Supervised Sequential Classification Under Budget Constraints

Kirill Trapeznikov
Boston University

Venkatesh Saligrama
Boston University

Abstract

In this paper we develop a framework for a sequential decision making under budget constraints for multi-class classification. In many classification systems, such as medical diagnosis and homeland security, sequential decisions are often warranted. For each instance, a sensor is first chosen for acquiring measurements and then based on the available information one decides (rejects) to seek more measurements from a new sensor/modality or to terminate by classifying the example based on the available information. Different sensors have varying costs for acquisition, and these costs account for delay, throughput or monetary value. Consequently, we seek methods for maximizing performance of the system subject to budget constraints. We formulate a multi-stage multi-class empirical risk objective and learn sequential decision functions from training data. We show that reject decision at each stage can be posed as supervised binary classification. We derive bounds for the VC dimension of the multi-stage system to quantify the generalization error. We compare our approach to alternative strategies on several multi-class real world datasets.

1 Introduction

We develop supervised learning algorithms for learning multi-class sequential classifiers. The need for sequential rules arise because we are limited by a budget in acquiring measurements. So we need to learn rules that tradeoff prediction error against acquisition costs. Such problems appear in many applications including homeland security and medical diagnosis. The goal in these scenarios is to classify examples with low cost sensors and limit the number of examples for which more expensive or time consuming informative sensor is required.[1] Consequently, we associate each stage with a new sensing modality with higher costs associated with later stages.

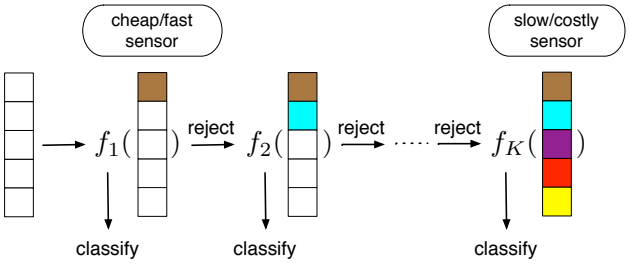


Figure 1: Multi-Stage System consists of K stages. Each stage is a classifier with a reject option. The system incurs a penalty of δ_{k+1} at kth stage if it rejects to seek more measurements. The kth classifier only sees the first k sensing modalities in making a decision.

An important aspect of homeland security and medical diagnosis applications is that each sensors/modality produce high dimensional measurements (such as images (X-Rays etc)). So, not only are the underlying distributions for the sensor measurements under different classes not known, but impossible to estimate from training data due to the inherent "curse of di-

Preliminary work. Under review by AISTATS 2013. Do not distribute.

[1] Modern passenger screening systems for explosives detection employ a suite of sensors such as X-ray backscatter scanners (cheap & fast), millimeter wave imagers (expensive & low-throughput), magnetometers, video, IR imagers in different bands, and/or physical (human) search. Such systems must maintain a throughput constraint in order to keep pace with arriving traffic. In clinical diagnosis, doctors use a suite of sensors for detecting and assessing the severity of (breast cancer) mammographic mass lesions (malicious or benign) including genetic markers, CT images from different views, 3-D CT tomographic reconstructions, optical tomography imaging, ultrasound imaging, elastography imaging, manual palpation, and biopsy, among others. Many of these sensors provide imagery input for individual human radiologist scoring. The different sensing modalities have diverse costs, in terms of health risks (radiation exposure) and monetary expense.

mensionality."

To deal with these issues we adopt a supervised discriminative learning approach by directly learning sequential decision rules from a suitable class. Specifically, we formulate a novel Empirical Risk Minimization (ERM) objective function over the class of sequential decision rules. We train sequential decision rules from a set of training examples in which measurements from all the sensing modalities as well as the ground truth labels are available. Our goal is to learn *sequential reject classifiers* that reduces cost of measurement acquisition and error in the *prediction (or testing) phase*. This concept of using training examples from all modalities for training decision rules is evidently not new and is known as *Prediction Time Cost Reduction approach* ([11]).

In contrast to much of the existing literature the novelty of our learning scheme is in emulating the stage-wise optimization objective of a closely related Markov Decision Problem (MDP). This MDP problem, which is based on knowledge of underlying probability models for sensor measurements, seeks to minimize multi-stage risk over all measurable decision rules. It turns out that the MDP problem can be decomposed into minimization of stage-wise risk functions which incorporate costs from future stages. We emulate this decomposition in the empirical setting by formulating a stage-by-stage empirical risk and seek to minimize this risk over a parametric class of decision strategies.

We also derive bounds for generalization error for our sequential decision rules. We consider the binary classification setting for simplicity. In this setting our system turns out to be a Boolean fusion of binary decision functions. Using this insight, we derive an upper bound on the VC dimension of the multi-stage reject classifier. We show that the VC dimension of a K-stage system grows as $K \log K$ times the maximum complexity of any stage. Our approach also enjoys other advantages. We can utilize "black box" classifiers that are pre-programmed into a sensing modality. In this context, our problem reduces to learning reject regions at each stage assuming that there is a confidence associated with each decision. In this setting, the complexity of our system only depends on the complexity of the highest reject region which is typically not very high.

1.1 Related Work

The subject of this paper is not new and has been studied in the Machine Learning community as early as [15]. Our work is closely related to the so called prediction time active feature acquisition (AFA) approach in the area of cost-sensitive learning. The goal there is to make sequential decisions of whether or not to acquire a new feature to improve prediction accuracy. Conventional methods can be divided into two categories:

Generative & Parametric Modeling: In a Bayesian setting, probability models are either known or the data is sufficiently low-dimensional that these models can be reliably estimated. Under these assumptions, [10, 12] model the decision process and infer feature dependencies while taking acquisition costs into account. [17, 3, 23] study strategies for optimizing decision trees while minimizing acquisition costs. The construction is usually based on some purity metric such as entropy. [11] propose a method that acquires an attribute if it increases an expected utility. However, all these methods require estimating a probability likelihood that a certain feature value occurs given the features collected so far. While surrogates based on classifiers or regressors can be employed to estimate likelihoods, this approach requires discrete, binary or quantized attributes. 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. Instead, we develop a discriminative learning approach and formulate a multi-stage empirical risk optimization problem to reduce measurement costs and misclassification errors.

Discriminative Learning Approaches: Our approach is the first framework to analyze the design of multi-class sequential decision systems in a non-Bayesian setting. Multiple stages of margin based reject classifiers have been considered in a time efficient feature extraction (TEFE) algorithm by [14] in the context image classification. This method employs a sequence of SVMs, each operating on features of increasing computational complexity. The main contribution of the work in [14] is in efficient training of each stage; the solution of previous stage is used to initialize SVM optimization problem of the following stage. However, the method uses a myopic strategy that does not take into account the performance of the entire system in learning the decisions. We compare this myopic strategy in the Experiments section and demonstrate significantly better performance. Besides the method mentioned above, we are not aware of any other approaches that seek to reduce measurement budget in a multi-stage and multi-class setting and are able to handle large dimensional training data.

