Cost-Sensitive Feature Acquisition and Classification

Shihao Ji, Lawrence Carin

Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291, USA

Abstract

There are many sensing challenges for which one must balance the effectiveness of a given measurement with the associated sensing cost. For example, when performing a diagnosis a doctor must balance the cost and benefit of a given test (measurement), and the decision to stop sensing (stop performing tests) must account for the risk to the patient and doctor (malpractice) for a given diagnosis based on observed data. This motivates a cost-sensitive classification problem in which the features (sensing results) are not given *a priori*; the algorithm determines which features to acquire next, as well as when to stop sensing and make a classification decision based on previous observations (accounting for the costs of various types of errors, as well as the rewards of being correct). We formally define the cost-sensitive classification problem and solve it via a partially observable Markov decision process (POMDP). While the POMDP constitutes an intuitively appealing formulation, the intrinsic properties of classification tasks resist application of it to this problem. We circumvent the difficulties of the POMDP via a myopic approach, with an adaptive stopping criterion linked to the standard POMDP. The myopic algorithm is computationally feasible, easily handles continuous features, and seamlessly avoids repeated actions. Experiments with several benchmark datasets show that the proposed method yields state-of-the-art performance, and importantly our method uses only a small fraction of the features that are generally used in competitive approaches.

Key words: Cost-sensitive classification, Partially observable Markov decision processes (POMDP), Hidden Markov models (HMMs), Variational Bayes (VB)

1 Introduction

A classifier is a function that maps a feature vector into a class label. Many traditional classifiers are “passive”, in that they assume a feature vector is given, ignoring the potential advantage of endowing the classifier with an active information-gathering function to acquire features. Furthermore, a traditional classifier is considered good if it assigns the class label correctly for as many examples as possible. This performance measure is too simplistic for many classification domains, in which one must balance the accuracy of a given classifier against the cost of acquiring the data. Medical diagnosis [1,2] may be the most ubiquitous scenario motivating the ideas considered here. In medical diagnosis, a doctor typically takes a few measurements (tests) on a given patient to determine the patient's health status. The measurements are usually taken sequentially, with low-cost measurements performed initially, followed if necessary by more costly and specialized measurements. At each step the doctor analyses the results obtained from previous tests and determines whether a further test (and which type of test) is required to gather more information; the doctor may also decide to terminate the sensing (tests) and make a final classification decision (diagnosis). Each test has an associated cost, and there are also costs/rewards associated with different diagnoses. The doctor would like to balance the cost of gathering additional information against the costs of a false diagnosis (and the rewards, or negative costs, of a correct diagnosis). An algorithm that can guide the active acquisition of information and balance the costs is often termed a cost-sensitive classifier [1].

Both generative and discriminative models [3] may be used to learn the mapping of a feature vector to a class label. Generative models estimate the joint probability of feature vectors and the class label. By incorporating prior domain knowledge, Bayes rule is applied to infer the posterior distribution of class labels after observing data. In contrast, discriminative classifiers estimate the conditional probability of the class label directly, partitioning the feature space into regions corresponding to different classes. Although discriminative models are often more robust than generative models [3], generative classifiers may be more suitable for cost-sensitive classification. Example advantages of generative classifiers are: (i) a direct means of incorporating prior domain knowledge, (ii) a direct integration into cost-reward algorithms such as POMDPs [4], and (iii) an iterative update of the posterior distribution as more features are added sequentially. These factors drive our use of generative models and classifiers in the analysis that follows.

Bayesian networks and their simplest form - naive Bayes - have been investigated in previous research on cost-sensitive classification [5, 6]. However, this previous work ignores the temporal information as the features (observations) are acquired in a sequential manner. We here use a hidden Markov model (HMM), allowing explicit consideration of the sequential order in which the observations are made. In the analysis that follows we motivate the meaning of the underlying HMM states, these distinct from the states associated with previous HMM applications, such as speech recognition [7] and bioinformatics [8].

The class-dependent HMM formulation can be integrated into the partially observable Markov decision process (POMDP). In [6], a related POMPD formulation has been considered for cost-sensitive classification. However, in that case each class is represented by a single POMDP state, under a naive Bayes assumption. Our formulation relaxes this strong assumption and generalizes to a more formal POMDP setting, in which the feature dependency is explicitly encoded by the hidden states of the HMM. We note that for a one-state HMM, our formulation degenerates to naive Bayes as in [6].

While the POMDP constitutes an intuitively appealing formulation, the intrinsic properties of classification tasks resist application of a POMDP to this problem. At least three factors restrict its use : (i) significant computational requirements, (ii) the difficulty of handling continuous features (observations), and (iii) the repeated actions that are permissible in a standard POMDP are often undesirable in classification problems (i.e., we often do not wish to repeat medical tests). We circumvent the difficulties of the POMDP via a myopic approach, with an adaptive stopping criterion linked to the standard POMDP. The myopic algorithm is computationally feasible, easily handles continuous features, and seamlessly avoids repeated actions. Experiments with several benchmark datasets show that the proposed method yields state-of-the-art performance, and importantly our method uses only a small fraction of the features that are generally used in competitive approaches.

The remainder of the paper is organized as follows. In Sec. 2 the formal definition of the cost-sensitive classification problem is presented, and we explain why an HMM formulation is appropriate, with HMM parameter learning and variational Bayes model selection techniques discussed in the Appendix. In Sec. 3 we map the cost-sensitive classification problem into a POMDP framework. Following the discussion of the advantages and limitations of this POMDP setting, an alterative myopic solution is addressed in Sec. 4. We present experimental results on several benchmark datasets in Sec. 5, followed in Sec. 6 by conclusions and a discussion of future work.

2 Bayesian Cost-Sensitive Classification

2.1 Probabilistic modeling of feature acquisition process

Assume a d-dimensional feature vector is represented as $(x_1, x_2, \cdots, x_d)$, where x_i represents the value of the ith feature. We wish to consider the problem for which features are measured sequentially, and depending on the attendant risk and sensing costs a final classification decision may be made without collecting all features. It is here assumed that each measurement is characterized by a single real number (feature); we may readily extend this to address problems for which a measurement is characterized by a feature vector.

Let $A_f = \{\rho_1, \rho_1, \cdots, \rho_d\}$ denote a set of feature-acquisition actions, with ρ_i representing the action of measuring the ith feature. We may then use the pair $(a_t = \rho_i, o_t = x_i)$ to represent that at time t the corresponding action a_t queries the ith feature, and the subsequent observation o_t has value x_i. Therefore, acquisition of T features may be expressed as $(a_1, o_1), (a_2, o_2), \cdots, (a_T, o_T)$, and $T = d$ if all features are collected (but often $T < d$). Taking into account the order in which the features are acquired, there are $d!/(d-T)!$ different ways $T \leq d$ measurements may be performed.

To motivate the HMM modeling of a feature-acquisition process, we assume that each feature is modeled as a Gaussian mixture model (GMM), with the marginal probability of feature x_i represented as

$$p(x_i) = \sum_{m_i=1}^{M_i} \alpha_{i,m_i} \mathcal{N}(x_i | \mu_{i,m_i}, \Gamma_{i,m_i}) \tag{1}$$

where M_i denotes the number of mixture components for the ith feature, α_{i,m_i} is the mixing coefficient for the m_i-th component, with $0 \leq \alpha_{i,m_i} \leq 1$ and $\sum_{m_i=1}^{M_i} \alpha_{i,m_i} = 1$; and μ_{i,m_i} and Γ_{i,m_i} respectively represent the associated mean and precision (inverse variance).

