Discriminative Learning can Succeed where Generative Learning Fails [⋆]

Philip M. Long [a] Rocco A. Servedio [b,∗,1] Hans Ulrich Simon [c]

[a] *Google, Mountain View, CA, USA*

[b] *Columbia University, New York, New York, USA*

[c] *Ruhr-Universität Bochum, Bochum, Germany*

Abstract

Generative algorithms for learning classifiers use training data to separately estimate a probability model for each class. New items are classified by comparing their probabilities under these models. In contrast, discriminative learning algorithms try to find classifiers that perform well on all the training data.

We show that there is a learning problem that can be solved by a discriminative learning algorithm, but not by any generative learning algorithm. This statement is formalized using a framework inspired by previous work of Goldberg [4].

Key words: algorithms, computational learning theory, discriminative learning, generative learning, machine learning

1 Introduction

If objects and their classifications are generated randomly from a joint probability distribution, then the optimal way to predict the class y of an item x is to maximize $\Pr[y|x]$. Applying Bayes' rule, this is equivalent to maximizing $\Pr[x|y]\Pr[y]$. This motivates what has become known as the *generative* approach to learning a classifier, in which the training data is used to learn

⋆ Reference [8] is a preliminary version of this paper but contains a flaw; see the more detailed note at the end of Section 1.

∗ Corresponding author.
Email address: rocco@cs.columbia.edu (Rocco A. Servedio).

1 Supported in part by NSF CAREER award CCF-0347282, by NSF award CCF-0523664, and by a Sloan Foundation Fellowship.

$\Pr[\cdot|y]$ and $\Pr[y]$ for the different classes y, and the results are used to approximate the behavior of the optimal predictor for the source (see [2,6]).

In the *discriminative* approach, the learning algorithm simply tries to find a classifier that performs well on the training data [12,6,10,7]. Discriminative algorithms can (and usually do) process examples from several classes together at once, e.g. maximum margin algorithms use both positive and negative examples together to find a large margin hypothesis separating the two classes.

The main result of this paper is a separation between generative and discriminative learning. We describe a learning problem and prove that it has the following property: a discriminative algorithm can solve the problem, but no generative learning algorithm can.

Our analysis demonstrates the possible cost of largely processing the examples from different classes separately, as generative methods do. Goldberg [4,5] was the first to study the effect of this limitation. He studied a modification of the PAC model in which

- the examples belonging to each class are analyzed separately,
- each analysis results in a scoring function for that class, and
- future class predictions are made by comparing the scores assigned by the different scoring functions.

He designed algorithms that provably solve a number of concrete learning problems despite the constraint of processing examples from different classes separately, and identified conditions that allow a discriminative PAC learner to be modified to work in the generative setting. The main open question posed in [4] is whether there is a learning problem that can be solved by a discriminative algorithm but cannot be solved by a generative algorithm. We establish our main result in a framework closely related to the one proposed in [4]. The main difference between our formulation and Goldberg's is that we define a learning problem to be a collection of possible joint probability distributions over items and their classifications, whereas Goldberg defined a learning problem to be a concept class as in the PAC model.

Related work. Aside from Goldberg's paper, the most closely related work known to us is due to Ng and Jordan [9]. They showed that Naive Bayes, a generative algorithm, can converge to the large-sample limit of its accuracy much more quickly than a corresponding discriminative method. For generative algorithms that work by performing maximum likelihood over restricted classes of models, they also showed, given minimal assumptions, that the large-sample limit of their accuracy is no better than a corresponding discriminative method. Note that these results compare a particular generative algorithm with a particular discriminative algorithm. In contrast, the analysis in this paper exposes a fundamental limitation faced by any generative learning algorithm, due to the fact that it processes the two classes separately.

Note. A preliminary version of this work [8] claimed a *computational* separation between discriminative and generative learning based on a cryptographic construction, but the proof was flawed. The current note deals only with the *information-theoretic* abilities and limitations of discriminative and generative algorithms, i.e. we are only concerned with sample complexity and not with the running time of learning algorithms.

Section 2 contains preliminaries including a detailed description and motivation of the learning model. In Section 3 we give our construction of a learning problem that separates the two models. Section 4 gives the proof.

2 Definitions and main result

Given a domain X, we say that a *source* is a probability distribution P over $X \times \{-1, 1\}$, and a *learning problem* $\mathcal{P}$ is a set of sources.