The detection cascade (popular in object detection) can be considered as a special case of our multi-stage sequential reject classifiers (MSRC). There is extensive literature on cascade design (see [22, 4] and references therein) but most cascades roughly follow the set-up

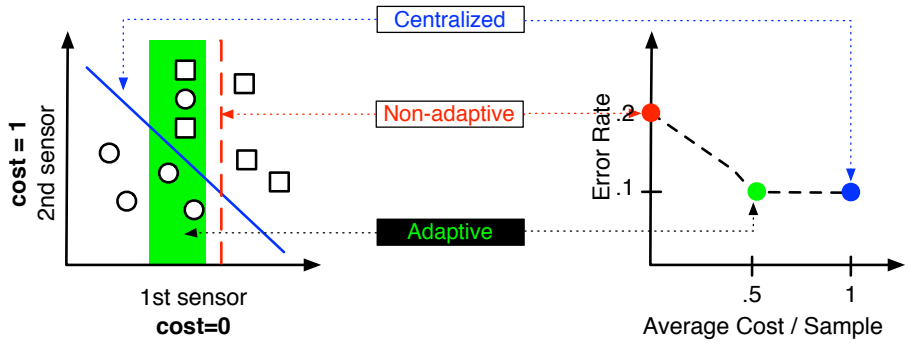


Figure 2: (Advantage of an adaptive 2 stage strategy: 10 samples, binary (squares, circles). The red line is the optimal decision when using only 1st stage modality. The blue line is optimal if using both. (2nd stage) The curve is classification error vs. average measurement cost. The red point corresponds to a non-adaptive strategy that uses only sensor 1 in making a decision. The blue is a centralized strategy that classifies using both modalities. The green is an adaptive reject strategy. The samples outside the green region are classified using only the first modality, and samples inside the region are rejected to stage 2 and are classified using both modalities. Note that blue and green have the same error, while the reject strategy (green) has to use 2nd stage sensor only for $\frac{1}{2}$ of examples, reducing the cost by a factor of 2.

introduced by [19] to reduce computation cost during detection. The fundamental differences between detection cascades and MSRC is the architecture. Detection cascades are primarily concerned with binary classification problems. They make partial binary decisions at each stage, delaying a positive decision until the final stage. In contrast, MSRCs can deal with multi-class problems and can make classification decisions at any stage. Conceptually, this distinction requires a fundamentally new approach; detection cascades work because their focus is on unbalanced problems with few positives and a large number of negatives; and so the goal at each stage is to admit large false positives with negligible missed detections. In contrast, our scheme at each stage is composed of a multi-class classifier as well as a rejection decision.

Sequential decisions have also been considered in such areas as network intrusion detection ([8, 13, 6]) and reducing the size of classifier ensembles ([7, 20]). However, these methods are domain specific and do not easily extended to the budget constrained setting.

At a technical level our system consists of a sequence of reject classifiers. Topic of reject classifiers have been considered in the Bayesian framework by [5]. More recently in the non-bayesian setting, researches, [21, 2, 16, 9], define a reject region within a small distance to the separating hyperplane in the SVM framework. We similarly define our reject region in relation to the decision boundary but allow it to be of higher complexity.

2 Problem Statement

Let $(\mathbf{x}, y) \in \mathcal{X} \times \{1, 2, \dots C\}$ be distributed according to an unknown distribution $\mathcal{D}$. A data point has K features, $\mathbf{x} = \{x_1, x_2, \dots, x_K\}$, 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: $\mathbf{x}^k = \{x_1, x_2, \dots x_k\}$. 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 decison with a reject option, f^k. It can either classify an example, $f^k(x^k) : \mathcal{X}^k \rightarrow \{1, 2, \dots, C\}$, or delay the decision until the next stage, $f^k(x^k) = r$ and incur a penalty of δ^{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. Define the system risk to be,

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

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 has been rejected up to kth stage.

$$R_k(\mathbf{x}^k, y, f^k) = \begin{cases} \delta^{k+1}, & f^k(\mathbf{x}^k) = r \\ 1, & f^k(\mathbf{x}^k) \neq y \ \wedge \ f^k(\mathbf{x}^k) \neq r \end{cases}$$

If x is active and is misclassified, the penalty is 1. If it is rejected then the system incurs a penalty of δ^k, and the state variable for that example remains at 1.

$$S^{k+1}(\mathbf{x}^{k+1}) = \begin{cases} S^k(\mathbf{x}^k), & f^k(\mathbf{x}^k) = r \\ 0, & \text{else} \end{cases}, \ S^0 = 1 \quad (2)$$

2.1 Markov Decision Problem (MDP)

In this section, we will digress from the discriminative setting and analyze the problem under the assumption

that the underlying distribution $\mathcal{D}$ is known. In doing so, we hope to discover some fundamental structure that will simplify our empirical risk formulation in the next section.

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

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

If we allow arbitrary decision functions then we can equivalently minimize conditional risk,

$$\min_{f^1,\ldots,f^K} \mathbf{E}\left[R(f^1,\ldots,f^K,\mathbf{x}^k,y) \mid \mathbf{x}\right] \tag{4}$$

This problem—by appealing to dynamic programming—remarkably reduces to a single stage optimization problem for a modified risk function. To see this, we denote the cost-to-go,

$$\tilde{\delta}^k(\mathbf{x}^k) = \delta^{k+1} + \min_{f^{k+1}\ldots f^K} \mathbf{E}\left[\sum_{t=k+1}^{K} S^t(\mathbf{x}^t) R_t(f^t,\mathbf{x}^t,y) \mid \mathbf{x}^k, S^k(\mathbf{x}^k)=1\right] \tag{5}$$

and the modified risk functional,

$$\tilde{R}_k(\mathbf{x}^k,y,f^k,\tilde{\delta}^k) = \begin{cases} \tilde{\delta}^k(\mathbf{x}^k), & f^k(\mathbf{x}^k)=r \\ 1, & f^k(\mathbf{x}^k)\neq y \ \wedge \ f^k(\mathbf{x}^k)\neq r \end{cases}$$

and prove the following theorem (see Suppl. for proof),

Theorem 1. *The optimal solution $f^1, f^2, \ldots f^K$ to the multi-stage risk in Eq. 4 decomposes to single stage optimization,*

$$f^k = \arg\min_f \mathbf{E}\left[\tilde{R}_k(\mathbf{x}^k,y,f,\tilde{\delta}^k) \mid \mathbf{x}^k\right] \tag{6}$$

and the solution is:

$$f^k(\mathbf{x}^k) = \begin{cases} \hat{y}, & \bar{\mathrm{P}}(\mathbf{x}^k) > 1-\tilde{\delta}^k(\mathbf{x}^k) \\ reject, & \bar{\mathrm{P}}(\mathbf{x}^k) \leq 1-\tilde{\delta}^k(\mathbf{x}^k) \end{cases} \tag{7}$$

$$\hat{y} = \arg\max_j \mathrm{P}(y=j \mid \mathbf{x}^k),\ \bar{\mathrm{P}}(\mathbf{x}^k) = \max_j \mathrm{P}(y=j \mid \mathbf{x}^k)$$

