Classification with Hybrid Generative/Discriminative Models

Rajat Raina, Yirong Shen, Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, CA 94305

Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract

Although discriminatively-trained classifiers are usually more accurate when labeled training data is abundant, previous work has shown that when training data is limited, generative classifiers can out-perform them. This paper describes a hybrid model in which a high-dimensional subset of the parameters are trained to maximize generative likelihood, and another, small, subset of parameters are trained to maximize conditional likelihood. We give a sample complexity bound showing that in order to fit the discriminative parameters well, the number of training examples required depends only logarithmically on the number of feature occurrences and feature set size. Experimental results show that hybrid models can provide lower test error than either their purely generative or purely discriminative counterparts, and can produce better accuracy/coverage curves than naive Bayes or logistic regression. We also discuss several advantages of hybrid models, and advocate further work in this area.

1 Introduction

Generative classifiers learn a model of the joint probability, $p(x, y)$, of the inputs x and the label y, and make their predictions by using Bayes rules to calculate $p(y|x)$, and then picking the most likely label y. In contrast, discriminative classifiers model the posterior $p(y|x)$ directly. It is widely believed that for many application domains, discriminative classifiers often achieve higher test set accuracy than generative classifiers (e.g., [6, 4, 14]). Nonetheless, generative classifiers also have several advantages, among them straightforward EM methods for handling missing data, and better performance when training set sizes are small. Specifically, it has been shown that a simple generative classifier (naive Bayes) outperforms its conditionally-trained, discriminative counterpart (logistic regression) when the amount of available labeled training data is small [11].

In an effort to obtain the best of both worlds, this paper explores a class of hybrid models for supervised learning that are partly generative and partly discriminative. In these models, a large subset of the parameters are trained to maximize the generative, joint probability of the inputs and outputs of the supervised learning task; another, much smaller, subset of the parameters are discriminatively trained to maximize the conditional probability of the outputs given the inputs.

Following Ng and Jordan [11], and motivated by an application in text classification as well as the desire to begin by exploring a simple, pure form of a hybrid algorithm, we describe and give results with logistic regression, naive Bayes, and a hybrid based on both. We also introduce two natural by-products of the hybrid model. First, a scheme for allowing different partitions of the variables to contribute more or less strongly to the classification decision—for an email classification example, modeling the text in the subject line and message body separately, with learned weights for relative contribution. Second, a method for improving accuracy/coverage curves of models that make incorrect independence assumptions, such as naive Bayes.

We also prove a sample complexity result in which the number of examples needed to fit the discriminative parameters increases only as the logarithm of the vocabulary size and document length. In experimental results we show that the hybrid model achieves significantly more accurate classification than either the purely generative or the purely discriminative approaches. We demonstrate that the hybrid model produces class posterior probabilities that better reflect empirical error rates, and as a result, improved accuracy/coverage curves.

2 The Model

We begin by briefly reviewing the multinomial naive Bayes classifier applied to text categorization [10], and then describe our hybrid model and its relation to logistic regression.

Let $\mathcal{Y} = \{0, 1\}$ be the set of possible labels for a document classification task, and let $\mathcal{W} = \{w_1, w_2, \ldots, w_{|\mathcal{W}|}\}$ be a dictionary of words. A document of N words is represented by the vector $X = (X_1, X_2, \ldots, X_N)$ of length N. The ith word in the document is $X_i \in \mathcal{W}$. Note that N can vary for different documents. The multinomial naive Bayes model assumes that the label Y is chosen from some prior distribution $P(Y = \cdot)$, the length N is drawn from some distribution $P(N = \cdot)$ independently of the label, and each word X_i is drawn independently from some distribution $P(W = \cdot|Y)$ over the dictionary. Thus, we have[1]:

$$P(X = x, Y = y) = P(Y = y)P(N = n) \prod_{i=1}^{n} P(W = x_i|Y = y) \qquad (1)$$

Since the length n of the document does not depend on the label and therefore does not play a significant role, we leave it out of our subsequent derivations.

The parameters in the naive Bayes model are $\hat{P}(Y)$ and $\hat{P}(W|Y)$ (our estimates of $P(Y)$ and $P(W|Y)$). They are set to maximize the joint (penalized) log-likelihood of the x and y pairs in a labeled training set, $M = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$. Let $n^{(i)}$ be the length of document $x^{(i)}$. Specifically, for any $k \in \{0, 1\}$, we have:

$$\hat{P}(Y = k) = \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = k\} \qquad (2)$$

$$\hat{P}(W = w_i|Y = k) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n^{(i)}} 1\{x_j^{(i)} = w_i,\ y^{(i)} = k\} + 1}{\sum_{i=1}^{m} n^{(i)} 1\{y^{(i)} = k\} + |\mathcal{W}|}, \qquad (3)$$

where $1\{\cdot\}$ is the indicator function ($1\{\text{True}\} = 1$,$1\{\text{False}\}=0$), and we have applied Laplace (add-one) smoothing in obtaining the estimates of the word probabilities. Using Bayes rule, we obtain the estimated class posterior probabilities for a new document as:

$$\hat{P}(Y = 1|X = x) = \frac{\hat{P}(X=x|Y=1)\hat{P}(Y=1)}{\sum_{y \in \mathcal{Y}} \hat{P}(X=x|Y=y)\hat{P}(Y=y)}$$

where

$$\hat{P}(X = x|Y = y) = \prod_{i=1}^{n} \hat{P}(W = x_i|Y = y). \qquad (4)$$

The predicted class for the new document is then simply $\arg\max_{y \in \mathcal{Y}} \hat{P}(Y = y|X = x)$.

