An LP for Sequential Learning Under Budgets

Joseph Wang
Boston University

Kirill Trapeznikov
Boston University

Venkatesh Saligrama
Boston University

Abstract

We present a convex framework to learn sequential decisions and apply it to the problem of learning under a budget. We consider the structure proposed in [1], where sensor measurements are acquired in a sequence. The goal after acquiring each new measurement is to make a decision whether to stop and classify or to pay the cost of using the next sensor in the sequence. We introduce a novel formulation of an empirical risk objective for the multi stage sequential decision problem. This objective naturally lends itself to a non-convex multilinear formulation. Nevertheless, we derive a novel perspective that leads to a tight convex objective. This is accomplished by expressing the empirical risk in terms of linear superposition of indicator functions. We then derive an LP formulation by utilizing hinge loss surrogates. Our LP achieves or exceeds the empirical performance of the non-convex alternating algorithm that requires a large number of random initializations. Consequently, the LP has the advantage of guaranteed convergence, global optimality, repeatability and computation efficiency.

1 Introduction

A majority of machine learning research has focused on improving performance of classification algorithms. Recently, costs in learning have gained importance, particularly the test time cost in decision making. This problem arises in classification systems constrained by a measurement acquisition budget. In this setting, a collection of sensors with varying costs is available to the decision system. The objective is to learn a classifier that utilizes inexpensive sensing modalities for majority of decisions and requests the expensive (and more informative) sensors only for the few difficult decisions. Such a strategy maintains classifier accuracy while reducing the average acquisition cost per decision.

Several researchers ([2, 3, 1, 4]) have made significant progress in developing algorithms to learn such decisions systems with promising experimental results. Due to the potentially high dimensional nature of sensor data, a discriminative learning approach is used to learn decision functions directly by minimizing an empirical risk objective over a training set. However, the optimization problems are inherently non-convex and most solutions resort to alternative minimization schemes [2, 3, 1, 4]. While experimental results demonstrate good performance, lack of global optimality prevents theoretical guarantees for the algorithms and the solutions.

In this work, we focus on the sequential decision framework studied in [1]. In this setting, the order in which sensors are acquired is given.[1] Typically, earlier stages use cheap or fast sensors while later stages can acquire expensive or slow sensors. The decision function at each stage controls whether to stop and classify if enough information has been acquired for a confident decision or to continue and acquire the next sensor measurement. (See Fig. 1) The authors in [3] introduce a global ERM problem and present an alternative minimization scheme to learn a decision function at each stage of the system.

Our main contribution is a novel convex formulation of the empirical risk problem. We reformulate the empirical risk in [1] into maximization of sums

Appearing in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

[1] For instance, such sequential decision problems arise in security screening and medical applications. In these scenarios, the objective is use a low cost modality (fast x-ray scanner, cheap blood test, etc) to make most classifications, and utilize an expensive modality (slow human inspection, invasive surgery) for as few as possible.

of indicator functions. This key transformation enables us to introduce convex surrogates for the indicator functions and, in turn, results in a convex optimization problem. Without our transformation, direct substitution of surrogates in the original empirical risk results in a non-convex bilinear formulation which is known to be NP-complete [5]. We upperbound this reformulated objective and reduce it to a linear program (LP). This LP formulation learns sequential decision systems on real data and has the following advantages of convex programming:

Convergence: The linear program is guaranteed to converge to a solution, whereas the alternating optimization approach of [1] has no such guarantee.

Global Optimum: The linear program converges to a globally optimal point, whereas the alternating optimization approaches of [1, 6, 7] at best can only guarantee convergence to a local minimum if the algorithm converges.

Repeatability: No random initialization is necessary, whereas previous approaches ([1, 6, 7]) for training sequential decisions all rely on multiple random initializations in an attempt to find a "good" local minimum.

Computational Efficiency: The linear program does not rely on random initialization or alternating optimization, allowing the solution to be found efficiently with a single optimization problem, whereas the alternating optimization approaches require repeatedly solving supervised learning problems. Additionally, given that the algorithm is a convex function, online training using approaches such as subgradient stochastic descent are feasible when the entire data set is not available as a whole.

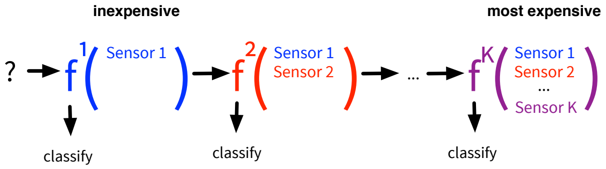


Figure 1: Multi-Stage System consists of K stages. Each stage is a classifier with a reject option. The system incurs a penalty of c_k at kth stage if it rejects to seek more measurements. The kth classifier only sees the first k sensing modalities in making a decision.

In the experimental section, we show that the proposed LP approach allows for dramatic reductions in average sensor acquisition cost while maintaining excellent classification performance on both synthetic and real world datasets. Additionally, we show performance that matches or exceeds non-convex optimization approaches while maintaining the previously mentioned advantages and clearly outperforming naive approaches.