The process of performing a sequence of measurements and observations (a_1, o_1), (a_2, o_2), $\cdots$, (a_T, o_T) may be described as follows. Assume that the first action a_1 corresponds to measuring feature $i \in \{1, 2, \cdots, d\}$. The associated observation o_1 is realized by first selecting a mixture component $m_i \in \{1, 2, \cdots, M_i\}$, based on the probabilities $\{\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,M_i}\}$. Once mixture component m_i is so selected, the associated feature x_i is realized with probability $\mathcal{N}(x_i | \mu_{i,m_i}, \Gamma_{i,m_i})$. Now assume action a_2 corresponds to measuring feature $j \in \{1, 2, \cdots, d\}$, with $j \neq i$. The joint probability is then represented as

$$p(x_i, x_j) = p(x_i)p(x_j|x_i) = p(x_j|x_i) \sum_{m_i=1}^{M_i} \alpha_{i,m_i} \mathcal{N}(x_i | \mu_{i,m_i}, \Gamma_{i,m_i}) \tag{2}$$

We again employ a GMM for feature j

$$p(x_j) = \sum_{m_j=1}^{M_j} \alpha_{j,m_j} \mathcal{N}(x_j | \mu_{j,m_j}, \Gamma_{j,m_j}) \tag{3}$$

and use $p(m_j|m_i)$ to represent the probability of observing mixture component m_j for feature j when component m_i was used for component i. If the sampled mixture components satisfy a Markovian assumption, and feature x_j is conditionally independent given the associated mixture component, then we have

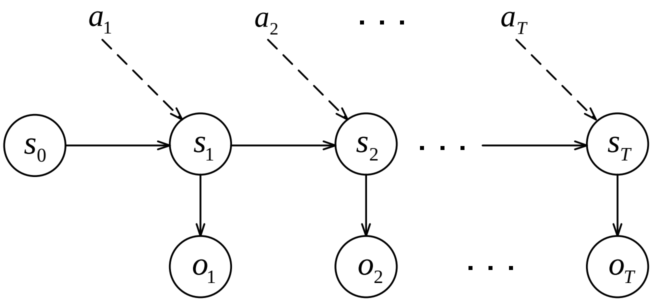


Fig. 1. A graphical representation of IOHMM modeling of a feature acquisition process for a given class, where s_t, a_t and o_t denote the state, action and observation at the time step t, respectively.

$$\begin{aligned} p(x_i, x_j) &= p(x_i)p(x_j|x_i) \\ &= \sum_{m_j=1}^{M_j} \sum_{m_i=1}^{M_i} \alpha_{i,m_i} \mathcal{N}(x_i|\mu_{i,m_i}, \Gamma_{i,m_i}) p(m_j|m_i) \mathcal{N}(x_j|\mu_{j,m_j}, \Gamma_{j,m_j}) \end{aligned} \quad (4)$$

This process may be continued for the $T \leq d$ sequential measurements that may be performed, motivating an HMM in which the underlying states are tied to mixture components associated with particular features, and the state-transition probabilities are tied to the probability of sampling mixture component m_j for feature j when feature component m_i was sampled for feature i, i.e., $p(m_j|m_i)$. The underlying sequence of sampled states (mixture components) is modeled via a Markov process, but these are unobservable $-$ we only observe the measured features, this motivating the *hidden* Markov model. We also note that any mixture model may have been employed above, not just a GMM.

We formally define the HMM modeling motivated above with an n-state[1] HMM with the parameters $\theta = \{\pi, \tau, \phi\}$, where π_s is the initial state distribution, $\tau^a_{s,s'} := p(s_t = s'|s_{t-1} = s, a_t = a)$ is the state-transition function representing the probability of transitioning from state s to state s' given action a, and $\phi_{s,o} := p(o_t = o|s_t = s)$ is the observation function describing the probability of observing o from state s. We note that this HMM modeling of a feature acquisition process is a generalization of the Input-Output Hidden Markov Model (IOHMM) [9], an extension of the standard HMM by allowing for transition probabilities between states conditioned by the actions. In our model, actions drive transition probabilities between states and we model observations for every state in the IOHMM (see Fig. 1).

With the HMM parameters θ defined above, the probability of observing

[1] n is often less than the total number of mixture components over all the features. This indicates that some states are shared among different features if these features have some overlap in feature space.

$(o_1, o_2, \cdots, o_T)$ for a given action sequence $(a_1, a_2, \cdots, a_T)$ is expressed as

$$p(o_1, o_2, \cdots, o_T | a_1, a_2, \cdots, a_T, \theta) = \sum_{s_0, \cdots, s_T} \pi_{s_0} \prod_{t=1}^{T} \tau^{a_t}_{s_{t-1}, s_t} \prod_{t=1}^{T} \phi_{s_t, o_t} \tag{5}$$

Although the underlying state sequence forms a first-order Markovian process, the observations are not Markovian; the dependencies among the observations are explicitly encoded by the hidden states. If we use a one-state HMM, (5) degenerates to a naive Bayes model, with the result

$$p(o_1, o_2, \cdots, o_T | a_1, a_2, \cdots, a_T, \theta) = \prod_{t=1}^{T} \phi_{o_t} \tag{6}$$

where ϕ_o represents the probability of observing o when querying features from the class parameterized by θ. In this case, the time index in (6) becomes meaningless and all the features are assumed to be independent of each other given the class label.

The HMM parameter θ employed here is trained using a variational Bayes (VB) procedure [10, 11], as discussed in the Appendix. In addition, the VB procedure allows model selection, where here this entails choosing the proper number of HMM states for a given class. Model selection based on the VB analysis is also discussed in the Appendix.

2.2 Bayesian HMM classifier

Now consider a classification problem in which a feature vector is mapped into a class label from a finite set $\mathcal{C} = \{1, 2, \cdots, |\mathcal{C}|\}$. Each class will, in general, be modeled as an HMM with a distinct number of states $n^{(c)}, \forall c \in \mathcal{C}$. We therefore employ the notation $\tau^{a}_{s_i^{(c)}, s_j^{(c)}} := p(s_j^{(c)} | s_i^{(c)}, a)$, $\phi_{s_k^{(c)}, o} := p(o | s_k^{(c)})$ and $\pi_{s_k^{(c)}}$ to represent the HMM parameters θ_c for class c; for example, $s_k^{(c)}$ represents the kth state of class c. By incorporating the prior distribution of class label $p(c)$, Bayes rule is applied to compute the posterior distribution of class label

$$p(c | o_1, \cdots, o_T, a_1, \cdots, a_T, \Theta) = \frac{p(o_1, \cdots, o_T | a_1, \cdots, a_T, \theta_c) p(c)}{\sum_{c \in \mathcal{C}} p(o_1, \cdots, o_T | a_1, \cdots, a_T, \theta_c) p(c)} \tag{7}$$

with $\Theta = \{p(c), \theta_c\}_{c \in \mathcal{C}}$ denoting the cumulative parameters of the Bayesian HMM classifier. The class that is most probable to have generated such a sequence is used as the estimated class label.

In contrast with traditional classifiers, which passively receive feature vectors (observation sequences) and do their best to predict the associated class labels, the Bayesian HMM classifier (7) can be used to actively select action sequence $(a_1, a_2, \cdots, a_T)$ to maximize its ability to discriminate among classes. Further, as discussed below, the probabilistic output of classification results also provides a mechanism to incorporate costs, adaptively determining what next sensing action to take, as well as when to stop sensing and make a classification decision.

