A REWARD-DIRECTED BAYESIAN CLASSIFIER

Hui Li, Xuejun Liao, and Lawrence Carin

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

ABSTRACT

We consider a classification problem wherein the class features are not given *a priori*. The classifier is responsible for selecting the features, to minimize the cost of observing features while also maximizing the classification performance. We propose a reward-directed Bayesian classifier (RDBC) to solve this problem. The RDBC features an internal state structure for preserving the feature dependence, and is formulated as a partially observable Markov decision process (POMDP). The results on a diabetes dataset show the RDBC with a moderate number of states significantly improves over the naive Bayes classifier, both in prediction accuracy and observation parsimony. It is also demonstrated that the RDBC performs better by using more states to increase its memory.

1. INTRODUCTION

A traditional Bayesian classifier can be viewed as a 4-tuple $\langle \mathcal{C}, \mathcal{X}, \mathcal{O}, \Omega_{c,o_1o_2\cdots o_d} \rangle$, where $\mathcal{C}$ is a finite set of class labels and $\mathcal{X} = \{x_1, x_2, \cdots, x_d\}$ is a finite set of class features, $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2 \times \cdots \times \mathcal{O}_d$ with $\mathcal{O}_i$ defining the set of possible observations of x_i, and Ω is the observation function with $\Omega_{c,o_1o_2\cdots o_d}$ denoting the probability of observing $[o_1, o_2, \cdots, o_d] \in \mathcal{O}$ given class label $c \in \mathcal{C}$. The goal of Bayesian classification is to correctly predict the class label of any given observation vector in $\mathcal{O}$. Denoting by $p(c)$ the *prior* distribution of class labels, its *posterior* distribution is computed by Bayes rule,

$$p(c|o_1, o_2, \cdots, o_d) = \frac{p(o_1, o_2, \cdots, o_d|c)p(c)}{\sum_{c\in\mathcal{C}} p(o_1, o_2, \cdots, o_d|c)p(c)} \quad (1)$$

A traditional Bayesian classifier makes predictions based on observations of all features in $\mathcal{X}$, with no mechanism for selecting the features to observe.

In many applications such as medical diagnosis, observing a feature may entail expensive instrumental measurement and time-consuming analysis. Given a limited budget, time, or other resources, it may not possible to observe all features. Moreover, some features may not be as helpful to diagnosis as others. Selectively observing the most useful features is important in minimizing the cost (negative reward). In other respects, some diseases may be more serious and require more accurate prediction than others. In such scenarios the classifier must jointly maximize prediction accuracy and observation reward (negative cost) by quantifying the reward/cost in a unified manner. In this paper we refer to this type of classification as reward-directed classification.

The problem of reward-directed classification has been investigated perviously by Bonet and Geffner [1], and Guo [2], under the naive Bayes assumption that the features $[x_1, x_2, \cdots, x_d]$ are independent conditional on the class label, i.e., $p(o_1, o_2, \cdots, o_d|c) = \prod_{i=1}^{d} p(o_i|c)$ for all $[o_1, o_2, \cdots, o_d] \in \mathcal{O}$. This assumption is very strong and can result in serious degraded classification performance in real applications, where the assumption is often violated.

In this paper we propose a reward-directed classification algorithm in which the naive Bayes assumption is relaxed. The key idea is to use a Markov chain as an internal representation of feature dependence. We demonstrate using a real medical data set that a Markov chain with a moderate number of states can significantly improve the classification accuracy as well as reduce observation cost.

2. THE PROPOSED REWARD-DIRECTED BAYESIAN CLASSIFIER (RDBC)

2.1. Intuitive Description of the RDBC

Before proceeding to the mathematical formulation, we give an intuitive description of the RDBC, emphasizing the aspects in which it is different from the traditional Bayesian classifier.