In many text classification applications, the documents involved consist of several disjoint regions that may have different dependencies with the document label. For example, a

[1]We adopt the notational convention that upper-case is used to denote random variables, and lower-case is used to denote particular values taken by the random variables.

USENET news posting can be considered to consist of a subject region and a body region.[2] Because of the strong assumptions in its model, naive Bayes treats the words in the different regions of a document in exactly the same way, ignoring the fact that perhaps words in a particular region (such as words in the subject) might be more "important." Further, it also tends to allow the words in the longer region to dominate. (Explained below.)

In the sequel, we assume that every input document X can be naturally divided into R regions $X^1, X^2, \ldots, X^R$. Note that R can be just 1. The regions are of variable lengths $N_1, N_2, \ldots, N_R$. For the sake of conciseness and clarity, in the following discussion we will focus on the case of $R = 2$ regions, the generalization offering no difficulties. Thus, the document probability in Equation (4) is now replaced with:

$$\begin{aligned} \hat{P}(X = x|Y = y) &= \hat{P}(X^1 = x^1|Y = y)\hat{P}(X^2 = x^2|Y = y) && (5) \\ &= \prod_{i=1}^{n_1} \hat{P}(W = x_i^1|Y = y) \prod_{i=1}^{n_2} \hat{P}(W = x_i^2|Y = y) && (6) \end{aligned}$$

Here, x_i^j denotes the ith word in the jth region. It is straightforward to see that naive Bayes will predict $y = 1$ if

$$\begin{aligned} &\sum_{i=1}^{n_1} \log \hat{P}(W = x_i^1|Y = 1) + \sum_{i=1}^{n_2} \log \hat{P}(W = x_i^2|Y = 1) + \log \hat{P}(Y = 1) \geq \\ &\quad \sum_{i=1}^{n_1} \log \hat{P}(W = x_i^1|Y = 0) + \sum_{i=1}^{n_2} \log \hat{P}(W = x_i^2|Y = 0) + \log \hat{P}(Y = 0) \end{aligned}$$

and predict $y = 0$ otherwise. In an email or USENET news classification problem, if the first region is the subject, and the second region is the message body, then $n_2 \gg n_1$, since message bodies are usually much longer than subjects. Thus, in the equation above, the message body contributes to many more terms in both the left and right sides of the summation, and the result of the "$\geq$" test will be largely determined by the message body (with the message subject essentially ignored or otherwise having very little effect).

Given the importance and informativeness of message subjects, this suggests that we might obtain better performance than the basic naive Bayes classifier by considering a modified algorithm that assigns different "weights" to different regions and normalizes for the length of the regions. Specifically, consider making a prediction using instead the modified inequality test:

$$\begin{aligned} &\frac{\theta_1}{n_1} \sum_{i=1}^{n_1} \log \hat{P}(W = x_i^1|Y = 1) + \frac{\theta_2}{n_2} \sum_{i=1}^{n_2} \log \hat{P}(W = x_i^2|Y = 1) + \log \hat{P}(Y = 1) \geq \\ &\quad \frac{\theta_1}{n_1} \sum_{i=1}^{n_1} \log \hat{P}(W = x_i^1|Y = 0) + \frac{\theta_2}{n_2} \sum_{i=1}^{n_2} \log \hat{P}(W = x_i^2|Y = 0) + \log \hat{P}(Y = 0) \end{aligned}$$

Here, the vector of parameters $\theta = (\theta_1, \theta_2)$ controls the relative "weighting" between the message subjects and bodies, and will be fit discriminatively. Specifically, we will model the class posteriors, which we denote by $\hat{P}_\theta$ to make explicit the dependence on θ, as[3]

$$\hat{P}_\theta(y|x) = \frac{\hat{P}(y)\hat{P}(x^1|y)^{\frac{\theta_1}{n_1}}\hat{P}(x^2|y)^{\frac{\theta_2}{n_2}}}{\hat{P}(Y{=}0)\hat{P}(x^1|Y{=}0)^{\frac{\theta_1}{n_1}}\hat{P}(x^2|Y{=}0)^{\frac{\theta_2}{n_2}} + \hat{P}(Y{=}1)\hat{P}(x^1|Y{=}1)^{\frac{\theta_1}{n_1}}\hat{P}(x^2|Y{=}1)^{\frac{\theta_2}{n_2}}} \quad (7)$$

We had previously motivated our model as taking into account different weights for different parts of the document. A second reason for using this model is that the independence assumption for naive Bayes is too strong. Specifically, with a document of length n, the classifier "assumes" that it has n completely independent pieces of evidence supporting its conclusion about the document's label. Putting n_r in the denominator of the exponent as a normalization factor can be viewed as a way of counteracting the overly strong independence assumptions.[4]

[2]Other possible text classification examples include: Emails consisting of subject and body; technical papers consisting of title, abstract, and body; web pages consisting of title, headings, and body.

[3]When it is clear from the context, we will sometimes replace $P(X = x|Y = y)$, $P(Y = y|X = x)$, $P(W = x_i|Y = y)$, etc. by their shorthand forms $P(x|y)$, $P(y|x)$, $P(x_i|y)$, etc.