1.1 Related Work

Several researchers have explored efficient algorithms for sequential learning and their applications to learning with test time costs.

Learning sequential decisions has also been studied extensively in unconstrained supervised learning. In the discriminative setting, attempts have been made to optimize the empirical risk formulated as a product of indicators [6, 7]. In the generative setting, the problem is loosely related to mixture of experts framework [8, 9].Alternating minimization is used, switching between learning the parameters of the "latent" distribution and training local classifiers using standard learning methods.

In this paper, we take a discriminative learning approach to learning with test time budgets and extend the work in [1] to a convex formulation. A related work in a discriminative setting is the time efficient feature extraction (TEFE) algorithm presented in [10]. This is a myopic approach consisting multiple stage of SVM classifiers. The decision whether to advance to the next stage is based solely on the margin of the current decision. These two methods are explained in more detail in the experiments section. The detection cascade (see [11, 12, 2] and references therein), a popular method in reducing computation cost in object detection, can be considered as a special case of our multi-stage sequential classifiers. However, detection cascades make partial binary decisions at each stage, delaying a positive decision until the final stage. Our approach can handle multi-class problems and can make a full classification decisions at any stage. More recently, to speed up web page ranking problems, [4] introduced a general tree of cost sensitive classifiers, where each node is parametrized with boosted weak learners. In this work, and in the cost sensitive cascade approach in [2], the authors formulate a global empirical risk objective but again resort to alternative minimization to deal with non-convexity.

In a Bayesian setting, [13, 14] model the problem of learning with test time budgets as an POMDP, [15, 16, 17] study cost sensitive decision trees, and [18] use an expected utility criteria. However, all these methods require estimating a probability likelihood that a certain feature value occurs given the features collected so far. In contrast, our problem domain deals with high dimensional measurements (such as images consisting of thousands of pixels), so estimating probability densities reliably is not possible. An alterantive approach avoiding estimation of probability distributions is to recast the problem into an imitation learning framework [19], however this requires a set of oracle actions to imitate and generally requires low-quality missing data classifiers to be learned in order to operate on the combinatorially many feature sets acquired by the policy. Additionally, the problem has been studied as a reinforcement learning problem [20, 21, 22], with the goal of parametrizing the value of each feature. As in the imitation learning framework, the missing data classifier is required, and additionally the reinforcement learning approach imposes a notion of stationarity and non-deterministic state transitions.

2 Budgeted Sequential Learning

We begin by introducing the sequential learning problem, defining the empirical risk objective in terms of indicator functions and highlighting the difficulties with this formulation.

Problem Statement Let $(\mathbf{x}, y) \in \mathcal{X} \times \{1, 2, \ldots C\}$ be distributed according to an unknown distribution $\mathcal{D}$. A data point has K features, $\mathbf{x} = \{x_1, x_2, \ldots, x_K\}$, and belongs to one of C classes indicated by its label y. A kth feature is extracted from a measurement acquired at kth stage. We define a truncated feature vector at kth stage: $\mathbf{x}^k = \{x_1, x_2, \ldots x_k\}$.[2] Let $\mathcal{X}^k$ be the space of the first k features such that $\mathbf{x}^k \in \mathcal{X}^k$.

The system has K stages, the order of the stages is fixed, and kth stage acquires a kth measurement. At each stage, k, there is a decision with a reject option, f^k. It can either classify an example, $f^k(\mathbf{x}^k) : \mathcal{X}^k \to \{1, 2, \ldots, C\}$, or delay the decision until the next stage, $f^k(\mathbf{x}^k) = r$ and incur a penalty of c_{k+1}. Here, r indicates the "reject" decision. f^k has to make a decision using only the first k sensing modalities. The last stage K is terminal, a standard classifier. We define the system risk as,

$$R(f^1, \ldots, f^K, x, y) = \sum_{k=1}^{K} S^k(\mathbf{x}^k) R^k(f^k, \mathbf{x}^k, y) \quad (1)$$

Here, R^k is the cost of classifying at kth stage, and $S^k(\mathbf{x}^k) \in \{0, 1\}$ is the binary state variable indicating whether x is classified at the kth stage.

$$R^k(\mathbf{x}^k, y, f^k) = \mathbb{1}_{f^k(x^k) \neq y_i} + \alpha \sum_{j=1}^{k} c_j \quad (2)$$

$$S^k(\mathbf{x}^k) = \begin{cases} 1, & f^j(\mathbf{x}^j) = r \wedge \ f^k(\mathbf{x}^k) \neq r, \forall j < k \\ 0, & \text{else} \end{cases}$$

If $\mathbf{x}$ is classified at stage k, the penalty is the sum of previous rejection penalties $c_1 + \ldots + c_k$ plus a penalty of 1 if the example is misclassified at stage k. The rejection penalty c_k can be thought of as the acquisition cost of feature k, with the parameter α controlling the tradeoff between average acquisition cost and budget. Small values of α penalize misclassification over acquisition cost, whereas large values of α encourage low acquisition cost at the expense of classification accuracy.