The main implication of this result is that if the cost-to-go function $\tilde{\delta}(\mathbf{x}^k)$ is known then the risk $\tilde{R}_k(\cdot)$ is only a function of the current stage decision f^k. Therefore, we can ignore all of the other stages and minimize a single stage risk. Effectively, we decomposed the multi-stage problem in Eq. 4 into a stage-wise optimization in Eq. 6. [2]

[2]Note that the modified risk functional, $\tilde{R}_k$, is remarkably similar to R_k except that the modified reject cost $\tilde{\delta}^k(\mathbf{x}^k)$ replaces the constant stage cost δ^k. Also, consider the range for which $\delta^k(\mathbf{x}^k)$ is meaningful. If we have C classes then a random guessing strategy would incur an average risk of $1-\frac{1}{C}$. Therefore the risk for rejecting, $\tilde{\delta}^k(\mathbf{x}^k) \leq 1-\frac{1}{C}$ in order to be a meaningful option. The work in [5] contains a detailed analysis of single stage reject classifier in a Bayesian setting.

2.2 Stage-Wise Empirical Risk Minimization

In this section, we assume that the probability model $\mathcal{D}$ is no longer 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: $(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_N,y_N)$

We will take advantage of the stage-wise decomposition of the MDP solution in Theorem 1 and formulate an empirical version of the stage risk $\tilde{R}_k(\cdot)$ in Eq. 6. However, this requires an empirical estimate of the cost-to-go, $\tilde{\delta}^k(\mathbf{x}_i^k) \rightarrow \tilde{\delta}_i^k$, since we are not estimating probability models. Note that by definition, $\tilde{\delta}^k(\mathbf{x}_i^k)$ is a only function of $f^{k+1},\ldots,f^K$. So the cost-to-go estimate is conveniently defined by the recursion,

$$\tilde{\delta}_i^{k-1} = S_i^k \tilde{R}_k(\mathbf{x}_i^k, y_i, f^k, \tilde{\delta}_i^k) + \delta^k,\ \forall i \tag{8}$$

Now, we can form the empirical version of the risk in Eq 6 and optimize for a solution at stage k over some family of functions, $\mathcal{F}^k$.

$$f^k(\mathbf{x}^k) = \arg\min_{f\in\mathcal{F}^k} \frac{1}{N}\sum_{i=1}^{N} S_i^k \tilde{R}_k(y_i,\mathbf{x}_i^k,f,\tilde{\delta}_i^k) \tag{9}$$

Note, the stage-wise decomposition significantly simplifies the ERM. The objective in Eq. 9 is only a function f^k given δ_i^k and the state S_i^k. To minimize an empirical versions of a multi-stage risk in Eq. 4 is more difficult due to stage interdependencies.

Multi-class decision with a reject option: Recall that at each stage, $f^k(\mathbf{x}^k)$, is a $C+1$ decision function where the extra decision is due to the reject option. Because of this additional decision, minimizing the empirical risk at each stage is still difficult. In order to simplify the problem, we factorize the reject option from the multi-class decision.

Assume that at each stage, our system has a fixed stage classifier, $d^k : \mathcal{X}^k \rightarrow \{1,\ldots C\}$ and its associated confidence function $\sigma_{d^k} : \mathcal{X}^k \rightarrow \mathbb{R}^+$. $\sigma(\cdot)$ reports how confident $d^k(\cdot)$ is in classifying $\mathbf{x}^k$. Our choice for $\sigma(\cdot)$ (described in Sec. 3) is based on the absolute margin of a binary classifier, which evidently is a popular heuristic for confidence [2]. Using this reduction, we propose the following parameterization of a multi-class classifier with a reject option at each stage.

$$f^k(x^k) = \begin{cases} d^k(\mathbf{x}^k), & \sigma_{d^k}(\mathbf{x}^k) > g^k(\mathbf{x}^k) \\ reject, & \sigma_{d^k}(\mathbf{x}^k) \leq g^k(\mathbf{x}^k) \end{cases} \tag{10}$$

We designate $g(\cdot)$ as a rejector at stage k. The reject region is constructed by thresholding the confidence measure $\sigma(\cdot)$ by $g(\mathbf{x})$. In the space where $g(\mathbf{x})$ is small, few examples are rejected. In the space where $g(\mathbf{x})$ is large, rejection is high. Note that $g(\mathbf{x})$ varies with $\mathbf{x}$.

This dependence on $\mathbf{x}$ is important because it enables $g(\mathbf{x})$ to selectively reject specific regions in the space. Our choice in parameterization mimics the optimal reject region: $\max_j \mathrm{P}(y = j \mid \mathbf{x}^k) \leq 1 - \tilde{\delta}^k(\mathbf{x}^k)$. Recall that the optimal binary classifier is $\arg\max_j \mathrm{P}(y = j \mid \mathbf{x}^k)$. So the reject region is the space around the boundary whose size varies as a function of $\tilde{\delta}^k(\mathbf{x}^k)$.

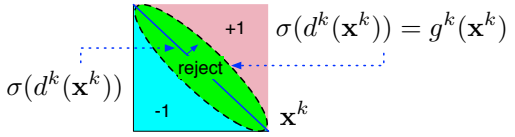


Figure 3: To illustrate our parametrization, consider a binary classification setting. $d^k(\cdot)$ is a hyperplane and the confidence $\sigma_{d^k}(\cdot)$ is the distance to this hyperplane. A possible reject region is constructed by thresholding the confidence by a rejector $g^k(\cdot)$. Note how the width of the reject region varies with $\mathbf{x}^k$ because g^k is a function of $\mathbf{x}^k$.