[4]θ_r can also be viewed as an "effective region length" parameter, where we assume that region r of the document can be treated as only θ_r independent pieces of observation. For example, note that if each region r of the document has θ_r words exactly, then this model reduces to naive Bayes.

After some simple manipulations, we obtain the following expression for $\hat{P}_\theta(Y=1|x)$:

$$\hat{P}_\theta(Y=1|x) = \frac{1}{1+\exp(-a-\theta_1 b_1 - \ldots - \theta_R b_R)} \tag{8}$$

where $a = \log \frac{\hat{P}(Y=1)}{\hat{P}(Y=0)}$ and $b_r = \frac{1}{n_r}(\log \frac{\hat{P}(x^r|Y=1)}{\hat{P}(x^r|Y=0)})$. With this formulation for $\hat{P}_\theta(y|x)$, we see that it is very similar to the form of the class posteriors used by logistic regression, the only difference being that in this case a is a constant calculated from the estimated class priors. To make the parallel to logistic regression complete, we define $b_0 = 1$, redefine θ as $\theta = (\theta_0, \theta_1, \theta_2)$, and define a new class posterior

$$\hat{P}_\theta(Y=1|x) = \frac{1}{1+\exp(-\theta^T b)} \tag{9}$$

Throughout the derivation, we had assumed that the parameters $\hat{P}(x|y)$ were fit generatively as in Equation 3 (and b_r is in turn derived from these parameters as described previously above). It therefore remains only to specify how θ is chosen. One method would be to pick θ by maximizing the conditional log-likelihood of the training set $M = \{x^{(i)}, y^{(i)}\}_{i=1}^m$:

$$\theta = \arg\max_{\theta'} \sum_{i=1}^m \log \hat{P}_{\theta'}(y^{(i)}|x^{(i)}) \tag{10}$$

However, the word generation probabilities that were used to calculate b were also trained from M. This procedure therefore fits the parameters θ to the training data, using "features" b that were also fit to the data. This leads to a biased estimator. Specifically, since what we care about is the generalization performance of the algorithm, a better method is to pick θ to maximize the log-likelihood of data that wasn't used to calculate the "features" b, because when we see a test example, we will not have had the luxury of incorporating information from the test example into the b's (cf. [15, 12]). This leads to the following "leave-one-out" strategy of picking θ:

$$\theta = \arg\max_{\theta'} \sum_{i=1}^m \log \hat{P}_{\theta',-i}(y^{(i)}|x^{(i)}), \tag{11}$$

where $\hat{P}_{\hat{\theta},-i}(y^{(i)}|x^{(i)})$ is as given in Equation (9), except that each b_r is computed from word generation probabilities that were estimated with the ith training example of the training set held out. We note that optimizing this objective to find θ is still exactly the same optimization problem as in logistic regresssion, and hence is convex and can be solved efficiently.

The predicted label for a new document under this scheme is just $\arg\max_{y\in\mathcal{Y}} \hat{P}_\theta(y|x)$. We call this scheme the *normalized hybrid* algorithm. For the sake of comparison, we will also consider a related scheme in which the exponents in Equation (7) are not normalized by n_r. In other words, we replace θ_r/n_r there by just θ_r. We refer to this latter scheme as the *unnormalized hybrid* algorithm.

3 Experimental Results

We now describe the results of experiments testing the effectiveness of our methods. All experiments were run using pairs of newsgroups from the 20newsgroups dataset [8] of USENET news postings. When parsing this data, we skipped everything in the USENET headers except the subject line; numbers and email addresses were replaced by special tokens NUMBER and EMAILADDR; and tokens were formed after stemming.

In each experiment, we compare the performance of the basic naive Bayes algorithm against the normalized hybrid algorithm and logistic regression with Gaussian priors on the parameters [6]. All results reported in this section are averages over 10 random train-test splits.

Figures 1 plots the average test error of the various algorithms versus training set size for classifying pairs of newsgroups. We find that in every experiment, for the training set sizes considered, the normalized hybrid algorithm with $R = 2$ has test error that is either the lowest or very near the lowest among all the algorithms. In particular, it almost always

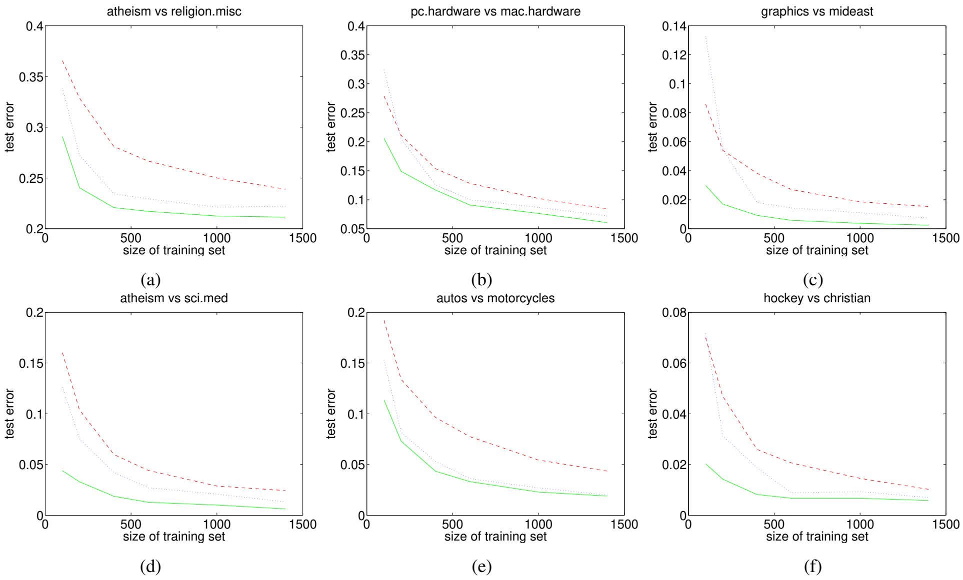

Figure 1: Plots of test error vs training size for several different pairs of newsgroups. Red dashed line is logistic regression; blue dotted line is standard naive Bayes; green solid line is hybrid algorithm. (Colors where available.) (If more training data were available, logistic regression would presumably out-perform naive Bayes; cf. [11, 6].)