2.1 Discriminative learning

The discriminative learning framework that we analyze is the *Probably Approximately Bayes* (PAB) [1] variant of the PAC [11] learning model. In the PAB model, in a learning problem $\mathcal{P}$ a learning algorithm A is given a set of m labeled examples drawn from an unknown source $P \in \mathcal{P}$. The goal is to, with probability $1 - \delta$, output a hypothesis function $h : X \to \{-1, 1\}$ which satisfies $\Pr_{(x,y)\sim P}[h(x) \neq y] \leq \mathrm{Bayes}(P) + \epsilon$, where $\mathrm{Bayes}(P)$ is the least error rate that can be achieved on P, i.e. the minimum, over all functions h, of $\Pr_{(x,y)\sim P}[h(x) \neq y]$.

We say that $\mathcal{P}$ is PAB-learnable if for any $\epsilon, \delta > 0$ there is a number $m = m(\epsilon, \delta)$ of examples such that A achieves the above goal for any source $P \in \mathcal{P}$.

2.2 Generative learning

Goldberg [4] defined a restricted "generative" variant of PAC learning. Our analysis will concern a natural extension of his ideas to the PAB model.

Roughly speaking, in the generative model studied in this paper, the algorithm first uses only positive examples to construct a "positive scoring function" $h_+ : X \to \mathbf{R}$ that assigns a "positiveness" score to each example in the input domain. It then uses only negative examples to construct (using the same algorithm) a "negative scoring function" $h_- : X \to \mathbf{R}$ that assigns a "negativeness" score to each example. The classifier output by the algorithm is the following: given example x, output 1 or -1 according to whether or not $h_+(x) > h_-(x)$.

We now give a precise description of our learning framework. In our model

- A sample $S = (x_1, y_1),\ldots,(x_m, y_m)$ is drawn from the unknown source P;
- An algorithm A is given a filtered version of S in which
  · examples (x_t, y_t) for which $y_t = 1$ are replaced with x_t, and
  · examples (x_t, y_t) for which $y_t = -1$ are replaced with $\diamond$
  and A outputs $h_+ : X \to \mathbf{R}$.
- Next, the same algorithm A is given a filtered version of S in which
  · examples (x_t, y_t) for which $y_t = 1$ are replaced with $\diamond$, and
  · examples (x_t, y_t) for which $y_t = -1$ are replaced with x_t
  and A outputs $h_- : X \to \mathbf{R}$.
- Finally, let $h : X \to \{-1, 1\}$ be defined as $h(x) = \operatorname{sgn}(h_+(x) - h_-(x))$. If $h_+(x) = h_-(x)$ then we view $h(x)$ as outputting $\perp$ (undefined).

Algorithm A is said to be a *generative PAB learning algorithm* for $\mathcal{P}$ if for all $P \in \mathcal{P}$, for all $0 < \epsilon < \frac{1}{2}$, $0 < \delta < 1$, there is a sample size $m = m(\epsilon, \delta)$ such that, given m examples, the hypothesis h obtained as above, with probability at least $1 - \delta$, satisfies $\Pr_{(x,y)\sim P}[h(x) \neq y] \leq \text{Bayes}(P) + \epsilon$.

It is easy to see that any learning problem that can be PAB learned in the generative framework we have described can also be learned in the standard PAB framework.

2.3 Main result

With these definitions in place we can state our main result:

Theorem 2.1 *There is a learning problem $\mathcal{P}$ that is learnable in the PAB model, but not in the generative PAB model.*

3 The construction

The domain is $X = \{0,1\}^* \times \{1,2,3\}$. With every $n \geq 1$ and every $\mathbf{r}, \mathbf{s} \in \{0,1\}^n$, we associate a source $P_{\mathbf{r},\mathbf{s}}$ with support contained in $\{0,1\}^n \times \{1,2,3\}$ and given as follows:

- It assigns probability $1/3$ to the pair $((\mathbf{r}, 1), 1)$ (that is, item $(\mathbf{r}, 1)$ and class 1).
- It assigns probability $1/3$ to the pair $((\mathbf{s}, 2), -1)$ (item $(\mathbf{s}, 2)$ and class -1).
- It assigns probability $\frac{1}{3n}$ to each $((\mathbf{e}_i, 3), (-1)^{r_i \oplus s_i})$, where $\mathbf{e}_i \in \{0, 1\}^n$ is the vector that has a 1 in the ith coordinate and zeroes everywhere else. Here $r_i \oplus s_i$ denotes the exclusive-or of the i-th components of $\mathbf{r}$ and $\mathbf{s}$.

The problem $\mathcal{P}$ witnessing the separation of Theorem 2.1 consists of all such $P_{\mathbf{r},\mathbf{s}}$. Note that for any source $P_{\mathbf{r},\mathbf{s}}$ the error rate of the Bayes optimal predictor is 0.

4 Proof of Theorem 2.1

Because $\mathbf{r}$ and $\mathbf{s}$ "give everything away," a discriminative algorithm can succeed easily.