Furthermore, we can rewrite the empirical risk in Eq. 9 using our parametrization,

$$\tilde{R}_k(\mathbf{x}_i^k, y_i, d^k, g^k) = \underbrace{\mathbb{1}_{[d^k(\mathbf{x}_i^k) \neq y_i]}}_{\text{error penalty}} \underbrace{\mathbb{1}_{[\sigma_{d^k}(\mathbf{x}_i^k) > g(\mathbf{x}_i^k)]}}_{\text{not rejected}} + \underbrace{\tilde{\delta}_i^k}_{\text{cost to go}} \underbrace{\mathbb{1}_{[\sigma_{d^k}(\mathbf{x}_i^k) \leq g(\mathbf{x}_i^k)]}}_{\text{rejected}} \tag{11}$$

Next, If we use this simplified form and hold the rest of the system constant then minimizing Eq. 9 with respect to $g^k(\mathbf{x}^k)$ over a family of functions $\mathcal{G}^k$ reduces to a supervised learning problem: (see Suppl. for proof)

Lemma 2. *If $d^k(\mathbf{x})$, S_i^k and $\tilde{\delta}_i^k$ are held constant then minimization over $g(\cdot)$ in Eq 9 reduces to:*

$$g^k(\mathbf{x}^k) = \arg\min_{g \in \mathcal{G}^k} \sum_{i=1}^{N} S_i^k \, |w_i| \, \mathbb{1}_{[b_i(g(\mathbf{x}_i^k) - z_i) \leq 0]} \tag{12}$$

$$w_i = \mathbb{1}_{[d^k(\mathbf{x}_i^k) \neq y_i]} - \tilde{\delta}_i^k, \;\; z_i = \sigma_{d^k}(\mathbf{x}_i^k), \;\; b_i = \operatorname{sgn}[w_i]$$

This simplified problem closely resembles minimizing weighted binary misclassification error. The pseudo labels b_i play an important role. Note the weight w_i is the difference between the risk of the current stage $d^k(\cdot)$ and the cost of rejecting, $\tilde{\delta}_i^k$. The label b_i is $+1$, if it is more costly to classify $\mathbf{x}_i$ at present stage and -1 if the penalty for rejecting is higher than classifying. This optimization finds a rejector $g(\cdot)$ such that the examples of pseudo class $+1$ are rejected and examples of class -1 are classified. Pseudo class $+1$ consists of examples with higher misclassification risk than rejection cost. Recall that S_i^k are just binary variables indicating whether $\mathbf{x}_i$ is still active at stage k.

In summary, given $S_i^k, \tilde{\delta}_i^k, d^k(\mathbf{x}^k)$, to solve for the rejector $g^k(\mathbf{x}^k)$ requires finding a binary decision with pseudo labels b_i and weights $|w_i|$ on the training set with respect to the indicator loss offset by z_i's.

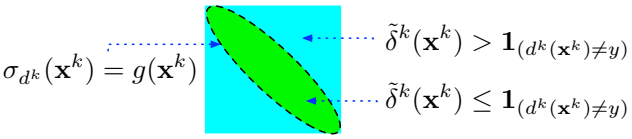


Figure 4: The figure illustrates the simplified optimization problem for $g^k(\mathbf{x}^k)$ in Lemma 2. The objective is to find a rejector function to fit the decision regions in the figure. The data in the green region has cost-go smaller then the risk of classifying at the current stage and therefore is to be rejected. The data outside the green has higher cost-to-go than misclassification risk and is to be not rejected.

3 Algorithm

In this section, using the simplified rejector subproblem from Lemma 2, we provide one possible implementation of the multi-stage in the setting of mulit-class to binary reduction and explain our stage-wise optimization. In our problem, we assume that we are either provided with stage classifiers $d^1, d^2, \ldots, d^K$ or train them a-priori. So our objective is to find the rejectors $g^1, g^2, \ldots g^{K-1}$ at each stage.

Embedding the reject option: Before we proceed to finding the rejectors, g^k, we explain how we implement pre-training of d^k. We utilize a well known technique for multi-class classification: reduction from multi-class to binary [1]. For each class $j \in \{1, 2, \ldots C\}$, we choose a binary codeword $\mathbf{c}_j \in \{+1, -1\}^M$ of length M. Let

$$\mathbf{h}(\mathbf{x}^k) = [h_1(\mathbf{x}^k) \; h_2(\mathbf{x}^k) \; \ldots \; h_M(\mathbf{x}^k)]$$

be a vector valued classifier such that $h_m : \mathcal{X}^k \to \mathbb{R}$. This approach reduces a multi-class problem to finding S binary classification functions, $h_m(\mathbf{x}^k)$, with respect to code labels c_{jm}. For each sub-problem m, we take the usual ERM approach and upper-bound the indicator error by a convex loss: $\mathbb{1}_{[z]} \leq \mathcal{L}[z]$ and fix a family of classifiers $\mathcal{H}^k$.

$$h_m^k(\mathbf{x}^k) = \arg\min_{h \in \mathcal{H}^k} \sum_{i=1}^{N} \mathcal{L}\left[c_{y_i m} h(\mathbf{x}_i^k)\right] \tag{13}$$

We use the logistic loss function, $\mathcal{L}[z] = \log(1 + \exp(-z))$ and set $\mathcal{H}^k$ to be a family of polynomial kernel classifiers [3]. Given an output $\mathbf{h}^k(\mathbf{x}^k)$, we use maximum projection decoding to assign a class estimate to the best matching codeword. We define a

[3] Polynomial kernel classifier of degree q is parametrized by a vector $\mathbf{a}$: $h(\mathbf{x}) = \sum_{i=1}^{N} a_i (\mathbf{x}_i^T \mathbf{x} + 1)^q$

stage classifier as

$$d^k(\mathbf{x}^k) = \arg \max_{j=\{1\dots C\}} \mathbf{c}_j^T \mathbf{h}^k(\mathbf{x}^k)$$

For example, in our implementation, we use one vs all coding. Here the length of the codeword is $M = C$ and, for class j, each element of the codeword $\mathbf{c}_j$ is -1 except that the jth position is $+1$, and $d^k(\mathbf{x}^k) = \arg\max_{j=\{1\dots C\}} h_j^k(\mathbf{x}^k)$.

For the confidence function, $\sigma(\cdot)$, we select an absolute maximum projection: $\left|\max_{j=\{1\dots C\}} \mathbf{c}_j^T \mathbf{h}^k(\mathbf{x}^k)\right|$. Our choice in $\sigma(\cdot)$ is inspired by an absolute margin of a binary classifier, $|h(\mathbf{x})|$, which is a popular heuristic measure of classifier confidence [2]. However, in a multi-class setting, we use the absolute value of the best matching projection onto the codeword as a measure of confidence instead of a single margin. For example, $\mathbf{c}^T\mathbf{h}(\mathbf{x}^k)$ is maximized when the classifier output matches the codeword exactly. A small value of the projection indicates that $\mathbf{h}(\cdot)$ has lower confidence in its classification.