outperforms the basic naive Bayes algorithm. The difference in performance is especially dramatic for small training sets.

Although these results are not shown here, the hybrid algorithm with $R = 2$ (breaking the document into two regions) outperforms $R = 1$. Further, the normalized version of the hybrid algorithm generally outperforms the unnormalized version.

4 Theoretical Results

In this section, we give a distribution free uniform convergence bound for our algorithm. Classical learning theory and VC theory indicate that, given a discriminative model with a small number of parameters, only a small amount of training data should be required to fit the parameters "well" [14]. In our model, a large number of parameters $\hat{P}$ are fit generatively, but only a small number (the θ's) are fit discriminatively. We would like to show that only a small training set is required to fit the discriminative parameters θ.[5] However, standard uniform convergence results do not apply to our problem, because the "features" b_i given to the discriminative logistic regression component also depend on the training set. Further, the θ_i's are fit using the leave-one-out training procedure, so that every pair of training examples is actually dependent.

For our analysis, we assume the training set of size m is drawn *i.i.d.*from some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Although not necessary, for simplicity we assume that each document has the same total number of words $n = \sum_{i=1}^{R} n_i$, though the lengths of the individual regions may vary. (It also suffices to have an upper- and a lower-bound on document length.) Finally, we also assume that each word occurs at most C_{max} times in a single

[5]For a result showing that naive Bayes' generatively fit parameters (albeit one using a different event model) converge to their population (asymptotic) values after a number of training examples that depends logarithmically on the size of the number of features, also see [11].

document, and that the distribution $\mathcal{D}$ from which training examples are drawn satisfies $\rho_{min} \leq P(y=1) \leq 1-\rho_{min}$, for some fixed $\rho_{min} > 0$.

Note that we do *not* assume that the "naive Bayes assumption" (that words are conditionally independent given the class label) holds. Specifically, even when the naive Bayes assumption does not hold, the naive Bayes *algorithm* (as well as our hybrid algorithm) can still be applied, and our results apply to this setting.

Given a set M of m training examples, for a particular setting of the parameter θ, the expected log likelihood of a randomly drawn test example is:

$$\varepsilon^M(\theta) \quad = \quad E_{(x,y)\sim\mathcal{D}} \log \hat{P}_\theta(y|x) \tag{12}$$

where $\hat{P}_\theta$ is the probability model trained on M as described in the previous section, using parameters $\hat{P}$ fit to the entire training set. Our algorithm uses a leave-one-out estimate of the true log likelihood, which we call the leave-one-out log likelihood:

$$\hat{\varepsilon}^M_{-1}(\theta) \quad = \quad \frac{1}{m}\sum_{i=1}^m \log \hat{P}_{\theta,-i}(y^{(i)}|x^{(i)}) \tag{13}$$

where $\hat{P}_{\theta,-i}$ represents the probability model trained with the ith example left out.

We would like to choose θ to maximize ε^M, but we do not know ε^M. Now, it is well-known that if we have some estimate $\hat{\varepsilon}$ of a generalization error measure ε, and if $|\hat{\varepsilon}(\theta)-\varepsilon(\theta)| \leq \epsilon$ for all θ, then optimizing $\hat{\varepsilon}$ will result in a value for θ that comes within 2ϵ of the best possible value for $\varepsilon(\theta)$ [14]. Thus, in order to show that optimizing $\hat{\varepsilon}^M_{-1}$ is a good "proxy" for optimizing ε^M, we only need to show that $\hat{\varepsilon}^M_{-1}(\theta)$ is uniformly close to $\varepsilon^M(\theta)$. We have:

Theorem 1 *Under the previous set of assumptions, in order to ensure that with probability at least $1-\delta$, we have $|\varepsilon^M(\theta)-\hat{\varepsilon}^M_{-1}(\theta)| < \epsilon$ for all parameters θ such that $||\theta||_\infty \leq \eta$, it suffices that $m = O(\mathrm{poly}(1/\delta, 1/\epsilon, \log n, \log|\mathcal{W}|, R, \eta)^R)$.*

The full proof of this result is fairly lengthy, and is deferred to the Appendix [13]. full version of this paper [13] (available online). From the theorem, the number of training examples m required to fit the θ parameters depends only on the logarithms of the document length n and the vocabulary size $|\mathcal{W}|$. In our bound, there is an exponential dependence on R; however, from our experience, R does not need to be too large for significantly improved performance. In fact, our experimental results demonstrate good performance for $R=2$.