The features used by the RDBC for prediction are not given *a priori* and the RDBC is responsible to choose the features to use, from a given feature set $\mathcal{X}$. The features are selected and observed sequentially. Assume the RDBC is instructed to observe n features and a given feature can be repeatedly observed. At the time of making the i-th observation, the RDBC has collected a list of past observations and the associated feature indices $\varepsilon_i = [a_0o_1, \cdots, a_{i-2}o_{i-1}]$, where o_j is an observation of feature $x_{a_{j-1}}$, $j = 1 \cdots i-1$. See Figure 1 for a graphical illustration of the relations of o and a. In choosing a_{i-1} (the feature index of o_i), the RDBC takes into account the list ε_i and the conditional distribution

$p(o_i o_{i+1} \cdots o_n |\ \varepsilon_i,\ a_{i-1} a_i^* \cdots a_{n-1}^*)$, where a_{j-1}^*, $i+1 \leq j \leq n$, is the optimal feature index for o_j given the RDBC is instructed to observe $n-j+1$ features $[o_j \cdots o_n]$. A policy of feature selection is learned with the goal of simultaneously maximizing the reward of correct prediction and minimizing the cost of observation and false prediction.

The RDBC uses an internal Markov chain to represent the feature dependence of a given class. Let $o_1, \cdots, o_n$ be the observations of n features $x_{a_0}, \cdots, x_{a_{n-1}} \in \mathcal{X}$, respectively. The RDBC expresses the class-conditional probability as

$$\begin{aligned} & p(o_1, \cdots, o_n | c, a_0, \cdots, a_{n-1}) \\ &= \textstyle\sum_{s_0 \cdots s_n \in \mathcal{S}_c} p(o_1, \cdots, o_n, s_0, \cdots, s_n | a_0, \cdots, a_{n-1}) \end{aligned} \quad (2)$$

where s_i is the internal state of o_i, $i = 1 \cdots n$, $\mathcal{S}_c$ is a finite set of internal states defined for class c, and s_0 is an initial state. See Figure 1 for a graphical illustration of the relations of s, o, and a.

It is clear that such a representation is sensitive to the order of $\{o_1 \cdots o_n\}$ and the associated $\{a_0 \cdots a_{n-1}\}$, which implies that different permutations of $\{(a_0 o_1), \cdots, (a_{n-1} o_n)\}$ appear different to the representation. This order information is necessary in the sequential feature selection process. However, the order sensitivity may make $p(o_1 \cdots o_n | c,\ a_0 \cdots a_{n-1})$ different for different permutations of $\{(a_0 o_1), \cdots, (a_{n-1} o_n)\}$, which is harmful as this probability is being treated as the joint probability of $\{o_1, \cdots, o_n\}$ conditional on $\{c, a_0, \cdots, a_{n-1}\}$ and should remain invariant regardless of the order. To preserve the order-invariance, training of this representation (i.e., estimation of its state transition probabilities and observation probabilities) must be based on a sufficient number of permutations of each $\{(a_0 o_1), \cdots, (a_{n-1}\ o_n)\}$ to make the permutations equally probable in the resulting representation.

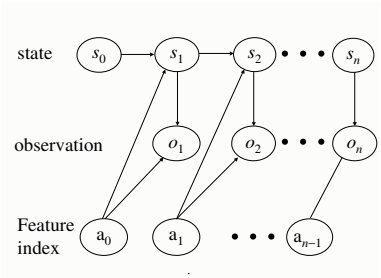

Fig. 1. Representation of feature dependence for a given class in the RDBC. Each node is dependent on (and only on) the nodes that emanates a directed edge to it. Though the internal state s is Markovian, the observation o is not, therefore the dependence among $o_1 \cdots o_n$ is well represented.

2.2. Mathematical Formulation of the RDBC

The proposed RDBC can be formulated as a Partially Observable Markov Decision Process (POMDP) [3] with a specialized state structure. Specifically, the RDBC is defined as a 8-tuple $\langle \mathcal{C}, \mathcal{X}, \mathcal{O}, \mathcal{S}, \mathcal{A}, T_{ss'}^a, \Omega_{s'o}^a, R \rangle$, where $\mathcal{C}$ is a finite set of class labels and $\mathcal{X} = \{x_1, x_2, \cdots, x_d\}$ is a finite set of class features; the remaining 6 elements are the elements in a standard POMDP and they are specified below.