If the distribution $\mathcal{D}$ is known the problem reduces to a POMDP and the optimal strategy is to minimize the expected risk,

$$\min_{f^1, \ldots, f^K} \mathbf{E}_{\mathcal{D}} \left[R(f^1, \ldots, f^K, \mathbf{x}, y) \right] \quad (3)$$

Empirical Risk Problem However, in our setting, the probability model $\mathcal{D}$ is not known and cannot be estimated due to high-dimensionality of the data. Instead, our task is to find multi-stage decision rules based on a given training set with full measurements: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$.

We formulate an expected risk minimization problem that approximates the expected risk with a sample average over the training set:

$$\min_{f^1, \ldots, f^K} \sum_{i=1}^{N} R(f^1, \ldots, f^K, \mathbf{x}_i, y_i) \quad (4)$$

Following the decomposition in [1], we simplify the empirical risk by decomposing the reject and the classification decisions:

$$f^k(x^k) = \begin{cases} d^k(\mathbf{x}^k), & g^k(\mathbf{x}^k) \leq 0 \\ reject, & g^k(\mathbf{x}^k) > 0 \end{cases} \quad (5)$$

As in [1], we assume that at each stage, our system has a fixed stage classifier, $d^k : \mathcal{X}^k \to \{1, \ldots C\}$.[3] And our goal is only to learn a binary reject decision function for each stage, $g^k : \mathcal{X}^k \to \mathbb{R}$. An example, $\mathbf{x}_i$ is rejected at stage k, if $g^k(\mathbf{x}^k)$ is greater than zero, and classified by $d^k(\mathbf{x}^k)$ otherwise.

[2] For simplicity we refer to x_k as a feature, however x_k need not be a scalar feature and is often a set of features associated with the kth stage

[3] Since the last stage is a terminal decision the reject decision at stage $g^K(\mathbf{x}) := -1$.

Using this decomposition of reject decisions, the empirical risk minimization in (4) can be expressed in terms of indicator functions, $\mathbb{1}_{[\cdot]}$. The resulting empirical risk is a product of indicator functions:

$$R(g^1, \ldots, g^K, \mathbf{x}, y) = \tag{7}$$
$$\sum_{k=1}^{K} \underbrace{\left(\mathbb{1}_{d^k(\mathbf{x}^k) \neq y_i} + \alpha \sum_{j=1}^{k} c_k \right)}_{\text{stage risk, } R^k(\cdot)} \underbrace{\mathbb{1}_{g^k(\mathbf{x}^k) \leq 0} \prod_{j=1}^{k-1} \mathbb{1}_{g^j(\mathbf{x}^j) > 0}}_{\text{state of } \mathbf{x}_i, \; S^k(\cdot)}$$

Stage risk R^k denotes the cost of acquiring features up the stage k and making an error at that stage. The second term denotes at which stage $\mathbf{x}_i$ is classified. The system only pays a penalty R^k at stage k only if the state, S^k. is non-zero.

Difficulty in Minimizing Empirical Risk Optimizing the product of indicator functions in (7), $\min \sum_i R(g^1, \ldots, g^K, \mathbf{x}_i, y_i)$, is a computationally challenging problem. The fundamental difficulty arises due to dependency between decision functions $g^1, \ldots, g^K$. For example, the cost of making a decision with g^1 for a particular example depends on the outputs of $g^2, \ldots, g^K$, and similarly, the distribution of examples operated by g^K is dependent on the decisions of classifiers $g^1, \ldots, g^{K-1}$.

One previously proposed approach to solving this problem is approximate block coordinate descent, where each individual binary reject function is solved while holding the rest of the system fixed, yielding a supervised learning problem at each step [1]. Unfortunately, this approach does not have any optimality or convergence guarantees and can be computationally expensive.

Previous solutions proposed for learning sequential decision functions introduce loss functions in place of the indicators and find a local minima of the resulting bilinear problem [6, 7]. However, directly replacing indicators with upper-bounding surrogates, such as hinge or logistic losses, yields a bilinear function, making global optimization intractable. As previously shown, the bilinear separation problem is NP-complete [5] and a global minima cannot be efficiently found. Instead, a local minima is found using alternating optimization, with each alternating optimization solved as a quadratic program.

3 Convex Sequential Learning

Rather than directly substituting surrogate functions for indicators in (7) and attempting to solve the previously described bilinear optimization problem, we reformulate the empirical risk objective. By doing so, the risk is transformed from a product of indicator functions to a maximization of sums of indicators. As a result, introducing convex upper-bounding surrogates no longer results in a computationally difficult bilinear problem, but instead yields a convex minimization problem, allowing for globally optimal solutions to be efficiently found.