5 Calibration Curves

We now consider a second application of these ideas, to a text classification setting where the data is not naturally split into different regions (equivalently, where $R=1$). In this setting we cannot use the "reweighting" power of the hybrid algorithm to reduce classification error. But, we will see that, by giving better class posteriors, our method still gives improved performance as measured on accuracy/coverage curves.

An accuracy/coverage curve shows the accuracy (fraction correct) of a classifier if it is asked only to provide $x\%$ coverage—that is, if it is asked only to label the $x\%$ of the test data on which it is most confident. Accuracy/coverage curves towards the upper-right of the graph mean high accuracy even when the coverage is high, and therefore good performance. Accuracy value at coverage 100% is just the normal classification error. In settings where both human and computer label documents, accuracy/coverage curves play a central role in determining how much data has to be labeled by humans. They are also indicative of the quality of a classifier's class posteriors, because a classifier with better class posteriors would be able to better judge which $x\%$ of the test data it should be most confident on, and achieve higher accuracy when it chooses to label that $x\%$ of the data.

Figure 2 shows accuracy/coverage curves for classifying several pairs of newsgroups from the 20newsgroups dataset. Each plot is obtained by averaging the results of ten 50%/50% random train/test splits. The normalized hybrid algorithm ($R=1$) does significantly better

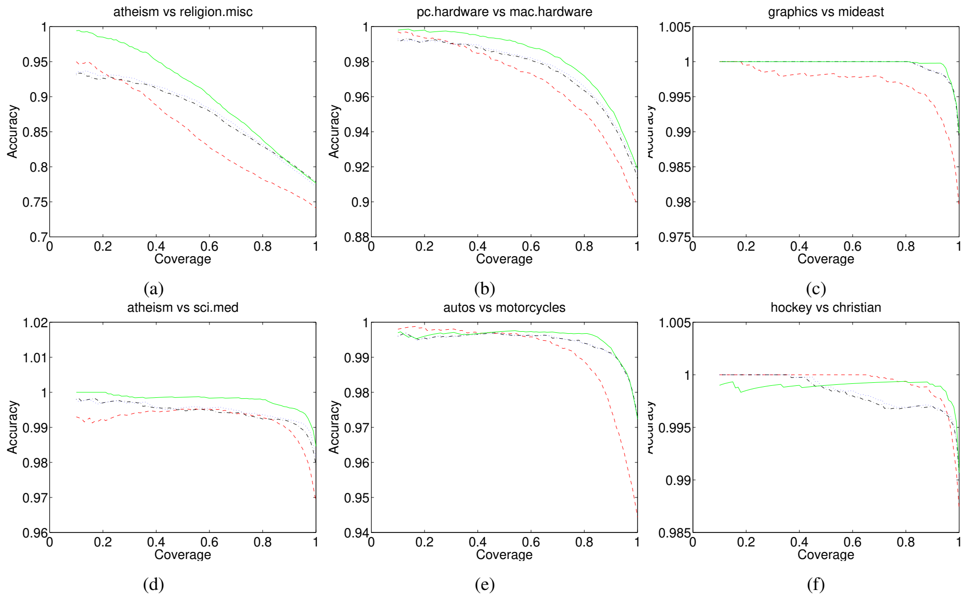

Figure 2: Accuracy/Coverage curves for different pairs of newsgroups. Green solid line is our normalized hybrid algorithm; black dash-dot line is naive Bayes; blue dotted line is unnormalized hyrid, and red dashed line is logistic regression. (Colors where available.)

than naive Bayes, and has accuracy/coverage curves that are higher almost everywhere. For example, in Figure 2a, the normalized hybrid algorithm with $R = 1$ has a coverage of over 40% at 95% accuracy, while naive Bayes' coverage is 0 for the same accuracy. Also, the unnormalized algorithm has performance about the same as naive Bayes. Even in examples where the various algorithms have comparable overall test error, the normalized hybrid algorithm has significantly better accuracy/coverage.

6 Discussion and Related Work

This paper has described a hybrid generative/discriminative model, and presented experimental results showing that a simple hybrid model can perform better than either its purely generative or discriminative counterpart. Furthermore, we showed that in order to fit the parameters θ of the model, only a small number of training examples is required.

There have been a number of previous efforts to modify naive Bayes for the purpose of obtaining more empirically accurate posterior probabilities. Lewis and Gale [9] use logistic regression to recalibrate naive Bayes posteriors in an active learning task. Their approach is similar to the lower-performing *unnormalized* version of our algorithm, with only one region. Bennett [1] studies the problem of using asymmetric parametric models to obtain high quality probability estimates from the scores outputted by text classifiers such as naive Bayes. Zadrozny and Elkan [16] describe a simple non-parametric method for calibrating naive Bayes probability estimates. While these methods can obtain good class posteriors, we note that in order to obtain better accuracy/coverage, it is not sufficient to take naive Bayes' output $p(y|x)$ and find a monotone mapping from that to a set of hopefully better class posteriors (e.g., [16]). Specifically, in order to obtain better accuracy/coverage, it is also important to *rearrange* the confidence orderings that naive Bayes gives to documents (which our method does because of the normalization).

Jaakkola and Haussler [3] describe a scheme in which the kernel for a discriminative classifier is extracted from a generative model. Perhaps the closest to our work, however, is the commonly-used, simple "reweighting" of the language model and acoustic model in speech recognition systems (e.g., [5]). Each of the two models is trained generatively; then a single weight parameter is set using hold-out cross-validation.