The $\mathcal{O}$ is a union of disjoint sets $\mathcal{O}_1$, $\mathcal{O}_2$, $\cdots$, and $\mathcal{O}_d$, with $\mathcal{O}_i$ denoting the set of possible observations of x_i. The $\mathcal{S}$ is a union of disjoint sets $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{|\mathcal{C}|}$, and $\{t\}$, with $\mathcal{S}_c$ the set of internal states for class c, t the terminal state, and $|\mathcal{C}|$ denoting the cardinality of $\mathcal{C}$. The $\mathcal{A} = \{1, \cdots, d, d+1, \cdots, d+|\mathcal{C}|\}$ is the set of possible actions; letting a be an action variable, $a = i$ denotes "observing feature x_i" and $a = d + c$ denotes "predicting as class c".

The T are the state-transition matrices with $T_{ss'}^a$ denoting the probability of transiting to state s' by taking action a in state s. The RDBC prohibits transition between internal states of different classes, therefore $T_{ss'}^a = 0$, $\forall\ a \in \mathcal{A}$, $s \in \mathcal{S}_c$, $s' \in \mathcal{S}_{c'}$, $c \neq c'$. In addition, the RDBC has a probability-one transition from any non-terminal state to the terminal state when the action a is "predicting", i.e., $T_{ss'}^a = 1, \forall\ d+1 \leq a \leq d+|\mathcal{C}|, s \neq t, s' = t$; and it has a uniformly random transition from the terminal state to an internal state of any class when the action a is "observing a feature", i.e., $T_{ss'}^a = 1/(|\mathcal{S}_c|\,|\mathcal{C}|), \forall\ 1 \leq a \leq d,\ s = t,\ s' \in \mathcal{S}_c$. The state transitions in the RDBC are illustrated in Figure 2 for a two-class problem ($|\mathcal{C}| = 2$), with two internal states defined for class 1 ($|\mathcal{S}_1| = 2$) and three internal states defined for class 2 ($|\mathcal{S}_2| = 3$).

The Ω are the observation functions with $\Omega_{s'o}^a$ denoting the probability of observing o after performing action a and transiting to state s'. The R is the reward function with $R(s, a)$ specifying the expected immediate reward that is received by taking action a in state s.

Using the definitions of the RDBC, we have the expansion

$$p(o_1 \cdots o_n, s_0 \ldots s_n | a_0 \cdots a_{n-1}) = p(s_0) \textstyle\prod_{i=1}^n T_{s_{i-1} s_i}^{a_{i-1}} \Omega_{o_i s_i}^{a_{i-1}} \quad (3)$$

where we assume that given class c the initial state is uniformly distributed in $\mathcal{S}_c$, i.e., $p(s_0) = \frac{1}{|\mathcal{S}_c|}$ given class c. When $|\mathcal{S}_c| = 1$, we have $p(s_i | s_{i-1}, a_{i-1}) = 1$ and consequently $p(o_1 \cdots o_n,\ s_0 \cdots s_n | a_0 \cdots a_{n-1}) = p(s_0) \prod_{i=1}^n p(o_i | s_i, a_{i-1})$, which is substituted into (2) to get

$$p(o_1, \cdots, o_n | c, a_0, \cdots, a_{n-1}) = \textstyle\prod_{i=1}^n p(o_i | c, a_{i-1}) \quad (4)$$

Equation (4) shows that the distribution of observations conditional class c reduces to a naive Bayes expression when a single state is defined for class c. This demonstrates that in order to capture feature dependence of a class, multiple states must be defined for the class.

2.3. Learning of the RDBC

To learn the RDBC, one first obtain $\mathcal{C}$, $\mathcal{X}$, $\mathcal{O}$, and $\mathcal{A}$ from the problem, determine $|\mathcal{S}_c|$ the number of internal states for each class c, and then estimate the transition matrices T and

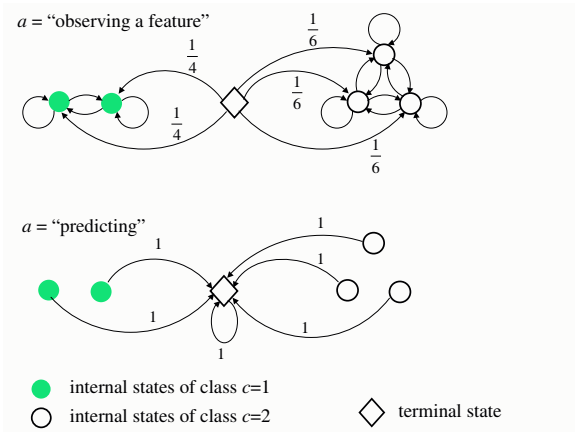

Fig. 2. An illustration of state transitions in the proposed RDBC. A solid circle denotes a state of class 1; a hollow circle denotes a state of class 2; the diamond denotes the terminal state. A directed edge connecting two states denotes a transition from the initial state to the destination state; the number marked by the edge denotes the probability of the associated state transition; an edge with no numbers indicates that the associated transition probability is to be estimated from training data.