In reformulating the empirical risk, we find it useful to define the quantities:

$$\pi_i^k = \mathbb{1}_{d^k(\mathbf{x}_i^k) = y_i} + \alpha \sum_{j=k+1}^{K} c_j \tag{8}$$

The value π_i^k is composed of two terms, an indicator, representing if $\mathbf{x}_i^k$ is correctly classified at stage k, and the sum of the penalties after stage k, which are not incurred if $\mathbf{x}_i^k$ is classified at stage k. These values represent the empirical risk "savings" if the observation $\mathbf{x}_i$ is classified at stage k as opposed to the worst case outcome. In this case, all sensor measurements are acquired incurring a penalty of $\sum_{i=1}^{K} c_k$ and the observation is incorrectly classified. The empirical penalty of classifying the observation $\mathbf{x}_i$ at stage k can therefore be expressed $1 + \alpha \sum_{i=1}^{K} c_k - \pi_i^k$.

Theorem 3.1. *The risk in* (7) *is equal to* (6).

Proof. We transform the risk in (7) with respect to the "savings" gained:

$$R(g^1, \ldots, g^K, \mathbf{x}, y) = 1 + \alpha \sum_{k=1}^{K} c_k$$
$$- \sum_{k=1}^{K} \pi^k \left(\mathbb{1}_{g^k(\mathbf{x}_i^k) \leq 0} \prod_{j=1}^{k-1} \mathbb{1}_{g^j(\mathbf{x}_i^j) > 0} \right).$$

$$R(g^1, \ldots, g^K, \mathbf{x}, y) = \underbrace{1 + \alpha \sum_{k=1}^{K} c_k}_{\substack{\text{maximum} \\ \text{possible} \\ \text{cost}}} - \underbrace{\sum_{k=1}^{K} \pi^k}_{\substack{\text{savings of} \\ \text{all stages}}} + \max_{k \in \{1, \ldots, K\}} \left[\underbrace{\sum_{j=1}^{k-1} \pi^j \mathbb{1}_{g^j(\mathbf{x}^j) > 0}}_{\substack{\text{savings lost from} \\ \text{stages before } k}} + \underbrace{\left(\sum_{j=k+1}^{K} \pi^k \right) \mathbb{1}_{g^k(\mathbf{x}^k) \leq 0}}_{\substack{\text{savings lost for} \\ \text{stages after } k}} \right] \tag{6}$$

The product of indicators can be expressed as a minimization over the indicators, or equivalently a maximization over negative indicators:

$$R(g^1,\ldots,g^K,\mathbf{x},y) = 1 + \alpha\sum_{k=1}^{K} c_k + \sum_{k=1}^{K}\Big(\pi^k$$
$$\max\left(-\mathbb{1}_{g^k(\mathbf{x}^k)\le 0}, -\mathbb{1}_{g^1(\mathbf{x}^1)>0}, \ldots, -\mathbb{1}_{g^{k-1}(\mathbf{x}^{k-1})>0}\right)\Big).$$

Next we change the sign of the indicators by changing the inequality directions:

$$R(g^1,\ldots,g^K,x,y) = 1 + \alpha\sum_{k=1}^{K} c_k - \sum_{k=1}^{K}\pi^k +$$
$$\sum_{k=1}^{K}\pi^k \max\left(\mathbb{1}_{g^k(\mathbf{x}^k)>0}, \mathbb{1}_{g^1(\mathbf{x}^1)\le 0}, \ldots, \mathbb{1}_{g^{k-1}(\mathbf{x}^{k-1})\le 0}\right)$$

Note that this form of the empirical risk is a maximization of linear functions, and substituting the indicators with convex surrogates yields a convex upper-bounding function. We further simply this expression to the form presented in (6) by taking advantage of dependencies of the indicator functions (see Supplementary for additional details). □

The reformulated empirical risk in (6) has the following interpretation. If we fix a k in the maximization term then, for an example $\mathbf{x}$ and decisions $g_1,\ldots,g_K$, we incur the penalty of the maximum possible cost minus the savings of all stages, plus the savings lost from stages before k and savings lost for stages after k. Therefore, for a fixed k, this empirical risk is a linear combination of indicators as opposed to a product of indicators as in (7). Recall that for a particular $\mathbf{x}$, only a single $S^{k^*}(\mathbf{x}^{k^*})$ is 1, and this k^* is the maximizer in (6).

The empirical risk formulation in (6) has a distinct advantage over the product of indicator formulation (7). Consider the upper-bounding convex surrogate function $L(z) \ge \mathbb{1}_{z\le 0}$. Replacing indicators with the surrogate function $L(\cdot)$ in (7) yields a bilinear expression, a fundamentally difficult optimization problem. In contrast, by expressing the risk as a maximization of sums of indicator functions, replacing the indicator functions in (6) with a convex surrogate functions yields a globally convex upper-bounding surrogate to the empirical risk function. We denote the upper-bounding risk with surrogate $L(\cdot)$ in place of the indicator function as $\hat{R}_L(g^1,\ldots,g^K,\mathbf{x},y)$ and the resulting convex optimization problem over the training set and a suitable family of functions $\mathcal{G}$:

$$\min_{g^1,\ldots,g^K\in\mathcal{G}^K}\sum_{i=1}^{N}\hat{R}_L(g^1,\ldots,g^K,\mathbf{x}_i,y_i) \qquad (9)$$

4 Learning Sequential Decisions as a Linear Program

For an upper-bounding convex surrogate function, we use a hinge-loss function $L(z) = max\,(0, 1-z)$. Replacing indicators in (6) with hinge-loss functions yields a linear program upper-bound of the empirical risk minimization problem.