In related work, there are also a number of theoretical results on the quality of leave-one-out estimates of generalization error. Some examples include [7, 2]. (See [7] for a brief survey.) Those results tend to be for specialized models or have strong assumptions on the model, and to our knowledge do not apply to our setting, in which we are also trying to fit the parameters θ.

In closing, we have presented one hybrid generative/discriminative algorithm that appears to do well on a number of problems. We suggest that future research in this area is poised to bear much fruit. Some possible future work includes: automatically determining which parameters to train generatively and which discriminatively; training methods for more complex models with latent variables, that require EM to estimate both sets of parameters; methods for taking advantage of the hybrid nature of these models to better incorporate domain knowledge; handling missing data; and support for semi-supervised learning.

Acknowledgments

Yirong Shen is supported by a NSF graduate fellowship. This work was supported in part by the Center for Intelligent Information Retrieval; and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010; and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

References

[1] Paul N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In *Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, 2003.

[2] Luc P. Devroye and T. J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Informatoin Theory*, 5, September 1979.

[3] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, 1998.

[4] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the cem algorithm. In *Advances in Neural Information Processing Systems 11*, 1998.

[5] D. Jurafsky and J. Martin. *Speech and language processing*. Prentice Hall, 2000.

[6] John Lafferty Kamal Nigam and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.

[7] Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Computational Learing Theory*, 1997.

[8] Ken Lang. Newsweeder: learning to filter netnews. In *Proceedings of the Ninth European Conference on Machine Learning*, 1997.

[9] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, 1994.

[10] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[11] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In *NIPS 14*, 2001.

[12] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[13] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. http://www.cs.stanford.edu/~rajatr/nips03.ps, 2003.

[14] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[15] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.

[16] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML '01*, 2001.

7 Appendix

Below we present the complete proof of our theoretical results. Lemmas 2 and 3 characterize the change in parameters as more examples are added to the training set. Corollary 4 uses these to derive a result that will be very useful in our analysis.

Theorem 5 presents the uniform convergence result for fixed parameters θ, and forms the core of our overall proof. Finally, the general uniform convergence bounds and the sample complexity bounds are presented.

Lemma 2 *Let $\hat{\phi}^M_{j|y=k} \equiv \log \hat{P}^M(w_j|y=k)$ be the logarithm of the probability estimated by Naive Bayes on the training set M. Given another set of examples S drawn from the same distribution as M, suppose the parameters are first estimated using M, and again using $M \cup S$. Then, with $m = |M| > 1$, $s = |S|$, for any $0 < c < 1/2$, and for constants $T_{1\ldots4}$ (possibly depending on c),*

$$|\hat{\phi}^M_{j|y=k} - \hat{\phi}^{M\cup S}_{j|y=k}| \quad < \quad T_1 \frac{s}{m^{1-c}} \quad \forall 1 \le j \le |\mathcal{W}|, k \in \{0,1\} \tag{14}$$

holds with probability at least $(1 - \max(2\exp(-T_2 m^{1-2c}), 2\exp(-T_3 s)))$
Also, a weaker but deterministic bound exists:

$$|\hat{\phi}^M_{j|y=k} - \hat{\phi}^{M\cup S}_{j|y=k}| \quad < \quad \log(T_4 n s^2) \quad \forall 1 \le j \le |\mathcal{W}|, k \in \{0,1\} \tag{15}$$

Proof: Let $C_{j,A}$ = Total number of occurrences of w_j in documents from set A.
Define sets M_k and S_k to contain all documents from class k in sets M and S respectively, and let $m_k = |M_k|$, $s_k = |S_k|$. Then, by definition:

$$\phi^M_{j|y=k} \quad = \quad \log\left(\frac{C_{j,M_k}+1}{nm_k + |\mathcal{W}|}\right)$$

Putting this into our expression, rewriting and using the triangle inequality:

$$\begin{aligned}\Delta \; = \; |\hat{\phi}^M_{j|y=k} - \hat{\phi}^{M\cup S}_{j|y=k}| \quad &= \quad |\log\left(\frac{C_{j,M_k}+1}{nm_k+|\mathcal{W}|}\right) - \log\left(\frac{C_{j,M_k}+C_{j,S_k}+1}{n(m_k+s_k)+|\mathcal{W}|}\right)| \\ &= \quad |\log\left(1+\frac{ns_k}{nm_k+|\mathcal{W}|}\right) - \log\left(1+\frac{C_{j,S_k}}{C_{j,M_k}+1}\right)| \\ &\le \quad \log\left(1+\frac{ns_k}{nm_k+|\mathcal{W}|}\right) + \log\left(1+\frac{C_{j,S_k}}{C_{j,M_k}+1}\right)\end{aligned}$$

The deterministic bound follows directly from here as $s_k < s$, $|\mathcal{W}| \ge 1$ and $C_{j,S_k} \le C_{max}s$. Using the inequality $\log(1+x) \le x$ for $x \ge 0$:

$$\begin{aligned}\Delta \quad &\le \quad \frac{ns_k}{nm_k+|\mathcal{W}|} + \frac{C_{j,S_k}}{C_{j,M_k}+1} \\ &< \quad \frac{s}{\rho_{min} m} + \frac{C_{j,S_k}}{C_{j,M_k}+1}\end{aligned}$$

where the last statement holds from the Chernoff bound with probability at least $(1 - 2\exp(-2\rho_{min}^2 m))$. It therefore suffices to prove a high probability bound on the relative number of occurrences of w_j in S_k and M_k. Consider two cases:

Case I: $P(w_j$ occurs at least once | document class is $y = k) \geq 1/m^c$
Intuitively, w_j occurs frequently, and we expect C_{j,M_k} to be high. Since each document is independent, we use the Chernoff bound and the high probability bound on $m_k \geq \rho_{min} m$ (as earlier) to get:

$$P(\text{\# documents in } M_k \text{ containing } w_j \leq m_k/2m^c) \quad \leq \quad 2\exp(-T_2 m^{1-2c}) \;=\; \epsilon_1$$

for some constant T_2 (both probabilities can be combined with a union bound) . Thus, w.p. at least $1 - \epsilon_1$, $C_{j,M_k} \geq m_k/2m^c \geq \rho_{min} m/2m^c$ again from the Chernoff bound for m_k (and this can be incorporated into the probability without changing its form). Since we also have $C_{j,S_k} \leq C_{max} s$, and $c < 1/2$ we get the required probability bound in this case.

Case II: $P(w_j$ occurs at least once | document class is $y = k) < 1/m^c$
Intuitively, w_j is rare enough for C_{j,S_k} to be zero with high probability. Using the Chernoff bound and the high probability bound that $s_k \geq \rho_{min} s$:

$$P(\text{\# documents in } S_k \text{ containing } w_j \geq 1) \quad < \quad 2\exp(-T_3 s) \;=\; \epsilon_2$$

where T_3 depends on c (and both probabilities can be combined with a union bound). Thus, w.p. at least $1 - \epsilon_2$, $C_{j,S_k} = 0$ and the bound works in this case as well. □

Our next lemma is a bound on the variation in the log likelihood $\hat{\varepsilon}(\theta)$ of a particular example (x, y) as the Naive Bayes probabilities $\hat{\phi}_{j|y=k}$ are varied.

Lemma 3 *Suppose our algorithm is used with R parameters $\theta = \{\theta_1, \theta_2, .., \theta_R\}$, such that $||\theta||_\infty \leq \eta$, to classify a new example (x, y). Let $\hat{\varepsilon}_{(x,y)}(\theta) = \log P(y|x)$, where the Naive Bayes probability estimates $\hat{\phi}_{j|y=k} \equiv \log \hat{P}(w_j|y = k)$ are used. If a single $\hat{\phi}_{...|y=...}$ changes by at most D, then $\hat{\varepsilon}_{(x,y)}(\theta)$ changes by at most ηD. If each $\hat{\phi}_{...|y=...}$ changes by at most D, then $\hat{\varepsilon}_{(x,y)}(\theta)$ changes by at most $2\eta R D$.*

Proof: Without loss of generality, assume the new example y is in class 1. Using the earlier notation and $\psi^r_{i|y=k} = \log \hat{P}(x^r_i|y = k)$, we rewrite $\hat{\varepsilon}_{(x,y)}(\theta) = \log(1 + \exp(H))$ with:

$$H \quad = \quad -\frac{\theta_1}{n_1}\sum_{i=1}^{n_1}\psi^1_{i|y=1} - \frac{\theta_2}{n_2}\sum_{i=1}^{n_2}\psi^2_{i|y=1} - \ldots + \frac{\theta_1}{n_1}\sum_{i=1}^{n_1}\psi^1_{i|y=0} + \ldots$$

Since $d\hat{\varepsilon}_{(x,y)}(\theta)/dH = exp(H)/(1 + exp(H)) < 1$, the change in $\hat{\varepsilon}_{(x,y)}(\theta)$ will be less than the change in H.

Also, computing the absolute values of the derivative of H w.r.t each parameter $\psi^{\cdots}_{\ldots|y=\ldots}$ and multiplying by D gives a bound on the maximum change in H when each parameter changes by at most D. Since $n_1, n_2, \ldots \geq 1$, this gives the single parameter change bound. When all parameters change, the contributions can be added:

$$\begin{aligned} \text{Change in } H \quad &\leq \quad (\frac{|\theta_1|}{n_1}D)n_1 + (\frac{|\theta_2|}{n_2}D)n_2 + \ldots + (\frac{|\theta_1|}{n_1}D)n_1 + \ldots \\ &\leq \quad 2\eta R D \end{aligned}$$

Thus, the change in $\hat{\varepsilon}_{(x,y)}(\theta)$ is also bounded by this quantity. □

Given a set M of m training examples drawn from a joint distribution $\mathcal{D}$, the true log likelihood of the model is:

$$\varepsilon^M(\theta) \quad = \quad E_{(x,y)\sim\mathcal{D}} \log P^M(y^{(i)}|x^{(i)}) \tag{16}$$

where P^M is the probability model trained on M. The following Corollary to Lemmas 2 and 3 will be useful.