Algorithm 1 Our Method

Input: $\{\mathbf{x}_i, y_i\}_{i=1}^N$, $\{d^k(\cdot)\}_{k=1}^K$, $\sigma_{d^k}(\cdot)$, $\{\delta^k\}_{k=1}^K$, P
Initialize: $S_i^k = 1, \forall(i,k)$
for $p = 1, 2, \dots, P$ **do**
  for $k = K-1, K-2, \dots, 1$ **do**
    Update $\tilde{\delta}_i^k$ according to Eq. 8
    Train $g^k(\mathbf{x}^k)$ according to Eq. 15
    Update S_i^k according to Eq. 2
  end for
end for
Output: for $k = 1, \dots K-1$,

$$f_k(\mathbf{x}^k) = \begin{cases} d^k(\mathbf{x}^k), & \sigma_{d^k}(\mathbf{x}^k) > g^k(\mathbf{x}^k) \\ reject, & \sigma_{d^k}(\mathbf{x}^k) \le g^k(\mathbf{x}^k) \end{cases}$$

Substituting multi-class to binary reduction into our parameterization in Eq. 10 yields a multi-class decision with a reject option:

$$\hat{y} = \arg\max_j \mathbf{c}_j^T \mathbf{h}^k(\mathbf{x}^k),\ \bar{h}(\mathbf{x}^k) = \max_j \mathbf{c}_j^T \mathbf{h}^k(\mathbf{x}^k)$$

$$f^k(\mathbf{x}^k) = \begin{cases} \hat{y}, & \left|\bar{h}(\mathbf{x}^k)\right| > g^k(\mathbf{x}^k) \\ reject, & \left|\bar{h}(\mathbf{x}^k)\right| \le g^k(\mathbf{x}^k) \end{cases} \quad (14)$$

Stage-wise optimization: To compute a rejector $g^k(\mathbf{x}^k)$, every stage except the kth is held constant. We upper-bound the $\mathbb{1}_{[z]} \le \mathcal{L}[z]$ in Lemma 2. For a convex loss $\mathcal{L}[\cdot]$ and a family polynomial kernels $\mathcal{G}^k$, the resulting optimization is a convex program,

$$g^k(\mathbf{x}^k) = \arg\min_{g\in\mathcal{G}^k} \sum_{i=1}^N S_i^k |w_i| \mathcal{L}\left[b_i(g(\mathbf{x}_i^k) - z_i)\right] \quad (15)$$

Once $d^1, d^2, \dots, d^K$ are precomputed, to train g^k's, we proceed by cyclic optimization of stages one at a time in reverse order: $g^{K-1}, g^{K-2}, \dots g^1$. Note that the weights w_i's capture the difference in risk between the current stage and the cost-to-go. The order of cyclic optimization is reversed due the recursive nature of the cost-to-go; δ_i^k is a function of the next stage. Initially, state variables S_i^k are set to one for all examples and stages. After the first pass through the stages outputs g^k's, the S_i^k's are updated. Using the updated state variables, g^k's are retrained in the second pass and so on. In experiments, we found that one pass is sufficient. For details refer to Algorithm 1. Here, P is the number of passes of cyclic optimization over stages.

Complexity of a Multi-Stage System: Using this particular parametrization, we can bound the VC-dimension of the entire system in the binary classification setting. (see Suppl. for proof)

Theorem 3. *Let $F(\mathbf{x})$ be the decision of our K-stage system in the binary class setting, and $F(\cdot) \in \mathcal{F}$. Let $\mathcal{H}^k$ be the family of stage classifiers, $\mathcal{G}^k$ is the family of rejectors at stage k*

$$\mathcal{VC}[\mathcal{F}] \le c_K \max_{k=1\dots K-1} \left\{\mathcal{VC}[\mathcal{H}^k] + \mathcal{VC}[\mathcal{G}^k], \mathcal{VC}[\mathcal{H}^K]\right\}$$

$$\text{where } c_K = 2(3K-2)\log\left(e(3K-2)\right) \quad (16)$$

Remarkably, the complexity increases as $K \log K$ in the number of stages K and is proportional to the most complex stage in the system. Also, note that since the rejector class $\mathcal{G}^k$ is typically of lower complexity than the stage classifiers, the overall complexity will be dominated by the VC dimension of stage classifiers $\max_k \mathcal{VC}[\mathcal{H}^k]$. However, if we are provided with "black box" classifiers d^k, then the complexity is bounded by $\max_k \mathcal{VC}[\mathcal{G}^k]$. In this case, $\sigma_{d^k}(\mathbf{x}^k)$ is simply an affine transformation of the class $\mathcal{G}$ which does not effect its VC dimension. ([18])

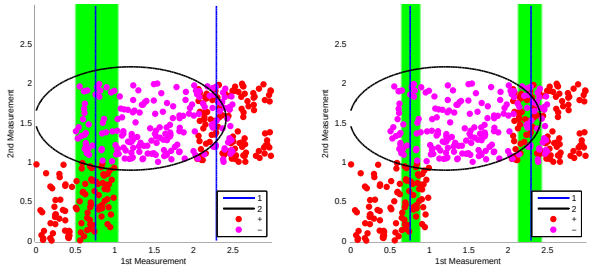
4 Experiments

Discriminative Myopic Strategy: For comparison, we consider a myopic strategy. This method is closely related to TEFE algorithm to [14]. The single stage multi-class classifier with reject option remains the same except that the confidence $\sigma_{d^k}(\mathbf{x}^k)$ is thresholded by a constant t_k to achieve a reject option:

$$f_{myop}^k(x^k) = \begin{cases} d^k(\mathbf{x}^k), & \sigma_{d^k}(\mathbf{x}^k) > t^k \\ reject, & \sigma_{d^k}(\mathbf{x}^k) \le t^k \end{cases} \quad (17)$$

The threshold t^k is chosen such that the kth stage will reject a constant fraction of the N examples in the training set. This strategy is completely myopic because t_k is chosen without considering the performance

of stages before or after the current stage. Disadvantage of such strategy is illustrated in Fig. 5.



(a) ours: error=.148 (b) myopic: error=.19

Figure 5: We display the decision boundaries of our method and the myopic approach for a fixed budget of 1.3. 1st stage classifier, d^1, is in **blue**. 2nd stage classifier, d^2, is **black**. The space that is rejected to 2nd stage is in **green**. Observe how our method only rejects the area around the first blue boundary. In contrast, myopic uniformly rejects samples around both boundaries even if the samples will be misclassifies at the second stage. This is because our strategy anticipates that the 2nd stage classifier cannot really classify examples around the second blue boundary and does not suffer the acquisition cost for those examples. This results in higher error for the same budget for myopic.

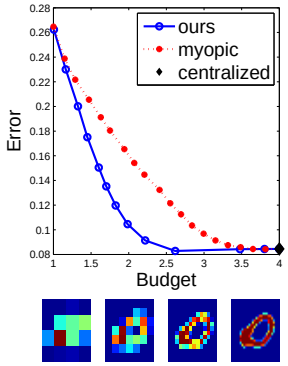


Figure 6: Here, we compare our method to myopic on the mnist data. We construct four stages of increasing resolution by averaging the original digit images. The experiment demonstrates the advantage of our approach. Also note that the performance of a full resolution sensor can be on achieved using a much lower resolution measurement.