Proposition 4.1. *For $L(z) = \max(0, 1-z)$ and $\mathcal{G}^K$ limited to linear functions of the data, the problem in (9) is equivalent to the linear problem:*

$$\min_{\substack{g^1,\ldots,g^K,\gamma_1,\ldots,\gamma_N\\ \beta_1^1,\ldots,\beta_N^K,\kappa_1^1,\ldots,\kappa_N^K}} \sum_{i=1}^{N}\gamma_i, \quad \textit{subject to:} \qquad (10)$$
$$\sum_{j=1}^{k-1}\pi_i^j\kappa_i^j + \left(\sum_{j=k+1}^{K}\pi_i^j\right)\beta_i^k \le \gamma_i,$$
$$1 - g^k(\mathbf{x}_i^k) \ge \beta_i^k,\ 1 + g^k(\mathbf{x}_i^k) \ge \kappa_i^k$$
$$\beta_i^k \ge 0,\ \kappa_i^k \ge 0,\quad \forall k\in[K],\ \forall i\in[N]$$

To convert the maximization over k in (6) to a set of linear constraints in (10), we introduce auxiliary variables γ^i. Similarly, to express the hinge loss, we introduce the auxilary variables, β_i^k, κ_i^k and their corresponding contraints. For simplicity of notation, we eliminate the constant terms in the objective of (6). We restrict the family of rejection decision functions $\mathcal{G}^K$ to be linear functions, however non-linear functions can also be trained with the proposed linear program through the use of expanded basis functions, $\phi(\mathbf{x})$. (See Suppl. for more details)

This linear programming formulation has multiple advantages over the existing non-convex alternating optimization approach [1]. The proposed program is a convex minimization problem, so a global optimum can efficiently be found. In contrast, previous approaches to solving problems of this form have only been shown to converge to a local minimum [6, 7] or cannot even guarantee convergence of any form [1]. As a result, these approaches rely on random initialization to improve performance, decreasing repeatability and reliability while increasing computation time. Finally, the proposed approach is computationally efficient when compared to non-convex approaches. Only a single linear program needs to be solved to return a solution, whereas alternating approaches require repeatedly solving supervised learning problems. The linear program is of similar order complexity compared to each of iteration of alternating optimization approaches. The number of variables in the linear program is of the order $O(KN)$, whereas each iteration of the alternating approach proposed in [1] requires solving K supervised learning problems with N training examples in each problem. Furthermore, the linear program can be efficiently solved with state of the art primal-dual methods, with an expected number of iterations $O(\sqrt{n}\log n)$, where n is the number of variables [23]. Finally, stochastic subgradient descent methods can be shown to converge to the global minimum [24], allowing for the sequential decision functions to be learned in the case where the training data is not available in aggregate and is instead only available in a streaming or batch setting.[4]

Budget Constraints: Our goal is to learn a set of sequential decision functions that minimize classification error subject to an average budget constraint. In order to learn decisions for different average budgets, we sweep over values of α, yielding multiple decision functions of varying error rates and average budgets. Note that a system matching a desired budget may not be learned, however any point in the convex hull of the error/budget points from learned systems is achievable by a randomized system. We therefore take the lower convex hull of the points in the space of average error vs. average cost to learn decision systems for any average budget constraint.

VC-Dimension: As shown in [7, 1], the VC-dimension of cascades is relatively small, growing on the order of $K\log(K)D$, where D is the maximum VC-dimension of the rejection decision functions $g^1,\ldots,g^K$ [25]. Intuitively, this implies that the rejection cascade does not dramatically increase complexity and that generalization error of the entire classification system is comparable to the generalization error of each individual classifier $d^1,\ldots,d^K$ (see [25, 7] for an in-depth analysis).

Regularization and Kernelization: Two issues that arise in the linear program formulated are whether the solution is unique and whether overfitting occurs. In this paper, we focus on linear rejection functions which generally avoid these problems. If the set of training examples is full-rank with respect to the dimension of the data and the costs $c_1,\ldots,c_K$ are all nonzero, the solution is unique. Due to the limited complexity of the linear function class and large training sets with respect to the dimension of the data, overfitting tends not to occur. However, in the case where uniqueness and overfitting arise, the natural solution is to include regularization in the objective function, such as the $\mathbf{L}^2$ norm. Regularization immediately removes non-unique solutions, with the optimal solution now the minimum-norm solution of the unregularized problem (for sufficiently small regularization coefficients). The $\mathbf{L}^2$ regularization term in combination with hinge-losses also allows trade-off between decision "error" and margins, preventing overfitting. Furthermore, addition of the $\mathbf{L}^2$ norm to the objective allows the problem to be kernelized, as the dual problem is entirely in the space of inner products $\phi(\mathbf{x}^k)\phi(\mathbf{x}^k)$ for some expanded basis function $\phi(\cdot)$. Although the problem can be kernelized by transforming the optimization from a linear program to a quadratic program, empirically we find linear functions to be sufficiently powerful for strong performance without the need for regularization.