2.3 Definition of cost-sensitive classification problem

In addition to the feature-acquisition actions A_f discussed above, we introduce a second type of action, defined by $A_c = \{\hat{\rho}_1, \hat{\rho}_2, \cdots, \hat{\rho}_{|\mathcal{C}|}\}$, with $\hat{\rho}_c$ representing the action that terminates feature acquisition and declares the object under interrogation to be class c. To evaluate the outcomes of a classification action, we define C_c a $|\mathcal{C}| \times |\mathcal{C}|$ matrix; matrix element $c_{uv}, \forall u, v \in \mathcal{C}$, represents the cost of declaring the item under interrogation to be class u, when in reality it is class v. We also define the set of costs for feature-acquisition actions as $C_f = \{c_1, c_2, \cdots, c_d\}$, with c_i representing the cost of sensing the ith feature, where here the costs could be monetary for instrument usage or could be in terms of time for laboratory analysis, but they should have the same units as used for C_c. For more information on defining the costs associated with a cost-sensitive classifier, one may refer to [12]. With the definitions outlined above, the cost-sensitive classification problem can now be described as follows:

Definition: Given a finite set of classes $\mathcal{C} = \{1, 2, \cdots, |\mathcal{C}|\}$, a cost-sensitive classifier finds an optimal action sequence $(a_1^*, a_2^*, \cdots)$, where $a_i^* \in A_f \cup A_c$, to identify the class label, while on average minimizing the cumulative costs defined as the sum of the feature acquisition and classification costs, C_f and C_c, respectively.

3 POMDP Formulation to Cost-Sensitive Classification

The cost-sensitive classification problem, based on an HMM classifier, can be directly formulated into a partially observable Markov decision process (POMDP). We proceed to this by first giving a brief introduction to the POMDP framework. The POMDP is a general setting for planning under uncertainty [4,13,14]. A PODMP can be represented by the following 7-tuple: $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, b_0, T, \Omega, C\}$, where $\mathcal{S}$ is a finite set of discrete states, satisfying a first-order Markov assumption, $\mathcal{A}$ is a set of discrete actions, and $\mathcal{O}$ is a set of observations providing incomplete or noisy state information. The POMDP model is parameterized by: $b_0(s)$, the initial belief state; $T^a_{s,s'} := p(s_t = s' | s_{t-1} = s, a_t = a)$, the state transition function describing the probability of transitioning from state s to state s' when taking action a; $\Omega^a_{s,o} := p(o_t = o|s_t = s, a_t = a)$, the observation function describing the probability of observing o from state s after taking action a; $C(s, a)$, the cost of executing action a in state s.

The mapping of the HMM to a POMDP is direct: $\mathcal{S} = \{s^{(c)}_k, \forall k, c\}$ is the set of states across all classes; $\mathcal{A} = A_f \cup A_c$ is the union of feature-acquisition actions A_f and classification actions A_c; and $\mathcal{O}$ is the union of all possible observations across all the features and all the classes. The POMDP parameters can be integrated from the Bayesian HMM classifier Θ, and they are specified as follows:

(1) **Initial belief**
$b_0(s)$ is the initial probability of being in state $s, \forall s \in \mathcal{S}$. The prior distribution of class label $p(c)$ can be incorporated into the POMDP by setting

$$b_0(s) = p(c)\pi_{s^{(c)}_k}, \text{ if } s = s^{(c)}_k \tag{8}$$

(2) **State-transition function**
A special structure is imposed on the state-transition probability, i.e., the state transition is restricted to the internal states within one class,

$$T^a_{s^{(u)}_i, s^{(v)}_j} = \begin{cases} \tau^a_{s^{(u)}_i, s^{(v)}_j}, & \text{if } u = v \\ 0, & \text{else} \end{cases}, \text{for } a \in A_f \tag{9}$$

This reflects the fact that for a given (unknown) class, the dynamic system must be in a subset of associated states and the feature-acquisition action cannot change the underlying class.

(3) **Observation function**
$\Omega^a_{s,o}$ is the probability of observing o from state s conditioned on action a. For the feature acquisition action, this corresponds to the observation function ϕ in HMM

$$\Omega^a_{s,o} = \phi_{s^{(c)}_k, o}, \text{if } s = s^{(c)}_k, \forall o \in \mathcal{O}, a \in A_f \tag{10}$$

Note that, in our mapping, the observation function $\Omega^a_{s,o}$ is only linked to the state but is the same for all $a \in A_f$; its action-dependency is fulfilled via the state-transition probabilities (see the motivation in Sec. 2.1).

(4) **Cost function**
For feature-acquisition actions A_f, the costs are assumed independent of which particular state is being interrogated,

$$C(s, a = \rho_i) = c_i, \forall s \in \mathcal{S}, a \in A_f \tag{11}$$

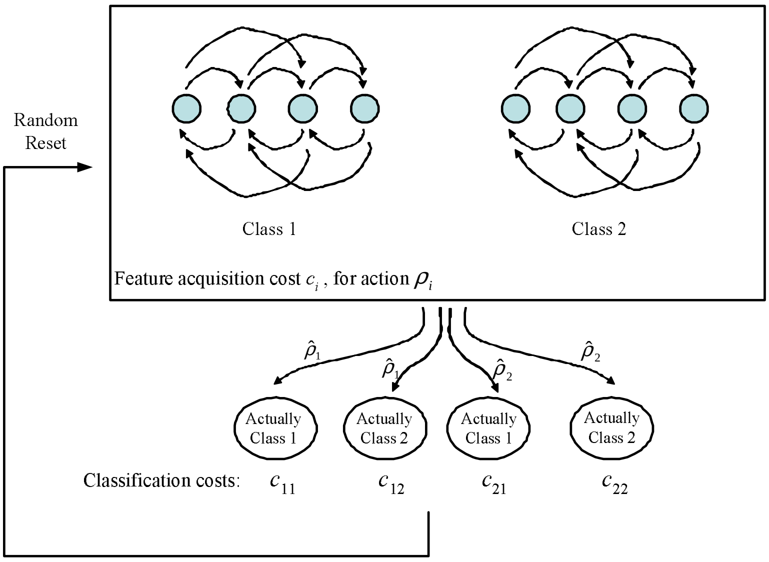


Fig. 2. Schematic of the POMDP formulation for simple case of binary classification problem. The action ρ_i (top box) corresponds to query the ith feature of a given item; the action $\hat{\rho}_c$ corresponds to stopping feature query and declaring class c. After the classification action, the algorithm randomly resets the next state of another item based on b_0.

Similarly, for the classification actions A_c, since the ultimate objective is classification, different classification actions will incur different costs depending on the true class under interrogation,

$$C(s, a = \hat{\rho}_u) = c_{uv}, \forall s \in S_v, a \in A_c \tag{12}$$

where $S_v = \{s_k^{(v)}, k\}$ denotes the set of states associated with class v. It shows that once we declare the object under interrogation to be in class u, if in reality it is in class v, a cost of c_{uv} is incurred independent of which specific state of class v is truly observed.

After taking a classification action, the POMDP model resets and it is assumed to transition to a state of a randomly selected item based on the initial belief distribution b_0, i.e., $p(s'|s,a) = b_0(s'), \forall a \in A_c$, and a new classification episode starts. In summary, we illustrated the POMDP formulation of the cost-sensitive classification problem in Fig. 2, for the simple case of two classes.