Performance Metric: A natural way to evaluate performance of a sequential decision system is to show the trade-off between system error and average acquisition budget. Recall that our algorithm requires parameters: $\delta_1, \ldots, \delta_K$. δ_k can be thought as a sensor cost such that cost of being classified at stage k is $\sum_{l=1}^{k} \delta_l$. To achieve different operating points on the error vs budget curve, we can scale these parameters by a constant: $\alpha\delta_1, \ldots, \alpha\delta_K$. For small values of α, measurement costs are small so more examples are rejected down the stages resulting in higher average acquisition budget. For large α, acquisition costs are high resulting in smaller budget. If we sweep α, we generate the error vs budget operating points of our system. For the myopic method, we simply sweep the constant fraction rejected at each stage. In the experiments, we designate a centralized performance as a strategy that uses all sensors for every example. For more implementation details please refer to the supplemental material.

Datasets: We evaluate performance of our method on several datasets (see Table 1). Since for most datasets measurement cost is not specified, we consider uniformly increasing cost structure. A sample using the 1st stage sensor incurs a cost of 1. To reach the second stage sensor the cost is 2 and so on. So for a four stage system, if a sample passes all four stages, it incurs a cost of 4. To demonstrate the difference in decision regions between our and myopic strategies we use a binary two stage synthetic data with two dimensions corresponding to two sensors. (Fig. 5) For the another illustrative example, we convert a popular digit recognition data, MNIST, into a four stage decision system. (Fig. 4) We designate the full resolution 28x28 pixel image as the last stage. To simulate the first three stages of increasing sensor quality, we average the original image down to three resolution levels, 4x4, 7x7 and 14x14 pixels. The next four datasets are from UCI. (Fig. 7) Landsat data consists of 3x3 pixel neighborhoods taken from a satellite image at four different hyper spectral bands. The objective is to correctly classify the soil type. We set four bands to be the four stages in our system. Covertype deals with classifying forest cover type. We set the first stage to be 10 binary measurements indicating soil type. The second stage is 4 binary measurements indicating wilderness area type. The last (3rd) stage consists of 40 measurements such as aspect, elevation, etc. Letter consists of features extracted from handwritten images. The 1st stage are 5 features describing letterbox position and pixel counts. The 2nd stage consists of more complex features such as spatial moments. The last stage is most complex consisting of edge information. Pima is a dataset dealing with diabetes diagnoses with specified costs. The 1st stage consists of 6 simple tests (1 dollar each) such as body mass index, age and etc. Next stage consists of a glucose blood test (17 dollars). The last stage is an insulin test (23 dollars). Threat dataset contains images taken of people wearing various explosives devices. The imaging is done in three modalities: infrared (IR), passive millimeter wave (PMMW), and active millimeter (AMMW). All the images are registered. We extract many patches from the images and use them as our training data. A patch carries a binary label, it either contains a threat or is clean. Since PMMW and IR are the fastest modalities but also least informative, we set them to stages 1 and 2. Stage 3 is an AMMW sensor that requires raster scanning a person and is slow but also the most useful. Overall, simulations demonstrate the advantage of our approach over a myopic strategy. In many datasets, performance close to the centralized

Dataset	Size	Stage 1	Stage 2	State 3	Stage 4	# Classes
synthetic	4,000	Sensor 1	Sensor 2	..	..	2
mammogram	830	CAD feat's	expert rating	..	..	2
pima	768	weight, age, ..	glucose test	insulin test	..	2
threat	1230	PMMW image	IR image	AMMW image	..	2
covertype	581012	soils	wild. areas	elev, aspect, ..	..	7
letter	20000	pixel counts	moments	edge feat's	..	26
mnist	70000	4 x 4 image	7 x 7 image	14 x 14 image	28 x 28 image	10
landsat	6435	Band 1	Band 2	Band 3	Band 4	7

Table 1: Dataset Descriptions

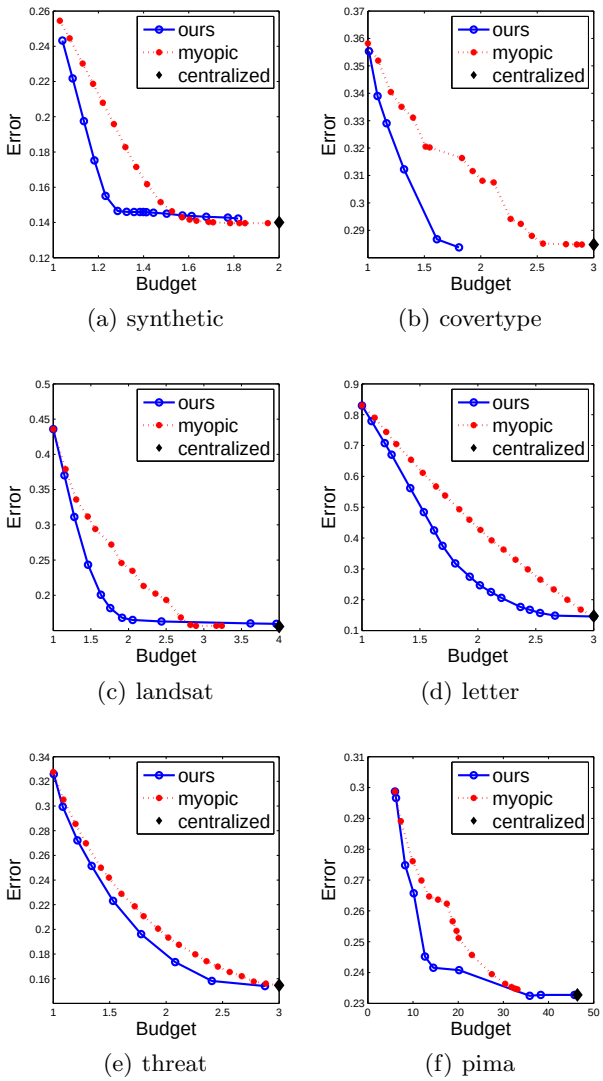
(best) strategy can be achieved with much lower average budget. Table 2 summarizes our experiments.

Dataset	Target Error	Myopic	**Ours**	Utility
synthetic	.147	52%	**28%**	
pima	.245	41%	**15%**	
threat	.16	89%	**71%**	
covertype	.285	79%	**40%**	
letter	.25	81%	**51%**	
mnist	.085	90%	**52%**	
landsat	.17	56%	**31%**	
mam	.173		**25%**	65%

Table 2: In this table we report an average percent of the maximum budget required to achieve the target error rate. The target rate is chosen to be close the error of the centralized strategy. Thus if there is a maximum of 2 stages and we obtain a value of 28% for our strategy it means that for only 28% of examples a 2nd stage is utilized without any degradation in error. Note that we only evaluate the expected utility approach on the mammogram dataset. The dimensionality of the other datasets is too high to parametrize the likelihood density reliably.