5 Experiments

We compare our LP approach to the discriminative myopic strategy and non-convex (alternating minimization) algorithm presented in [1].

Discriminative Myopic Strategy: The discriminative myopic strategy rejects observations by thresholding classification confidence at each stage:

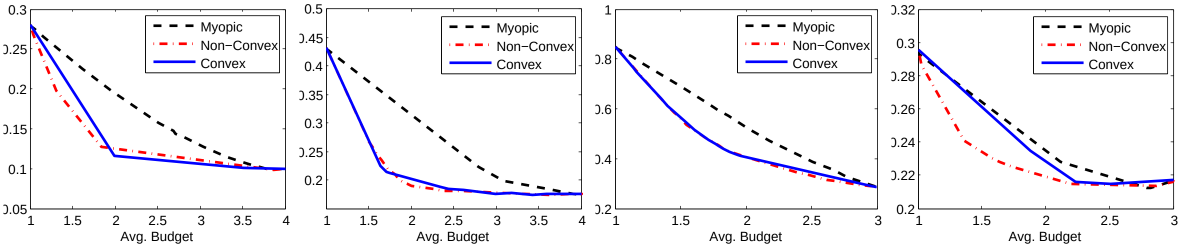
$$g^k_{myop} = \begin{cases} -1 & \text{if } \sigma_{d^k}(\mathbf{x}^k) \le t^k \\ 1 & \text{otherwise} \end{cases}$$

where $\sigma_{d^k}(\mathbf{x}^k)$ is the confidence of the classifier d^k on observation $\mathbf{x}^k$ and t^k is a constant threshold. In practice, we fix choose the threshold t^k at each stage k to reject a constant fraction of examples. This strategy does not consider future cost when rejecting, instead looking only at current uncertainty and is therefore considered myopic [1].

Alternating Minimization Algorithm: The non-convex algorithm attempts to minimize the empirical risk of the system as formulated in (7) using alternating minimization [1]. After a random initialization, the algorithm attempts to optimize each rejection decision g^k by fixing all other rejection decision functions and minimizing the empirical risk (7). The resulting optimization problem for learning

[4] Although training with streaming data is possible, LP packages tend to solve the problem faster than stochastic subgradient descend methods, so online training methods are not employed in the experiments.

Dataset	Classes	Training Size	Test Size	Stage 1	Stage 2	Stage 3	Stage 4
synthetic	2	1000	1000	Sensor 1	Sensor 2	Sensor 3	-
MNIST	10	60000	10000	4×4 image	7×7 image	14×14 image	28×28 image
landsat	6	4435	2000	Band 1	Band 2	Band 3	Band 4
letter	26	16000	4000	Pixel Count	Moments	Edge Features	-
pima	2	768	-	Weight, Age,...	Glucose	Insulin	-



(a) MNIST (b) landsat (c) letter (d) pima

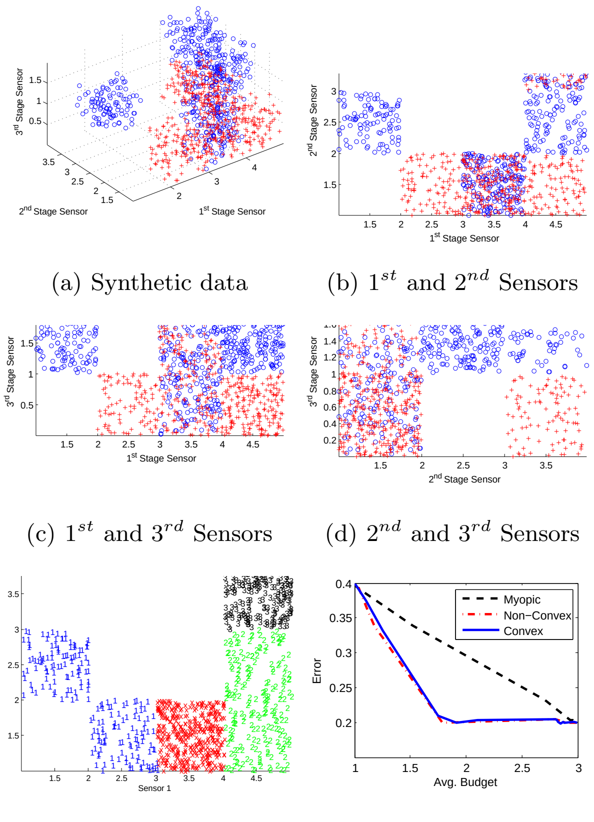
Figure 3: Comparison of error vs. average budget trade-off between a myopic approach, a non-convex optimization approach, and our linear programming algorithm. Our linear programming approach clearly out performs the myopic approach, and generally matches or exceeds the non-convex approach with the added benefit of reduced computational cost, repeatability, and guaranteed convergence. In the case of the pima dataset, the LP is outperformed by the non-convex approach for small budgets due to the discreteness of the first stage data.