Lemma 4.1 *Problem $\mathcal{P}$ can be solved using at most $2 \log \frac{2}{\delta}$ examples.*

Proof. If $(\mathbf{r}, 1)$ and $(\mathbf{s}, 2)$ are both in the training data, a discriminative algorithm can determine the classifications of all remaining elements of the domain, as the correct classification of $(\mathbf{e}_i, 3)$ is $(-1)^{r_i \oplus s_i}$. Consider an algorithm that does this, and behaves arbitrarily if it has not seen both $(\mathbf{r}, 1)$ and $(\mathbf{s}, 2)$. The probability that at least one of $(\mathbf{r}, 1)$, $(\mathbf{s}, 2)$ is not present in a sample of m examples is at most $2 \times (2/3)^m$. Solving for m completes the proof.

Now we show that generative algorithms must fail. Our argument uses the probabilistic method as in [3]. We will show that any algorithm A must perform poorly on a randomly chosen source. This implies that there is a source on which A performs poorly.

Lemma 4.2 *Fix a generative learning algorithm A. For any $n \geq 2$, if*

- *$\mathbf{r}$ and $\mathbf{s}$ are chosen uniformly at random from $\{0, 1\}^n$,*
- *$m \leq n$ examples are chosen according to $P_{\mathbf{r},\mathbf{s}}$,*
- *the positive and negative examples are separately passed to A as in the definition of the generative PAB learning framework, and*
- *the invocations of A output h_+ and h_-,*

then with probability at least $1/40$ (over the random draw of $\mathbf{r}, \mathbf{s}$ and the random draw of the m-element sample from $P_{\mathbf{r},\mathbf{s}}$,) the source $P_{\mathbf{r},\mathbf{s}}$ puts weight at least $1/40$ on pairs (x, y) for which $\operatorname{sgn}(h_+(x) - h_-(x)) \neq y$.

Proof. Suppose that $\mathbf{r}, \mathbf{s} \in \{0,1\}^n$ are chosen randomly, and that

$$(x_1, y_1), ..., (x_m, y_m), (x, y)$$

are chosen independently at random according to $P_{\mathbf{r},\mathbf{s}}$.

The proof proceeds by first lower bounding the conditional probability that the hypotheses h_+, h_- output by A collaborate to predict the class y of x incorrectly, given that a particular event E occurs. The proof is completed by lower bounding the probability of E.

Event E is defined as follows: a draw of $\mathbf{r}, \mathbf{s}, (x_1, y_1), ..., (x_m, y_m), (x, y)$ satisfies event E if there is some i such that $x = (\mathbf{e}_i, 3)$ and none of $x_1, \ldots, x_m$ is $(\mathbf{e}_i, 3)$.

Suppose that event E occurs. Let i be the value in $\{1, ..., n\}$ such that $x = (\mathbf{e}_i, 3)$ and none of $x_1, \ldots, x_m$ is $(\mathbf{e}_i, 3)$. Consider any fixed setting of values for all components of $\mathbf{r}$ except for r_i, and all components of $\mathbf{s}$ except s_i. Similarly consider any fixed setting of values for $(x_1, y_1), ..., (x_m, y_m)$ such that $x_t \neq (\mathbf{e}_i, 3)$ for all $t \in \{1, ..., m\}$. (Note that if x_j is set to $(\mathbf{r}, 1)$ or $(\mathbf{s}, 2)$ then the i-th component is not yet fixed.) Let us denote this more specific event by E'.

Now consider the probability distribution obtained by conditioning on E'; note that the only remaining randomness is the choice of $r_i, s_i \in \{0,1\}$. According to this distribution, each of the four possible pairs of values for r_i, s_i are equally likely, and, in each case, the corresponding class designation for x is $(-1)^{r_i \oplus s_i}$. However, after conditioning on E', the scoring function h_+ is completely determined by the value of r_i (recall that when algorithm A constructs h_+ it may well receive the example $(\mathbf{r}, 1)$ but it does not receive the example $(\mathbf{s}, 2)$). This implies that the value $h_+(x)$ is completely determined by the value of the bit r_i. Similarly, the value $h_-(x)$ is completely determined by the value of the bit s_i. Consequently, $\mathrm{sgn}(h_+(x) - h_-(x))$ is a function of $(r_i, s_i) \in \{0,1\}^2$; further, since r_i and s_i only affect $h_+(x)$ and $h_-(x)$ respectively, $\mathrm{sgn}(h_+(x) - h_-(x))$ is in fact a *linear threshold function* of the variables r_i and s_i. It is well known that a linear threshold function cannot compute a parity over two boolean variables. Therefore, given event E', there must be at least one combination of values for r_i and s_i such that A predicts $(-1)^{r_i \oplus s_i}$ incorrectly. Since all combinations of values for r_i and s_i are equally likely, the conditional probability that A predicts x incorrectly given E' is at least $1/4$. It follows that the conditional probability that A predicts x incorrectly given event E is at least $1/4$.