observation functions Ω from a training data set, using the standard Expectation-Maximization (EM) method [4]. Upon completion of these, one obtains the RDBC representation of the class-conditional distribution of observations, as given by (2) and (3).

One then determines a reward function R according to the objective in the problem, and learns a policy for choosing the actions. The goal in policy learning is to maximize the expected future reward (value) [3]. The most widely used policy learning method for POMDP is value iteration. Denote by V_n the value function when looking n steps ahead (i.e., with a horizon length n), value iteration iteratively estimate V_n, starting from $n = 0$ and proceeding backwards to the desired horizon length N. Exact value iteration for POMDP is usually intractable because the computation grows exponentially with horizon length n. Approximate methods must be used instead, of which the point-based value iteration (PBVI) [5] is an efficient algorithm with a computation complexity growing polynomially with n. The PBVI represents a practical algorithm and we use it to learn the policy in our experiment.

3. EXPERIMENTAL RESULTS

We evaluate the performance of the proposed RDBC on the Pima Indians Diabetes dataset [6], a public data set available at http://www.ics.uci.edu/ mlearn/MLSummary.html. The dataset consists of 768 medical instances for diabetes diagnosis. Each instance consists of 8 features, representing 8 distinct medical measurements. The observation costs of the 8 features, which are summarized in Table 1, are based on information from the Ontario Ministry of Health (1992) [7]. Each feature is quantized into 5 uniform bins, yielding a set of $8 \times 5 = 40$ possible observations, i.e., $|\mathcal{O}| = 40$. Each instance has a diagnostic result of either "healthy" or "diabetes", which are referred to class 1 and class 2 in our results. The 768 instances are randomly split into a training set of 512 instances and a testing set of 256 instances. For each experimental setting, we perform 10 independent trials of the random split and generate the mean and standard deviation of the results from the 10 trials.

Table 1. Observation Cost of the Pima dataset

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

There are 10 actions (i.e., $|\mathcal{A}| = 10$), including 8 observation actions and 2 prediction actions. We consider three configurations of internal states for the two classes. In the first configuration, class 1 has 6 internal states and class 2 has 5; in the second configuration, both classes have 10 internal states; in the third configuration, both classes have 1 internal state, which is the naive Bayes case. For a given state configuration, the reward function $R(s, a)$ is constructed as follows: when action a is one of the 8 observation actions, $R(s, a) = -$(cost of x_a) regardless of s; when action a is one of the 2 prediction actions, $R(s, a) = \$50$ if $s \in \mathcal{S}_a$ (correct prediction) and $R(s, a) = -\lambda$ if $s \notin \mathcal{S}_a$ (false prediction), where λ is the cost of a false prediction. We vary λ in the range $[\$0, \$200]$ and present each result as a function of λ.

The state transition probabilities involving the terminal state are computed analytically as in Section 2.2. The remaining entries of $T^a_{ss'}$ as well as Ω_{os} are estimated from the training data set. For each training instance, the 8 observations (of 8 features), denoted $o_1, o_2, \cdots, o_8$, are randomly permutated to produce 20 permutated versions of $\{(a_0 o_1), (a_1 o_2), \cdots, (a_7 o_8)\}$ (where $a_0 = 1, a_1 = 2, \cdots, a_7 = 8$). The 512 training instances yield 512×20 permutations in total, which are used to estimated $T^a_{ss'}$ and Ω_{os}. The PBVI [5] is used to learn the policy.