Upon constructing the POMDP model as outlined above, POMDP policy design can be accomplished by using many existing algorithms [4,13,14], among which PBVI [13] represents a practical algorithm that can find an approximate policy in polynomial time (rather than exponential time required by the exact solution). This yields a non-myopic policy that maps a belief state to an action, with the goal of minimizing the expected sum of costs [4]. Although policy design is a time-consuming process, the computation is performed offline. After learning the policy, policy-based action selection for feature acquisition and final classification is essentially instantaneous.

4 Myopic Solution to Cost-Sensitive Classification

4.1 Advantages and limitations of POMDP formulation

While the POMDP constitutes an intuitively appealing formulation, it presents several challenges for the problem of interest here. For example, [15] has shown that finding the exact solution for a POMDP is PSPACE-complete, which is intractable for most practical problems of interest. Thus, approximate techniques (e.g., [13,14]) have been applied in many real problems to find a policy with the hope that it is close to optimal. Further, existing POMDP algorithms only deal with discrete observations; there is no feasible (even approximate) algorithm at this moment that can treat continuous observations well. This means to apply the POMDP, features must be quantized before any processing. This is often undesirable since information may be lost by quantization, deteriorating classification performance. Perhaps the most undesirable property of the POMDP formulation, for the applications of interest here, is the issue of repeated actions (querying of the same feature multiple times). Repeated actions may be feasible under some circumstances for which an item has features that obey an i.i.d probability distribution and the observation may change independently on each sampling. However, in many cases, for a given item under test, at minimum the sequence of identical sensing actions will yield highly correlated observations, and often the observations will be identical. It is therefore undesirable in many cases to allow multiple queries (no information is gained from these queries, and a sensing cost is incurred).

The traditional POMDP formulation does not have a mechanism to forbid action repetition. This is because in a traditional POMDP, the belief state b_T is the sufficient statistic of the history $(a_1, \cdots, a_T, o_1, \cdots, o_T, b_0)$, and the optimal POMDP policy maps a belief state to an action [4]. However, for the problem considered here, for which the repeated action is undesirable, the belief state is not a sufficient statistic since all previous actions $(a_1, a_2, \cdots, a_T)$ must be remembered to avoid repetition. The optimal policy, in this case, must be a function of the belief state and all possible combinations of actions, and this increases the complexity of finding the optimal policy substantially (a non-stationary policy is required, accounting for what previous actions were taken). Our initial analysis of this case suggests that this problem is only computationally tractable for small d (small number of features or sensing actions), but this is a subject for future research.

4.2 Proposed myopic approach

We circumvent the difficulties of the POMDP formulation by considering a myopic approximation to the non-myopic POMDP solution, with an adaptive stopping criterion implicitly linked to the POMDP setting, i.e., when the expected future reduction in Bayes risk is not justified by the expected future cost in feature acquisition, a cost-sensitive classifier should stop querying more features and a final classification is made. Yielding the non-myopic property of the POMDP rewards the proposed method with significant flexibility to avoid repeated actions, while also allowing consideration of continuous features.

After performing a sequence of T actions and making T observations, we may compute the belief state for any state $s \in \mathcal{S}$ as

$$b_T(s) = p(s|o_1, \cdots, o_T, a_1, \cdots, a_T, b_0) = p(s|o_T, a_T, b_{T-1}) \tag{13}$$

Equation (13) reflects that the belief state b_T is a sufficient statistic for a given history: (a_1, $\cdots$, a_T, o_1, $\cdots$, o_T, b_0), and b_T can be updated from b_{T-1} by incorporating the latest action-observation pair [4]:

$$\begin{aligned} b_T(s') &= \frac{p(o_T|s', a_T, b_{T-1})p(s'|a_T, b_{T-1})}{p(o_T|a_T, b_{T-1})} \\ &= \frac{p(o_T|s', a_T, b_{T-1}) \sum_s p(s'|a_T, b_{T-1}, s)p(s|a_T, b_{T-1})}{p(o_T|a_T, b_{T-1})} \\ &= \frac{p(o_T|s', a_T) \sum_s p(s'|s, a_T)b_{T-1}(s)}{p(o_T|a_T, b_{T-1})} \end{aligned} \tag{14}$$

where the denominator $p(o_T|a, b_{T-1})$ may be viewed as a normalization constant, independent of s', allowing $b_T(s')$ to sum to one.

For the POMDP formulation addressed in Sec. 3, in which each class is modeled by a distinct HMM, the belief state in (14) may also be used to compute the posterior distribution of class label as

$$p(c|o_1, \cdots, o_T, a_1, \cdots, a_T, b_0) = p(c|b_T) = \sum_{s \in S_c} b_T(s) \tag{15}$$

One may show that (15) is equivalent to (7), indicating that an HMM classifier is essentially embedded in the belief states, and it can be efficiently computed by using the belief update (14). By incorporating the classification costs C_c, the Bayes risk associated with b_T can be computed as

$$R(b_T) = \min_u \sum_{v \in \mathcal{C}} c_{uv} p(v|b_T) = \min_u \sum_{v \in \mathcal{C}} c_{uv} \sum_{s \in S_v} b_T(s) \tag{16}$$

and a classification is effected by declaring the class that has the minimal Bayes risk.

The Bayes risk (16) tells us how to use the belief state to make an optimal decision at any point in the feature-acquisition process. The question on when to make a classification is answered by comparing the cost of taking a further feature-acquisition action against the expected reduction in Bayes risk, as addressed below.

Assume that a feature-acquisition action $a \in \bar{A}_f$ could be taken next, where $\bar{A}_f$ is composed of all actions A_f except the actions that were taken previously, $(a_1, a_2, \cdots, a_T)$. The expected Bayes risk associated with a may be computed as

$$R_E(b_T, a) = \sum_{o \in \mathcal{O}} \min_u \left[\sum_{v \in \mathcal{C}} c_{uv} \sum_{s' \in S_v} \sum_{s \in \mathcal{S}} p(o|s', a) p(s'|s, a) b_T(s) \right] \qquad (17)$$

where the summation is over all possible observation from a finite set $\mathcal{O}$ for a discrete POMDP. The utility of action a is then evaluated by

$$\hat{C}(b_T, a) = c_a - [R(b_T) - R_E(b_T, a)] \qquad (18)$$

where c_a represents the cost of taking action a. If $\hat{C}$ for all $a \in \bar{A}_f$ is positive, it indicates that the cost of feature acquisition exceeds the expected reduction in Bayes risk, thus acquiring more features is not justified and a classification is made based on b_T. Otherwise, if $\hat{C}$ for some $a \in \bar{A}_f$ is negative, it shows that it is still beneficial to acquire features and the action that has the minimal $\hat{C}$ should be taken next until a classification action is justified. In summary, we present the flowchart of the myopic algorithm in Table 1.

This myopic strategy is performed exactly and there is no offline policy design, as required in the non-myopic POMDP. To better understand the complexity of the myopic algorithm, let $|B|$ be the number of elements in the set B. The computation complexity of choosing one action is order $|\mathcal{S}|^2|\mathcal{O}||\bar{A}_f|$. This is much more efficient than that of the non-myopic POMDP design, which is order $|\mathcal{S}|^2|\mathcal{A}||V|^{|\mathcal{O}|}$, with V representing the set of α-vectors in the previous *backup* step [4, 13].