each g^k is equivalent to a weighted binary supervised learning problem. As with our linear program, this algorithm attempts to minimize the empirical risk, however convergence and global optimality cannot be guaranteed. Additionally, this algorithm can be computationally expensive, as multiple initialization and passes through the system may be required. As in the LP, the average system budget is controlled by a trade-off parameter α. In the experimental results shown, for each parameter α, the alternating optimization algorithm is randomly initialized 5 times, with each initialization running through the system for 10 iterations (or fewer if the system converges and ceases to change between iterations). Training of each rejection function g^k is done by solving a weighted logistic regression problem.

Performance Metric: To evaluate performance of the sequential decision systems, we compare average acquisition cost vs. system error. For the myopic approach, this is achieved by sweeping the threshold t^k. For both our linear programming approach and the alternating minimization algorithm, the system is trained for varying values of α, yielding systems of varying average acquisition cost and error. Small values of α lead to a system with lower error rates but higher average acquisition cost, whereas large values of α result in systems with low average acquisition cost at the expense of increased system error. The lower convex hull of the learned systems is shown, as any average budget and error in the convex hull of systems is achievable through a randomized system (where each observation is randomly sent to one of the systems with varying weight).

Synthetic Example: To demonstrate the performance of our algorithm, we first experiment on a synthetic dataset. This dataset is based on the synthetic dataset presented in [1]. We need to extend this experiment to three dimensions. In the two dimensional case, the regularized version of our algorithm is exactly equivalent to the algorithm presented in [1] trained using an SVM, as no alternating optimization required. As shown in Fig. 2(e), observations require different sensor measurements to be correctly classified. Some data cannot be classified using any sensors, denoted by red x in the figure. The goal when learning sequential decision processes is to learn a system that does not acquire new features for the x observations while acquiring the necessary features to classify the rest of the data. The myopic approach does not take into account the future performance of the system, and therefore acquires measurements for thex observations, whereas the convex and non-convex decision systems do not acquire new features for these observations, reducing budget while mainaining classification accuracy.

Datasets: In addition, we compare performance of the sequential decision systems on 4 real world datasets used in [1]. For all examples, we assume the cost of acquiring each new feature is 1, and therefore the average cost of the system is the number of features acquired. For the MNIST dataset, lower quality sensors are simulated by downsampling the original 28×28 images to resolutions of 4×4, 7×7, and 14×14. The goal is to correctly classifying the



(a) Synthetic data

(b) 1^{st} and 2^{nd} Sensors

(c) 1^{st} and 3^{rd} Sensors

(d) 2^{nd} and 3^{rd} Sensors

(e) Required Sensor

(f) Error vs. Budget

Figure 2: (a) Synthetic dataset used to compare performance. (b,c,d) Projection of the data along different axes. (e) The numbers represent the sensor measurement required to distinguish between the two classes (for example, an observation labeled 2 requires the second sensor distinguish between classes). Points marked by x represent data in a region where classification is never better than random, regardless of the sensor measurements acquired. (f) Comparison of the error vs. budget tradeoff of the myopic approach, non-convex approach, and LP approach. The LP approach is clearly superior to the myopic approach, and matches the performance of the non-convex approach which require a significant number of random initializations.

digit using the lowest resolution sensor possible. The 3 other datasets are from the UCI repository [26]. The landsat dataset consists of 3 × 3 satellite images of the same area at four different hyperspectral bands, with a goal of correctly classifying the type of soil imaged. The letter dataset consists of features extracted from hand written digits, with the first five features generated from position and pixel counts, the next 7 features in the second stage correspond to more complex features such as spatial moments, and the final 4 features in stage 3 correspond to the most complex features, such as edge based features. The objective for the pima dataset is to diagnose diabetes, with patient history information in the first stage, a glucose test in the second stage, and an insulin test in the final stage. Note that for the pima dataset, we show performance on the entire dataset due to the limited set size and lack of a benchmark training/test split. For the MNIST, landsat, and letter datasets, the benchmark splits are used, with performance shown on the test sets. For all of these datasets, the classifiers at each stage $(d^1, \ldots, d^K)$ are linear functions trained using a standard multiclass logistic regression approach.

Dataset	Target Error	Myopic	Non-Convex	Convex
synthetic	0.21	96%	39%	37%
MNIST	.11	81%	51%	33%
landsat	.19	71%	42%	44%
letter	.4	73%	51%	51%
pima	.22	73%	48%	51%

Table 1: Average percentage of the budget required to achieve a desired error rate. The target rate is chosen to be close to the error achieved using the entire set of features (the target error rates are approximately 95% of the improvement gained using all features compared to using only the initial features). The percentage of the budget required is with respect to the maximum budget. For example, if there are 3 stages and a budget of 50% indicates that on average, each example gains one additional feature in order to achieve the target error.