In testing, the policy is followed until a prediction action is selected and executed to make s transit to the terminal state to complete the present prediction phase. We compute three performance indexes at the end of each prediction phase: correct classification rate, observation cost accumulated, and feature repetition rate. Assume that at the end of a prediction phase, n observations are made of $m < n$ features (some features are observed more than once), then the feature repetition

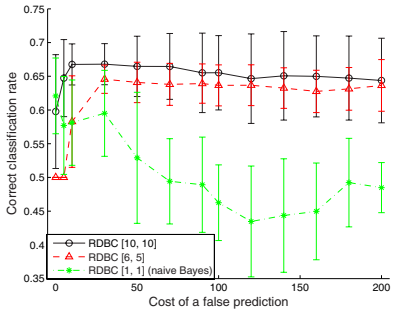

Fig. 3. Correct classification rate as a function of false prediction cost. The mean and error bars are generated from 10 independent trials of random split of training and test instances.

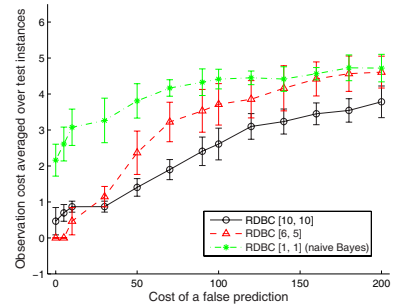

Fig. 4. Observation cost averaged over test instances, as a function of false prediction cost. The mean and error bars are generated from 10 independent trials of random split of training and test instances.

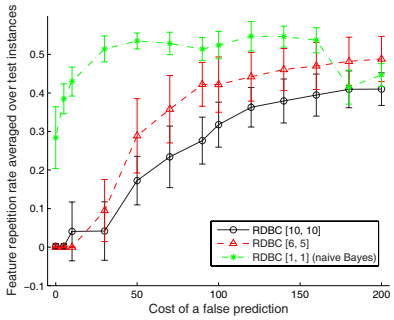

Fig. 5. Feature repetition rate averaged over test instances, as a function of false prediction cost. The mean and error bars are generated from 10 independent trials of random split of training and test instances.

rate is computed as $(n - m)/n$.

The results obtained on the Pima data are summarized in Figures 3, 4, and 5. In each of the figures, black solid line denotes RDBC with 10 internal states for each class, red dashed line denotes RDBC with 6 internal states for class 1 and 5 internal states for class 2, green dotted line denotes the RDBC with 1 internal state for each class (the naive Bayes case).

Figures 3 and 4 show that, with a larger number of internal states for each class, higher correct classification rates are achieved at lower observation costs. This striking comparison can be explained by Figure 5, which shows that with increased internal states, the feature repetition rate is reduced. In the Pima data set, the features are noise free, so there is no sense in observing a given feature multiple times. The only reason that could lead to repetitive observation of the same feature is that the classifier is memoryless and does not remember that it has observed a feature before. It is obvious that a single state for each class does not provide memory to the classifier and therefore the naive Bayes classifier has the highest feature repetition rate. In contrast, the RDBC with 10 states for each class has the best memory, which gives it the lowest feature repetition rate. Repetitively observing the same feature is harmful in the Pima data: it increases cost and yet provides no new information to improve classification. This explains Figures 3 and 4.

4. CONCLUSIONS

We have presented a reward-directed Bayesian classifier (RDBC) that preserves the feature dependence in its internal states. The proposed RDBC is formulated as a POMDP. The results on a diabetes dataset show the RDBC with a moderate number of states significantly improves over the naive Bayes classifier, both in prediction accuracy and observation parsimony. It is also demonstrated that the RDBC performs better by using more states to increase its memory.

5. REFERENCES

[1] B. Bonet and H. Geffner, "Learning sorting and decision trees with pomdps," *International Conference on Machine Learning (ICML)*, 1998.

[2] A. Guo, "Decision-theoretic active sensing for autonomous agents," *AAMAS*, July 2003.

[3] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, 1998.

[4] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77(2), pp. 257–285, 1989.

[5] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for pomdps," in *International Joint Conference on Artificial Intelligence (IJCAI)*, August 2003, pp. 1025 – 1032.

[6] P. D. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, vol. 2, pp. 369–409, 1995.

[7] Ontario Ministry of Health, "Schedule of benefits: Physician services under the health insurance act," *Ontario: Ministry of Health*, October 1 1992.