4.3 Continuous features

The above myopic algorithm considered a discrete set of observations, to make the connection with a traditional POMDP. However, the simplicity of the myopic algorithm enables us to readily handle continuous features. In this

Table 1
Flowchart of the myopic algorithm for the Bayesian cost-sensitive classification.

```
1. **Given**: Bayesian HMM classifier parameters $\Theta$, costs for feature acquisition $C_f$, costs for classification $C_c$, and a randomly selected instance with a $d$-dimensional feature vector $F$.
2. $b_0(s) = p(c)\pi_{s_k(c)}$, if $s = s_k^{(c)}$
3. $T = d$
4. **for** $t = 1 : T$
      $a_t = \arg\min_{a \in A_f - \{a_1, \cdots, a_{t-1}\}} \hat{C}(b_{t-1}, a)$
      **if** $\hat{C}_{min} > 0$
         $\hat{a} = \arg\min_{u \in A_c} [\sum_{v \in \mathcal{C}} c_{uv} \sum_{s \in S_v} b_{t-1}(s)]$ and return $\hat{a}$
      **end**
      $o_t = F(a_t)$
      $b_t$ =belief_update$(b_{t-1}, a_t, o_t)$ eq.(14)
   **end**
5. $\hat{a} = \arg\min_{u \in A_c} [\sum_{v \in \mathcal{C}} c_{uv} \sum_{s \in S_v} b_T(s)]$ and return $\hat{a}$
```

case, the summation in (17) becomes an integration

$$R_E(b_T, a) = \int p(o|b_T, a) R(b_{T+1}|b_T, a, o) do \tag{19}$$

As discussed in Sec. 2.1, the observation probability of each state is represented by a Gaussian distribution. Since each feature is a real number, we may use a 1-D Gaussian distribution to represent one state in the POMDP model, thus $p(o|b_T, a_{T+1})$ can be computed analytically with a 1-D mixture of Gaussian distribution. However, the minimization operator in computing $R(b_{T+1}|b_T, a, o)$ hinders analytic computation of (19). Therefore, we employ a sampling technique to evaluate it approximately. For example, if we use K samples $\{\tilde{o}_1, \tilde{o}_2, \cdots, \tilde{o}_K\}$ generated from density $p(o|b_T, a)$, then the expected Bayes risk may be estimated as

$$R_E(b_T, a) \approx \frac{1}{K} \sum_{i=1}^{K} R(b_{T+1}|b_T, a, \tilde{o}_i) \tag{20}$$

However, we do not compute this term as in (20), since a large number of samples are required to obtain accurate estimation. Alternatively, we convert the continuous observation probability (1-D Gaussian distribution) into a discrete one (multinomial distribution), which entails the approximate evaluation of (19) as in (17). To do this, we quantize the feature space into a finite set of bins with the *sorted* centers of bins represented by $\{\bar{o}_1, \bar{o}_2, \cdots, \bar{o}_K\}$. Then the probability of observing $\bar{o}_i$ in $s \in \mathcal{S}$ can be approximated as

$$\Omega_{s,\bar{o}_i} = \int_{(\bar{o}_{i-1}+\bar{o}_i)/2}^{(\bar{o}_i+\bar{o}_{i+1})/2} \mathcal{N}(x|\mu_s, \Gamma_s)dx, \quad \text{with } \bar{o}_0 = -\infty, \bar{o}_{K+1} = \infty \tag{21}$$

where μ_s and Γ_s are the mean and precision of the Gaussian distribution, respectively, representing state s.

The effectiveness of this approximation is addressed as follows. As demonstrated in the experiments, prior to applying the proposed method, all the features are normalized to zero mean and unit variance; thus, the range of the observations is fairly small, and we use k-means [16] to determine the centers of bins, which results in a quantized representation that is close to the original continuous feature space. In addition, (19) is only used for choosing the next action, and the advantage of handling continuous features remains in the more accurate belief update (14).

We emphasize that the discretization of (21) is very different than the discrete-observation POMDP, in that the number of dicretization bins can be arbitrarily large, since the computation complexity of the myopic algorithm has linear growth in number of discrete observations, while the discrete POMDP has exponential growth (see Sec. 4.2).

5 Experimental Results

We assess the performance of the cost-sensitive classifier on three well-known benchmark datasets: the *Pima Indians diabetes*, the *ionosphere*, and the *Wisconsin diagnostic breast cancer* (WDBC), which are accessible from the UCI public website: `http://www.ics.uci.edu/~mlearn/MLRepository.html`. For the *Pima Indians diabetes*, the goal is to decide whether a subject has diabetes or not, based on 8 measured variables; for the *ionosphere*, the problem is to classify radar returns from the ionosphere, based on 34 features; and for the WDBC, the task is to produce a benign/malignant diagnosis from a set of 30 numerical features. Although all of these problems are binary classification tasks, the POMDP and the myopic algorithm do not have this restriction and can be applied to any multi-class problem. For POMDP policy design, we use the PBVI algorithm [13] to find an approximate policy. We consider two approaches when employing the POMDP policy for feature selection and classification: (1) the POMDP-repeat, in which the repeated actions are allowed and we just follow the policy until a classification action is selected, (2) the POMDP-norepeat, in which the repeated actions are forbidden and at each step we select the best action among all the remaining actions (in terms of the details of the POMDP [13], we always choose the α-vector that is optimal from among those that have not been used yet). Although the

POMDP-norepeat can avoid repeated actions, the non-myopic policy, learned in a standard POMDP setting, does not have any mechanism to avoid repeated actions. Thus, in a rigorous sense, the action sequences produced by the POMDP-norepeat are not optimal. We present this set of results only to evaluate the importance of avoiding repeated actions, and to compare with other simple modifications to the POMDP that one could consider to avoid repeated actions.

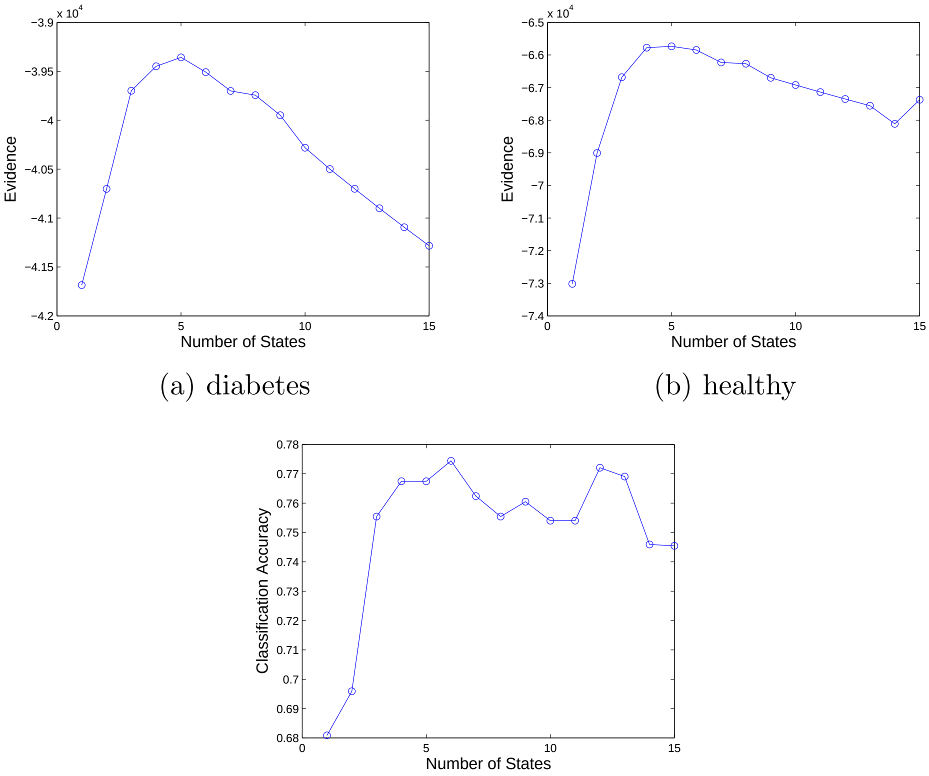
In all the experiments, prior to applying the algorithms, all features are normalized to zero mean and unit variance, as in [17, 18]. For the POMDP solution, which can only deal with discrete observations, we further quantize the normalized features into 40-dimensional discrete alphabet using k-means [16]. When performing model selection (see Appendix B), we use the VB-HMM algorithm in [10] for discrete observations, i.e., for POMDP, and use the VB-CHMM algorithm in [11] for continuous observations. Since we have applied the VB method to select the model, the parameters of the cost-sensitive classifier can be obtained directly from the mean values of the posterior density of HMM parameters learned from the VB method. For the myopic algorithm that deals with continuous features, we use (21) to convert the continuous observation function into a discrete one with the $K = 40$ centers of bins determined by k-means [16].

