ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory*, 1998.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020.

Chang, M.-W., Ratinov, L., and Roth, D. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting of the Association of Computational Linguistics*, 2007.

Chapelle, O., Sindhwani, V., and Keerthi, S. S. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 2008.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. RandAugment: Practical automated data augmentation with a reduced search space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with Cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Dulac-Arnold, G., Zeghidour, N., Cuturi, M., Beyer, L., and Vert, J.-P. Deep multiclass learning from label proportions. Technical report, arXiv, 2019. 1905.12909.

Dykstra, R. L. An iterative procedure for obtaining I-projections onto the intersection of convex sets. *The Annals of Probability*, 1985.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 2010.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2005.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Joachims, T. Transductive inference for text classification using Support Vector Machines. In *International Conference on Machine Learning*, 1999.

Joachims, T. Transductive learning via spectral graph partitioning. In *International Conference on Machine Learning*, 2003.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kuck, H. and de Freitas, N. Learning about individuals from group statistics. In *Conference on Uncertainty in Artificial Intelligence*, 2005.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 2010.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

Lipton, Z. C., Wang, Y.-X., and Smola, A. J. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, 2018.

Mann, G. and McCallum, A. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Annual Meeting of the Association for Computational Linguistics*, 2008.

Mann, G. S. and McCallum, A. Simple, robust, scalable semi-supervised learning via expectation regularization. In *International Conference on Machine Learning*, 2007.

McLachlan, G. J. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 1975.

Musicant, D. R., Christensen, J. M., and Olson, J. F. Supervised learning by training on aggregate outputs. In *International Conference on Data Mining*, 2007.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 2000.

Nowlan, S. J. and Hinton, G. E. A soft decision-directed LMS algorithm for blind equalization. *IEEE Transactions on Communications*, 1993.

Renegar, J. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 1988.

Riloff, E., Wiebe, J., and Wilson, T. Learning subjective nouns using extraction pattern bootstrapping. In *HLT-NAACL*, 2003.

Rosenberg, C., Hebert, M., and Schneiderman, H. Semi-supervised self-training of object detection models. In *IEEE Workshop on Applications of Computer Vision*, 2005.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2016.

Scudder, H. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965.

Shen, Y. and Sanghavi, S. Learning with bad training data via Iterative Trimmed Loss Minimization. In *International Conference on Machine Learning*, 2019.

Sindhwani, V., Keerthi, S. S., and Chapelle, O. Deterministic annealing for semi-supervised kernel machines. In *International Conference on Machine Learning*, 2006.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

Widrow, B., McCool, J., Larimore, M. G., and Johnson, C. R. Stationary and nonstationary learning characteristics of the LMS adaptive filter. In *Aspects of Signal Processing*.

Wilson, E. B. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 1927.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. 2019.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with Noisy Student improves ImageNet classification. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, 1995.

Zagoruyko, S. and Komodakis, N. Wide Residual Networks. In *British Machine Vision Conference*, 2016.

Zhu, X. and Ghahramani, Z. Learning from labeled and unlabeled data with label propagation. Technical report, CMU CALD, 2002. CMU-CALD-02-107,.