Discussion: As seen in Figs. 2 and 3 and Table 1, the linear programming formulation clearly outperforms the myopic approach. In general, the linear programming approach matches or exceeds the performance of the non-convex optimization approach while offering numerous advantages, as discussed in Section 4. Only in the case of the pima dataset does the non-convex approach appear to outperform the linear programming approach for a small set of budget values. This is due to the fact that the linear programming approach does not produce a system for small budget values (apart from a system that never acquires new measurements, where $g^1 < 0$). For smaller budgets, the system is instead created by randomly sampling between a system with a higher budget and a system that never acquires new sensor measurements. We believe this effect arises because the first stage of the pima dataset consists solely of discrete features, and therefore partitioning of the data into arbitrary-sized groups in the LP setting (done by increasing the margins of some examples while decreasing the margins of others) is difficult. This is supported empirically, as once the budget increases beyond 2 (where on average each example sees the second real-valued sensor), performance of the system matches the non-convex approach.

Acknowledgements

This research was partially supported by NSF Grants 0932114 and 1330008.

References

[1] K. Trapeznikov and V. Saligrama. Supervised sequential classification under budget constraints. In *AISTATS*, 2013.

[2] M. Chen, Z. Xu, K.˜Q. Weinberger, O. Chapelle, and D. Kedem. Classifier cascade: Tradeoff between accuracy and feature evaluation cost. In *AISTATS*, pages 235–242, 2012.

[3] K Trapeznikov, V Saligrama, and D A Castañon. Mulit-Stage Classifier Design. In *Machine Learning*, pages 1–24, 2013.

[4] Zhixiang Xu, Matt Kusner, Minmin Chen, and Kilian Q Weinberger. Cost-sensitive tree of classifiers. In *ICML*, pages 133–141, 2013.

[5] Nimrod Megiddo. On the complexity of polyhedral separability. *Discrete & Computational Geometry*, 3(1).

[6] Kristin P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2, 1993.

[7] Joseph Wang and Venkatesh Saligrama. Local supervised learning through space partitioning. In *NIPS*. 2012.

[8] Clodoaldo A.M. Lima, Andre L.V. Coelho, and Fernando J. Von Zuben. Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences*, 177(10):2049 – 2074, 2007.

[9] Support vector machines experts for time series forecasting. *Neurocomputing*, 51(0).

[10] Li-Ping Liu, Yang Yu, Yuan Jiang, and Zhi-Hua Zhou. TEFE: A Time-Efficient Approach to Feature Extraction. In *International Conference on Data Mining (ICDM)*, pages 423–432, 2008.

[11] P Viola and M Jones. Robust Real-time Object Detection. *International Journal of Computer Vision*, 4:34–47, 2001.

[12] C. Zhang and Z. Zhang. A Survey of Recent Advances in Face Detection. Technical report, Microsoft Research, 2010.

[13] Shihao Ji and Lawrence Carin. Cost-sensitive feature acquisition and classification. *Pattern Recognition*, 40(5):1474–1485, 2007.

[14] A Kapoor and E Horvitz. Breaking Boundaries: Active Information Acquisition Across Learning and Diagnosis. In *NIPS*, 2009.

[15] V S Sheng and C X Ling. Feature value acquisition in testing: A sequential batch test algorithm. In *ICML*, pages 809–816, 2006.

[16] M Bilgic and L Getoor. Voila: Efficient feature-value acquisition for classification. In *Association for the Advancement of Artificial Intelligence (AAAI): Conference on Artificial Intelligence*, volume 22, page 1225, 2007.

[17] V.˜B. Zubek and T.˜G. Dietterich. Pruning Improves Heuristic Search for Cost-Sensitive Learning. In *ICML*, pages 19–26, 2002.

[18] P Kanani and P Melville. Prediction-time Active Feature-Value Acquisition for Cost-Effective Customer Targeting. In *NIPS*, 2008.

[19] He He, Hal Daume III, and Jason Eisner. Imitation learning by coaching. In *NIPS*, pages 3158–3166, 2012.

[20] Trevor Darrell Sergey Karayev, Mario Fritz. Dynamic feature selection for classification on a budget. In *ICML: Workshop on Prediction with Sequential Models*, 2013.

[21] Róbert Busa-Fekete, Djalel Benbouzid, and Balázs Kégl. Fast classification using sparse decision dags. In *29th ICML*, 2012.

[22] Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: a sequential approach to sparsity. In *Machine Learning and Knowledge Discovery in Databases*, pages 375–390. Springer, 2011.

[23] Kurt M. Anstreicher, Jun Ji, Florian A. Potra, and Yinyu Ye. Probabilistic analysis of an infeasible-interior-point algorithm for linear programming. *Math. Oper. Res.*, 24(1):176–192, January 1999.

[24] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

[25] Eduardo D Sontag. VC Dimension of Neural Networks. In *Neural Networks and Machine Learning*, pages 69–95, 1998.

[26] A. Frank and A. Asuncion. UCI machine learning repository, 2010.