5.1 Experiments on the Pima Indian diabetes dataset

The Pima dataset includes 768 instances with each instance having 8 features, representing 8 distinct medical tests, and there is a label of either "diabetes" or "healthy" for each. In our experiments, these 768 cases are randomly split into a training set of 512 examples and a testing set of the remaining 256. From each training case, we randomly generate $p = 20$ feature acquisition processes (by permuting the order in which the features were acquired; see Appendix A) and obtain totally 512 × 20 features acquisition processes to train a cost-sensitive classifier.

We first use the VB model selection technique discussed in Appendix B to determine the optimal number of states for each class, and then we construct the cost-sensitive classifier by using the mean values of the posterior density corresponding to the optimal number of states. An example of results on model selection by using VB-CHMM [11] is shown in Fig. 3.

In Figs. 3(a) and 3(b), we compute the evidence for every distinct number of states (from 1 to 15) for the class "diabetes" and the class "healthy", respectively, and the optimal number of states is identified as five for each class. For the number of states beyond these optimal values, the evidence decreases



(a) diabetes

(b) healthy

(c) classification accuracy

Fig. 3. Experimental results of model selection on the Pima data with randomly selected 512 training instances and 256 testing instances. (a) model selection for class "diabetes"; (b) model selection for class "healthy"; (c) classification accuracy of the myopic solution on the testing instances with the cost defined as $c_i = 0$, $c_{uu} = 0$, and $c_{uv} = 1$, and the number of states is the same for both label types.

rapidly. This is because of the appearance of one or more redundant states that collapses onto a specific data point, and the existence of redundant states does not increase the average likelihood but leads to an increasing penalty manifested in the second term of (B.6). To evaluate the effectiveness of the model-selection technique, in Fig. 3(c) we also present the classification accuracy on the test instances for every distinct number of states, by using the myopic algorithm with the defined costs: $c_i = 0$ for $i = 1, 2, \cdots, d$, $c_{uu} = 0$ for $u \in \mathcal{C}$ and $c_{uv} = 1$ for $u \neq v$, $\forall u, v \in \mathcal{C}$. Note that such a definition of costs causes the objective of the cost-sensitive classifier to maximize the classification accuracy. The results in Fig. 3(c) shows that the cost-sensitive classifier yields the near-best classification accuracy when the number of states is five for each class, which is consistent with the model selection results presented in Figs. 3(a) and 3(b).

To assess the performance of the POMDP and the myopic solutions for cost-sensitive classification, we apply the POMDP-repeat, the POMDP-norepeat and the myopic algorithm on the Pima data with the costs obtained directly

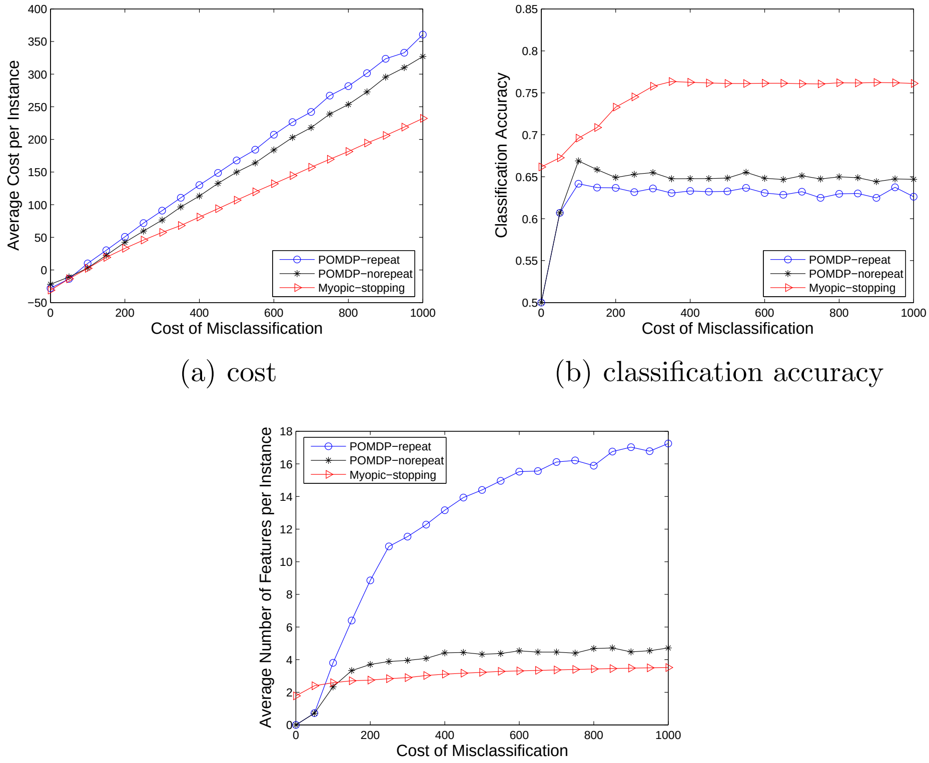
Table 2
Test costs for the Pima Indians Diabetes dataset [2].

Feature Index	**Description**	**Test Cost**
1 times pregnant	number of times pregnant	\$1.00
2 glucose tol	glucose tolerance	\$17.61
3 diastolic bp	diastolic blood pressure	\$1.00
4 triceps	triceps skin fold thickness	\$1.00
5 insulin	serum insulin test	\$22.78
6 mass index	body mass index	\$1.00
7 pedigree	diabetes pedigree function	\$1.00
8 age	age in years	\$1.00

from a real medical diagnosis. In Table 2 we summarize the test costs for the 8 features, based on the information from [2]. In addition, we also define the cost of getting a case correctly diagnosed $c_{uu} = -\$50$ (noting that negative cost is positive reward). By changing the cost of misclassification over the range $c_{uv} = [\$0, \$1000]$ for $u \neq v$, we compare the performances of all three methods evaluated on three quantities: (a) the average cost for diagnosing a "patient", (b) the classification accuracy, and (c) the average number of features acquired for each "patient".