Parametric Expected Utility Strategy: To illustrate the difficulty of estimating likelihoods, we compare to an expected utility in [11]. An expected margin difference measures how a new attribute, if acquired, would be useful for an example. $U(x^k) = \sum_{x_{k+1}\in\mathcal{X}_{k+1}} \left| f^k(\mathbf{x}^k) - f^{k+1}([\mathbf{x}^k, x_{k+1}]) \right| \mathrm{P}(x_{k+1}|\mathbf{x}^k)$. An $\mathbf{x}^k$ is rejected to the next stage if its utility $U(\mathbf{x}^k) \geq t_k$ is greater than a threshold. Here, $\mathcal{X}_{k+1}$ denotes the possible values that x_{k+1} can take. Note this approach requires estimating $\mathrm{P}(x_{k+1}|\mathbf{x}^k)$[4], therefore the $(k+1)$th measurement has to be discrete or distribution needs to be parametrized. Also, it is unclear how to utilize utility in systems with more than two stages. Due to this limitation, we only compare this method on the mammogram dataset. (Fig. 8) Here, the second stage is an integer radiologist rating on the scale $1:5$ while the first stage is a three dimensional feature extracted form a CAD image.

[4]While there are many different ways to estimate a probability likelihood we used a Gaussian mixture due to its computational efficiency



(a) synthetic (b) covertype (c) landsat (d) letter (e) threat (f) pima

Figure 7: (a - f) illustrate error vs budget trade-off for our method and myopic various dataset. Clearly, our method is superior to myopic and can achieve performance of a centralized classifier (black diamond) with a significantly lower acquisition budget.

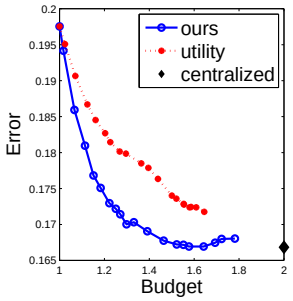


Figure 8: We compare our method to an expected utility approach on the mammogram dataset with the last stage consisting of integer expert rating

References

[1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, Sept. 2001.

[2] P. Bartlett and M. Wegkamp. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.*, 2008.

[3] M. Bilgic and L. Getoor. Voila: Efficient feature-value acquisition for classification. In *AAAI*, 2007.

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

[5] C. Chow. On optimum recognition error and reject tradeoff. *Inf. Theory, IEEE*, 1970.

[6] L. Cordella and C. Sansone. A multi-stage classification system for detecting intrusions in computer networks. *Pattern Anal. Appl.*, 2007.

[7] W. Fan, F. Chu, H. Wang, and P. S. Yu. Pruning and dynamic scheduling of cost-sensitive ensembles. In *AAAI*, 2002.

[8] W. Fan, W. Lee, S. J. Stolfo, and M. Miller. A multiple model cost-sensitive approach for intrusion detection. In *ECML*, 2000.

[9] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu. Support vector machines with a reject option. In *NIPS*, 2008.

[10] S. Ji and L. Carin. Cost-sensitive feature acquisition and classification. In *Pattern Recognition*, 2007.

[11] P. Kanani and P. Melville. Prediction-time active feature-value acquisition for cost-effective customer targeting. In *NIPS*, 2008.

[12] A. Kapoor and E. Horvitz. Breaking boundaries: Active information acquisition across learning and diagnosis. In *NIPS*, 2009.

[13] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *J. Comput. Secur.*, 2002.

[14] L.-P. Liu, Y. Yu, Y. Jiang, and Z.-H. Zhou. Tefe: A time-efficient approach to feature extraction. In *ICDM*, dec. 2008.

[15] D. J. MacKay. Information-based objective functions for active data selection. *Neural Comp.*, 1992a.

[16] E. Rodríguez-Díaz and D. Castañón. Support vector machine classifiers for sequential decision problems. In *IEEE CDC*, 2009.

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

[18] E. Sontag et al. Vc dimension of neural networks. *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, 168:69–96, 1998.

[19] P. Viola and M. Jones. Robust real-time object detection. In *Int. J. of Comp. Vis*, 2001.

[20] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD*, 2003.

[21] C. Yuan and D. Casasent. A novel support vector classifier with better rejection performance. In *CVPR*, 2003.

[22] C. Zhang and Z. Zhang. A survey of recent advances in face detection. In *Microsoft Research Technical Report*, 2010.

[23] V. B. Zubek and T. G. Dietterich. Pruning improves heuristic search for cost-sensitive learning. In *ICML*, 2002.

5 Supplementary Material

5.1 Proofs

Proof of Theorem 1 To simplify our derivations, we assume uniform class prior probability: $\mathrm{P}_y\left[y=\hat{y}\right]=\frac{1}{c},\ \hat{y}=1,\ldots,C$. However, our results can be easily modified to account for a non-uniform prior. The expected conditional risk can be solved optimally by a dynamic program, where a DP recursion is,

$$J_K(\mathbf{x}^K,S^K)=\min_{f^K}\mathbf{E}_y\left[S^K(\mathbf{x}^K)R_k(y,\mathbf{x}^K,f^K)\right] \tag{18}$$

$$J_k(\mathbf{x}^k,S^k)=\min_{f^k}\Big\{\mathbf{E}_y\left[S^k(\mathbf{x}^k)R_k(y,\mathbf{x}^k,f^k)\right]+ \tag{19}$$

$$\mathbf{E}_{\mathbf{x}^{k+1}\ldots\mathbf{x}^K}\left[J_{k+1}(\mathbf{x}^{k+1},S^{k+1})\mid\mathbf{x}^k\right]\Big\} \tag{20}$$

Consider kth stage minimization, f^k can take $C+1$ possible values $\{1,2,\ldots C,\ r\}$ and $J_k(\mathbf{x}^k,S^k)$ can be recast as an conditional expected risk minimization,

$$J_k(\mathbf{x}^k,S^k=1)=$$
$$\min_{f^k}\left\{\underbrace{\mathrm{P}_y\left[y\neq\hat{y}\mid\mathbf{x}^k\right]}_{f^k(\mathbf{x}^k)=\hat{y}},\underbrace{\delta^k+\mathbf{E}_{\mathbf{x}^{k+1}\ldots\mathbf{x}^K}\left[J_{k+1}(\mathbf{x}^{k+1},1)\mid\mathbf{x}^k\right]}_{f^k(\mathbf{x}^k)=r}\right\} \tag{21}$$