It is straightforward to lower bound (in fact, exactly compute) the probability of event E. Since all pairs (x_i, y_i) are drawn independently, the probability of event E is easily seen to be

$$\frac{1}{3} \times \left(1 - \frac{1}{3n}\right)^m.$$

If $m \leq n$, this probability is at least

$$\frac{1}{3} \times \left(1 - \frac{1}{3n}\right)^n \geq \frac{1}{5}.$$

Thus, the overall probability that $\mathrm{sgn}(h_+(x) - h_-(x)) \neq y$ is at least 1/20. This easily yields the lemma.

From this we can easily establish the following which proves Theorem 2.1:

Lemma 4.3 *$\mathcal{P}$ is not learnable in the generative PAB model.*

Proof. Fix algorithm A. Suppose, in Lemma 4.2, we first choose $\mathbf{r}$ and $\mathbf{s}$ from $\{0,1\}^n$, and then choose the random examples from $P_{\mathbf{r},\mathbf{s}}$. Then the expectation, over $\mathbf{r}$ and $\mathbf{s}$, of

$$\Pr_{(x_1,y_1),\ldots,(x_m,y_m)}[P_{\mathbf{r},\mathbf{s}} \text{ puts weight at least } 1/40 \text{ on } (x,y) \text{ such that } \mathrm{sgn}(h_+(x) - h_-(x)) \neq y]$$

is at least 1/40. This means that there is a particular choice of $\mathbf{r}$ and $\mathbf{s}$ for which

$$\Pr_{(x_1,y_1),\ldots,(x_m,y_m)}[P_{\mathbf{r},\mathbf{s}} \text{ puts weight at least } 1/40 \text{ on } (x,y) \text{ such that } \mathrm{sgn}(h_+(x) - h_-(x)) \neq y] > 1/40.$$

Thus, at least n examples are needed to learn $P_{\mathbf{r},\mathbf{s}}$ whenever ϵ and δ are each at most 1/40. By fixing ϵ and δ at 1/40, and choosing n arbitrarily large, we can see that there is no fixed sample size, as a function of ϵ and δ, that suffices to PAB-learn arbitrary members of $\mathcal{P}$ to accuracy ϵ with probability $1 - \delta$.

Note that the proof does not depend on the fact that the same algorithm was applied to the positive and negative examples. Furthermore, a straightforward extension of the proof generalizes Lemma 4.2 to generative learning algorithms that are probabilistic.[2] Thus, we get the following for free.

Theorem 4.4 *Suppose the generative PAB learning model is relaxed so that separate (and possibly probabilistic) algorithms can be applied to the positive and negative examples. Then it remains true that there is a learning problem that can be solved in the standard PAB model, but not in the generative PAB model.*

[2] Include a fixed choice of the learner's random bits into the restricted event E'.

5 Conclusions and Future Work

We presented a learning problem in the Probably Approximately Bayes framework which has the property that a discriminative algorithm can solve the problem, but no generative algorithm can solve the problem. One drawback of our construction is that it is arguably somewhat artificial and contrived. While it nevertheless serves to separate the two learning models, it would be interesting to come up with a more natural construction that also successfully separates the models.

A goal for future work is to extend our separation to the Probably Approximately Correct (PAC) learning model. Another goal is to explore computational separations between discriminative and generative learning.

References

[1] Svetlana Anoulova, Paul Fischer, Stefan Pölt, and Hans Ulrich Simon. Probably almost bayes decisions. *Information and Computation*, 129(1):63–71, 1996.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd ed.)*. Wiley, 2000.

[3] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. G. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–251, 1989.

[4] P. Goldberg. When Can Two Unsupervised Learners Achieve PAC Separation? In *Proceedings of the 14th Annual COLT*, pages 303–319, 2001.

[5] P. Goldberg. Some Discriminant-Based PAC Algorithms. *Journal of Machine Learning Research*, 7:283–306, 2006.

[6] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in NIPS 11*, pages 487–493. Morgan Kaufmann, 1998.

[7] T. Jebara. *Machine learning: discriminative and generative*. Kluwer, 2003.

[8] P. M. Long and R. A. Servedio. Discriminative learning can succeed where generative learning fails. In *Proc. 19th Conference on Computational Learning Theory*, pages 319–334, 2006.

[9] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *NIPS*, 2001.

[10] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. *NIPS*, 2004.

[11] L. G. Valiant. A theory of the learnable. In *Proc. 16th Annual ACM Symposium on Theory of Computing (STOC)*, pages 436–445. ACM Press, 1984.

[12] V. Vapnik. *Estimations of dependences based on statistical data*. Springer, 1982.