For each experimental setting (i.e., each distinct c_{uv}), we perform 10 independent trials with each trail implemented on a training set of 512 instances and a testing set of 256 instances, generated by randomly splitting the 768 instances (as in [2]), with the average results shown in Fig. 4. As we can see, the myopic solution outperforms the POMDP-norepeat, and the POMDP-norepeat outperforms the POMDP-repeat, over all three quantities: it pays on average less cost for diagnosing a "patient", it has higher classification accuracy, and it uses on average much fewer tests for diagnosing a "patient". As discussed in Sec. 4, the inferior performance of the POMDP-repeat is caused mainly by repeated actions that acquire the same feature multiple times. Comparing the performance of the POMDP-norepeat with the POMDP-repeat, we know that these repeated actions do not aid or even deteriorate the classification performance, but drive cost up unnecessarily. Although both the POMDP-norepeat and the myopic algorithm can avoid repeated action, the POMDP-norepeat is not a non-myopic strategy in a rigorous sense as indicated by the inferior performance compared with the myopic algorithm. As a reference, the ICET algorithm presented in [2] obtained a classification accuracy of about 74% (measured from the figure), our myopic algorithm is about 76%, although the costs definition of the ICET is more complex than Table 2.

All computer code employed in this study was implemented in unoptimized



(a) cost

(b) classification accuracy

(c) number of features

Fig. 4. Performance of the POMDP and the myopic solution on the Pima dataset, compared on three quantities: (a) the average cost of diagnosing a "patient", (b) the classification accuracy, and (c) the average number of features queried for each "patient". The results are averaged on 10 trials, with each trail implemented on a training set of 512 instances and a testing set of 256 instances, generated by randomly splitting the 768 instances.

Matlab. However, to give a sense of the computational complexity, for each experimental setting the offline POMDP policy design required 1 hour of CPU with the PBVI algorithm [13], using a Pentium IV with 2.8 GHz CPU. In these computations the PBVI was implemented on a total of 16 states (averaged across 10 trails; model selection was performed separately for each set of training data), 10 actions (8 feature-acquisition actions and 2 classification actions), and 40 discrete observations. Selection of the actions for feature acquisition and final classification was essentially instantaneous based on the policy. In comparison, the myopic algorithm required 0.01 seconds of CPU per action, and about 12 seconds on all 256 test cases.

5.2 Performance on three benchmark datasets

In the next set of experiments we further assess the classification performances of the cost-sensitive classifier on the *Pima Indian diabetes*, the *ionosphere*,

Table 3
Classification performances on three benchmark datasets. The cost-sensitive classifier is tested with the defined costs: $c_i = 1$, $c_{uu} = -10$, and $c_{uv} = 200$. The quantities within the parentheses denote the average number of features queried for classifying a given example. We emphasize that both POMDP results are undermined by the respective set of approximations employed.

Method	**Pima**	**Ionosphere**	**WDBC**
Myopic algorithm	76.31% (2.59)	92.28% (3.19)	95.24% (2.57)
POMDP-norepeat	71.43% (4.92)	89.05% (4.08)	92.40% (2.61)
POMDP-repeat	68.16% (17.4)	82.82% (6.80)	90.83% (4.05)
SVM	77.34%†	88.60%†	96.65%*
Decision tree (C4.5)	73.83%†	91.45%†	n.a.
Linear discriminant	n.a.	n.a.	92.94%*
Gaussian process	77.37%$^\diamond$	92.01%$^\diamond$	97.03%*

† : the results are obtained from online reports [19] by using the world renowned WEKA data mining software developed at the University of Waikato, Hamilton, New Zealand;
* : the results are obtained from [17] by transferring the numbers of test errors into the correct percentages;
$^\diamond$: the results are obtained from [18] with the best reported correct percentages.

and the WDBC, with the defined cost functions: $c_i = 1$ for $i = 1, 2, \cdots, d$, $c_{uu} = -10$ for $u \in \mathcal{C}$ and $c_{uv} = 200$ for $u \neq v$, $\forall u, v \in \mathcal{C}$. Note that such a definition of costs attempts to force the cost-sensitive classifier to use as few features as possible while maximizing the classification accuracy. Thus, in this case, we can compare the cost-sensitive classifier with traditional classifiers for classification accuracy. For the Pima dataset, the problem description is the same as that of Sec. 5.1, but we use 10-fold cross validation to evaluate its classification performance (as in [18,19]). For the *ionosphere* data, there are a total of 351 samples with each sample represented by a 34-dimenional feature vector; the results were obtained with 10-fold cross validation (as in [18,19]). For the WDBC problem, there are a total of 569 instances with each instance having 30 features; the results reported were obtained by averaging over 30 random partitions with 300 training instances and 269 testing instances (as in [17]). Table 3 reports the classification performances achieved by the myopic algorithm, the POMDP-norepeat, the POMDP-repeat, and several other state-of-the-art techniques in literature. On all datasets considered, the myopic algorithm consistently outperforms the POMDP-repeat and the POMDP-norepeat on the classification accuracy and on the number of features queried. Again, as expected, the POMDP-norepeat outperforms the POMDP-repeat by removing repeated actions. We also note that as the classification domain increases (larger d), the issue of repeated actions becomes less severe, as observed by the decreasing difference between the numbers of features queried by the myopic algorithm and by the POMDP-repeat. Table 3 also shows that the myopic algorithm yields classification accuracies that are close to or even better than state-of-the-art algorithms, and importantly the myopic algorithm uses only a small fraction of the features that are generally used in the competitive approaches.

6 Conclusions

Feature selection and classification are framed jointly as a cost-sensitive classification problem, solved by the POMDP technique and by a myopic algorithm simplification that employs a stopping criterion linked to the standard POMDP. Our approach for feature selection and classification is different from traditional methods (see [20] and the references therein): (i) the traditional methods do not consider the costs of querying features and do not aim to balance the costs of acquiring features with Bayes risks; (ii) our method selects the feature in a sequential manner until no additional feature acquisition is justified followed by a classification, while traditional methods, such as kernel machines [20], select all the useful features simultaneously by setting the weights on the redundant features to zero. Encouragingly, our approach obtains similar classification performance relative to state-of-the-art algorithms for the three benchmark datasets considered.

While the non-myopic POMDP constitutes an intuitively appealing formulation to the cost-sensitive classification problem, the myopic approach may be more effective in terms of computational complexity, handling continuous features and avoiding repeated actions. The only bottleneck of the proposed method is in the learning phase for building the HMM classifier. Once the model has been learned, the feature selection and classification are accomplished efficiently. For problems with very large sets of features, the fast HMM learning algorithms (e.g., [21]) could be applied to speedup HMM learning, and enable the proposed method to be applicable on larger problems.

While the results of the myopic approach appear promising, based on the set of results reported here, there may be some examples for which a non-myopic approach may be preferable. For example, in some problems a sequence of inexpensive tests (features) may be ineffective individually, but may collectively by highly discriminative. A myopic approach may not select such features, rather seeking a single feature that may be discriminative but highly expensive. Future research is warranted to this non-myopic POMDP policy design while avoiding repeated actions, especially for the classification domains with a large number of features; this will entail development of non-stationary policies that keep track of which features have been selected.