Define,

$$\tilde{\delta}(x^k)=\delta^{k+1}+\mathbf{E}_{x^{k+1}\ldots x^K}\left[J_{k+1}(x^{k+1},S^{k+1}=1)\right]$$

and rewrite the conditional risk in 21,

$$f^k=\arg\min_f\left\{\underbrace{1-\mathrm{P}_y\left[y=\hat{y}\mid\mathbf{x}^k\right]}_{f(\mathbf{x}^k)=\hat{y}},\underbrace{\tilde{\delta}^k(\mathbf{x}^k)}_{f(\mathbf{x}^k)=r}\right\} \tag{22}$$

Reject is the optimal decision if,

$$\min_{\hat{y}}\left\{1-\mathrm{P}_y\left[y=\hat{y}\mid\mathbf{x}^k\right]\right\}\geq\qquad\tilde{\delta}^k(\mathbf{x}^k) \tag{23}$$

$$\max_{\hat{y}}\left\{\mathrm{P}_y\left[y=\hat{y}\mid\mathbf{x}^k\right]\right\}\leq\qquad 1-\tilde{\delta}^k(\mathbf{x}^k) \tag{24}$$

If reject is not the optimal strategy then a class is chosen to maximize the posterior probability:

$$f^k(\mathbf{x}^k)=\arg\max_{\hat{y}\in\{1,\ldots,c\}}\left\{\mathrm{P}_y\left[y=\hat{y}\mid\mathbf{x}^k\right]\right\} \tag{25}$$

which is exactly our claim.

Proof of Lemma 2 Define an auxiliary variable corresponding to the error penalty term and absolute value of the maximizing codeword projection respectively:

$$e_i=\mathbb{1}_{\left[d^k(\mathbf{x}_i^k)\neq y_i\right]},\ z_i=\sigma_{d^k}(\mathbf{x}_i^k) \tag{26}$$

$$\tilde{R}_k^i(\cdot)=e_i\mathbb{1}_{[g(x^k)-z_i<0]}+\tilde{\delta}_i^k\mathbb{1}_{[g(x^k)-z_i\geq 0]} \tag{27}$$

$$=e_i\mathbb{1}_{[g(x^k)-z_i<0]}+\tilde{\delta}_i^k\left\{1-\mathbb{1}_{[g(x^k)-z_i<0]}\right\} \tag{28}$$

$$=\left\{e_i-\tilde{\delta}_i^k\right\}\mathbb{1}_{[g(x^k)-z_i<0]}+\tilde{\delta}_i^k \tag{29}$$

Define weights $w_i = e_i - \tilde{\delta}_i^k$ and drop the $\tilde{\delta}_i^k$ term since it does not depend on $g(\cdot)$. Our goal is to minimize $\sum S_i^k \tilde{R}_k^i$ over g. We will split the summation into two sets:

$$= \sum_{w_i \geq 0} S_i^k w_i \mathbb{1}_{\left[\left(g(\mathbf{x}_i^k) - z_i\right) \leq 0\right]} + \sum_{w_i < 0} S_i^k w_i \mathbb{1}_{\left[\left(g(x_i^k) - z_i\right) \leq 0\right]} \tag{30}$$

$$= \sum_{w_i \geq 0} S_i^k w_i \mathbb{1}_{\left[\left(g(x_i^k) - z_i\right) \leq 0\right]} + \sum_{w_i < 0} S_i^k w_i \left\{1 - \mathbb{1}_{\left[\left(g(x_i^k) - z_i\right) > 0\right]}\right\} \tag{31}$$

If discard the constant term $\sum_{w_i<0} S_i^k w_i$ and introduce pseudo labels $b_i = \begin{cases} +1, & w_i \geq 0 \\ -1, & w_i < 0 \end{cases}$ then,

$$\arg\min_g \sum_{i=1}^N S_i^k \tilde{R}_k^i = \arg\min_g \sum_{i=1}^N S_i^k \left|w_i\right| \mathbb{1}_{\left[b_i\left(g(x_i^k) - z_i\right) \leq 0\right]} \tag{32}$$

Proof of Theorem 3 At each stage the reject decision can be expressed in terms of three boolean decisions:

$$\mathbb{1}_{[|h^k(\mathbf{x}^k)| - g^k(\mathbf{x}^k) \leq 0]} = \underbrace{\mathbb{1}_{[h^k(\mathbf{x}^k) > 0]}}_{\text{Decision 1}} \underbrace{\mathbb{1}_{[h^k(\mathbf{x}^k) - g^k(\mathbf{x}^k) \leq 0]}}_{\text{Decision 2}} + \underbrace{\mathbb{1}_{[h^k(\mathbf{x}^k) \leq 0]}}_{\text{Decision 1}} \underbrace{\mathbb{1}_{[-h^k(\mathbf{x}^k) - g^k(\mathbf{x}^k) \leq 0]}}_{\text{Decision 3}} \tag{33}$$

If the rejectors ($g^k \in \mathcal{G}^k$) and stage classifiers ($h^k \in \mathcal{H}^k$) belong to families with finite VC dimensions then the complexity of Decision 2 and Decision 3 is $\mathcal{VC}[\mathcal{G}^k] + \mathcal{VC}[\mathcal{H}^k]$

The system classifier, F, is composed of K stages. Each of the first $K-1$ stages can be expressed as a boolean function of 3 boolean decisions. The last stage is a single boolean decision. So the output F can be expressed as a boolean function of $3(K-1)+1 = 3K-2$ functions. We know the VC dimension for each of the functions. Using this fact and Lemma 2 in [18] we obtain our result.

5.2 Implementation Details

For large datasets ($N > 1000$), we split them 50/10/40% into train, validation and test sets. The performance reported is on the test set. For smaller datasets ($N < 1000$), we perform 50 random 70/10/20% splits and report the average performance over the trials. Each subproblem reduces to minimizing a weighted binary error problem with respect to a logistic loss. Polynomial kernel classifier of degree q is parametrized by a vector $\mathbf{a}$:

$$h(x) = \sum_{i=1}^N a_i (\mathbf{x}_i^T \mathbf{x} + 1)^q$$

The optimization over the polynomial kernel classifier is performed using newton gradient descent method. Table 1 shows the degree of polynomial kernels used in our simulations.

Dataset	$\mathcal{H}^1$	$\mathcal{G}^1$	$\mathcal{H}^2$	$\mathcal{G}^2$	$\mathcal{H}^3$	$\mathcal{G}^3$	$\mathcal{H}^4$
synthetic	2	2	2				
mam	2	0	2				
pima	2	0	2	0	2		
threat	5	5	5	5	5		
covertype	1	1	1	1	1		
letter	7	2	7	2	7		
mnist	1	1	1	1	1	1	1
landsat	3	2	3	2	3	2	3

Table 3: Stage Complexity: we use polynomial kernel classifiers. This table displays the degree of the polynomial kernel used at each stage for the rejector and the stage classifier