A HMM parameter learning

In Sec. 2.1 we provided a motivation for modeling the sequential acquisition of features as an HMM, with the underlying states linked to the statistics of the features. In practice, to obtain the HMM parameters $\theta = \{\pi, \tau, \phi\}$, learning is required based on the training data. Assume we are given a set of training items $\{\boldsymbol{x}_i\}_{i=1}^{L}$ from a class, where $\boldsymbol{x}_i = (x_1^i, x_2^i, \cdots, x_d^i)$ represents the ith training item with a feature vector length of d. Since for each d-dimensional feature vector there are $d!$ distinct feature-acquisition processes (corresponding to all possible ways of acquiring these d features), then totally $L \times d!$ feature-acquisition processes can be constructed for training an HMM. In practice, this is a large dataset even for the classification problem with small d. To speedup the HMM training, we randomly select a small set of $p \ll d!$ feature-acquisition sequences from each feature vector, and form a training set of size $L \times p$. To learn the HMM parameters, we can use the expectation-maximization (EM) algorithm [7, 22] or the variational Bayesian (VB) method [10, 11, 23], as discussed in Appendix B.

B Variational Bayes model selection for HMMs

One of the key tasks in the application of HMM is to determine a suitable number of HMM states. There are many approaches to model selection for graphical models, e.g., HMMs. One may refer to [23, 24] for a general introduction to this problem. Among all the methods, variational Bayes (VB) represents a principled and practical approache for model selection. We present the basic ideas of the VB approach for the case of HMMs, and for more details one may refer to [10, 11, 23].

Assume that $\mathcal{D}$ represents the complete set of sequential data associated with one class, and the integer M represents the number of states considered for the HMM. The marginal likelihood or "evidence" for an M-state HMM is represented as

$$p(\mathcal{D}|M) = \int p(\mathcal{D}|\theta, M)p(\theta|M)d\theta \tag{B.1}$$

where θ denotes the HMM parameters. The integration in (B.1) is typically computationally intractable, even in very simple cases. Most existing methods, such as Markov Chain Monte Carlo (MCMC) and the Laplace approximation [24] either require vast computational resources to obtain accurate estimates or crudely approximate all the posteriors via a normal distribution. Between these two extremes, the VB method attempts to approximate the integration as accurately as possible while remaining computationally tractable. This is accomplished via the approximation of the integration (B.1) by a lower bound

[23], which can be derived from a fundamental relationship between the log-likelihood, negative free energy and the Kullback-Leibler (KL) divergence as follows:

$$\log p(\mathcal{D}) = F(q) + KL(q||p) \tag{B.2}$$

with

$$F(q) = \int q(S,\theta) \log \frac{p(\mathcal{D}, S, \theta)}{q(S,\theta)} dS d\theta \tag{B.3}$$

$$KL(q||p) = \int q(S,\theta) \log \frac{q(S,\theta)}{p(S,\theta|\mathcal{D})} dS d\theta \tag{B.4}$$

where S is the hidden state sequence over which the data sequence $\mathcal{D}$ is observed, $q(S,\theta)$is the variational posterior over model parameters and hidden variables, and $p(S,\theta|\mathcal{D})$ is the true posterior density that to be estimated. Since the KL divergence is non-negative and is zero for identical distributions, this indicates that $F(q)$ is a strict lower bound on $\log p(\mathcal{D})$,

$$\log p(\mathcal{D}) \geq F(q) \tag{B.5}$$

with equality if the variational posterior density equals the true posterior density, i.e., $q(S,\theta) = p(S,\theta|\mathcal{D})$.

The objective of VB is to maximize this lower bound by tuning the variational posterior $q(S,\theta)$ such that as the variational posterior approaches the true posterior $p(S,\theta|\mathcal{D})$, the bound becomes tight, thus the marginal log-likelihood $\log p(\mathcal{D})$ can be approximated efficiently by $F(q)$. To make this maximization tractable, VB relies on the concept of conjugate priors and the factorization of variational posterior, i.e., $q(S,\theta) = q(S)q(\theta)$, with the resulted algorithm generalizing the standard Expectation Maximization (EM) algorithm, whose convergence is guaranteed.

To understand why the VB objective function $F(q)$ is a good score for model selection, it is useful to rewrite (B.3) as

$$F(q) = \left\langle \log \frac{p(\mathcal{D}, S|\theta)}{q(S)} \right\rangle_{S,\theta} - KL(q(\theta)||p(\theta)) \tag{B.6}$$

where the average in the first term is taken with respect to $q(S,\theta)$. The first term corresponds to the average log-likelihood, representing how well the model fits the data. The second term is the KL distance between the prior and posterior over the parameters. As the number of parameters increases, the KL distance follows and consequently penalizes the complex model.

References

[1] R. Greiner, A. Grove, D. Roth, Learning active classifiers, in: Proc. of the Thirteenth International Conference on Machine Learning, 1996.

[2] P. D. Turney, Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm.

[3] A. Y. Ng, M. I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, in: Proc. Advances in Neural Information Processing Systems, Vol. 14.

[4] L. P. Kaelbling, M. L. Littman, A. R. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence 101 (1998) 99–134.

[5] L. Pedersen, M. Wagner, D. apostolopoulos, W. Whittaker, Autonomous robotic meteorite identification in antarctica, in: IEEE International Conference on Robotics and Automation, 2001.

[6] A. Guo, Decision-theoretic active sensing for autonomous agents, in: Proc. of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2003.

[7] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proc. IEEE 77 (1989) 257–286.

[8] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge University Press, 1998.

[9] Y. Bengio, P. Frasconi, An input/output hmm architecture, in: Proc. Advances in Neural Information Processing Systems 7, 1995.

[10] D. MacKay, Ensemble learning for hidden markov models, Tech. rep., Department of Physics, University of Cambridge (1997).

[11] S. Ji, B. Krishnapuram, L. Carin, Variational bayes for continuous hidden markov models and its application to active learning, IEEE Trans. Pattern Analysis and Machine Intelligence 28 (2006) 522– 532.

[12] P. D. Turney, Types of cost in inductive learning, in: Workshop on Cost-Sensitive Learning at the 17th ICML, 1996, pp. 15–21.

[13] J. Pineau, G. Gordon, S. Thrun, Point-based value iteration: An anytime algorithm for pomdps, in: Proc. Int. Joint Conf. On Artificial Intelligence, Acapulco, Mexico, 2003.

[14] M. Hauskrecht, Value-function approximations for partially observable markov decision processes, Journal of Artificial Intelligence Research 13 (2000) 33–94.

[15] C. Papadimitriou, J. Tsitsiklis, The complexity of markov decision processes, Mathematics of Operations Research 12 (3) (1987) 441–450.

[16] C. M. Bishop, Neural networks for pattern recognition, Oxford University Press, 1995.

[17] M. Seeger, Bayesian model selection for support vector machines, gaussian processes, and other kernel classifiers, in: Proc. Advances in Neural Information Processing Systems 12, 2000, pp. 603–609.

[18] M. Kuss, C. E. Rasmussen, Assessing approximations for gaussian process classification, in: Proc. Advances in Neural Information Processing Systems 18, 2006.

[19] Automatic knowledge miner, reports for benchmark data sets.
URL `http://www.auknomi.com/reports.html.`

[20] F. Li, Y. Yang, E. Xing, From lasso regression to feature vector machine, in: Proc. Advances in Neural Information Processing Systems 18, 2006.

[21] S. Siddiqi, A. Moore, Fast inference and learning in large-state-space hmms, in: Proc. of the 22nd International Conference on Machine Learning, 2005.

[22] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society B 39 (1977) 1–38.

[23] H. Attias, A variational bayesian framework for graphical models, in: Proc. Advances in Neural Information Processing Systems 12, 2000.

[24] D. M. Chickering, D. Heckerman, Efficient approximations for the marginal likelihood of bayesian networks with hidden variables, Machine Learning 29 (2-3) (1997) 181–212.