Corollary 4 *Suppose the parameters* $\hat{\phi}_{j|y=k} \equiv \log \hat{P}(w_j|y=k)$ *are estimated first from the set of examples A, and then from the set* $A \cup B$*, where* $|B| = m^t$ $(t < 1)$, $|A \cup B| = m$, $|A| = a = m - m^t$*, and A, B are drawn from the same distribution* $\mathcal{D}$*. For given parameters* θ *with* $||\theta||_\infty \leq \eta$*, let the log likelihood of an example* (x, y) *drawn from* $\mathcal{D}$ *be* $\varepsilon^{A}_{(x,y)}(\theta)$ *and* $\varepsilon^{A\cup B}_{(x,y)}(\theta)$ *respectively. Then:*

1. $\varepsilon^{A}_{(x,y)}(\theta)$ *and* $\varepsilon^{A\cup B}_{(x,y)}(\theta)$ *are close with high probability. Concretely, for* $0 < c < 1/2$, $0 < \tau < 1 - c - t$ *and m large enough* ($m > \Omega(\log(nR))$)
$$P(|\varepsilon^{A}_{(x,y)}(\theta) - \varepsilon^{A\cup B}_{(x,y)}(\theta)| > \frac{1}{m^\tau}) \quad < \quad T_5 R \frac{m^{t+\tau}}{a^{1-c}} \tag{17}$$
holds for some constant T_5.

2. *The maximum possible difference is bounded at least as:*
$$|\varepsilon^{A}_{(x,y)}(\theta) - \varepsilon^{A\cup B}_{(x,y)}(\theta)| \quad < \quad 2\eta R \log(T_4 n m^{2t}) \tag{18}$$
for some constant $T_4 > 0$.

Proof: 1. Let $\mathcal{F}$ denote the fraction of $\hat{\phi}$'s that change more than $D = T_1 m^t / a^{1-c}$ (the constant T_1 from Lemma 2 is used here) when estimated on A and $A \cup B$. Using Lemma 2, $E[\mathcal{F}] \leq \epsilon = \max(2\exp(-T_2 a^{1-2c}), 2\exp(-T_3 m^t))$. From the Markov inequality,
$$P(\mathcal{F} \geq f) \quad \leq \quad \epsilon / f$$
Also, from Lemma 2, the change in any $\hat{\phi}$ is at most $\log(T_4 n m^{2t})$.
Let $\Delta\varepsilon(\theta) = |\varepsilon^{A}_{(x,y)}(\theta) - \varepsilon^{A\cup B}_{(x,y)}(\theta)|$. Using Lemma 3, the above bounds on the changes in $\hat{\phi}$, and the fact that there are at most $2n$ parameters used for any document,
$$E[\Delta\varepsilon(\theta)] \quad \leq \quad \frac{\epsilon}{f} \cdot 2\eta R \log(T_4 n m^{2t}) + \left[f \cdot 2n \cdot \eta \log(T_4 n m^{2t}) + 2\eta R D \right]$$
With $f = 1/mn$, the Markov inequality applied to $\Delta\varepsilon(\theta)$ yields (for $m = \Omega(\log n)$):
$$P(\Delta\varepsilon(\theta) \geq \frac{1}{m^\tau}) \quad \leq \quad \epsilon \cdot \theta(m, n, R) + T_5 R \frac{m^{t+\tau}}{a^{1-c}}$$
where $0 < \tau < 1$, T_5 is a constant and $\theta(m, n, R)$ is a polynomial of total degree less than $3 + \tau$. Since ϵ decreases exponentially with m, choosing $m = \Omega(\log(nR))$ leads to the required probability bound.
2. This follows from Lemma 2 and Lemma 3. □

Note that the above corollary applies to the likelihood of any example (x, y) as long as it is drawn from the same distribution $\mathcal{D}$. Thus, it also applies to the true log likelihoods $\varepsilon^A(\theta)$ and $\varepsilon^{A\cup B}(\theta)$ since they represent the expected value of the log likelihood.

Our algorithm uses an estimate of the true log likelihood, which we call the leave-one-out log likelihood:
$$\hat{\varepsilon}^{M}_{-1}(\theta) \quad = \quad \frac{1}{M} \sum_{i=1}^{m} \log P^{M}_{-i}(y^{(i)}|x^{(i)}) \tag{19}$$
where P^{M}_{-i} represents the probability model trained with training set $M \setminus \{(x^{(i)}, y^{(i)})\}$, i.e., the set M with the ith example left out.

The main result of this section is to show that the leave-one-out log likelihood is a "good" estimate of the unknown true likelihood, irrespective of the underlying distribution $\mathcal{D}$; and the accuracy of the estimate increases "quickly" as more and more training examples are provided.

We now present such a uniform convergence result for fixed parameters θ.

Theorem 5 (Uniform Convergence With Fixed Parameters) *Suppose* M *is a set of training examples* $\{(x^{(i)}, y^{(i)})\}$, $1 \leq i \leq m$*, drawn from* $\mathcal{D}$*. The* R *model parameters*

θ are fixed, with $||\theta||_\infty \leq \eta$. Then, the true log likelihood $\varepsilon^M(\theta)$, and the leave-one-out log likelihood $\hat{\varepsilon}^M_{-1}(\theta)$ converge asymptotically. We prove a relation of the form:

$$P(|\varepsilon^M(\theta) - \hat{\varepsilon}^M_{-1}(\theta)| < 1/m^\nu) \quad > \quad 1 - T\frac{R^3 \log^2(mn)}{m^\theta} \tag{20}$$

for constants $T, \nu, \theta > 0$ and m large enough ($m > \Omega(\log(nR))$. Thus the error tends to zero as $m \to \infty$, with only a logarithmic dependence on document length n, and no direct dependence on vocabulary size $|\mathcal{W}|$.

Proof outline: Our proof roughly consists of the following steps:

- Split the m examples in the training set M into two parts A and B such that $|B| = m^t$ for some $0 < t < 1$. By construction, B is much smaller than M, but still contains many examples. Intuitively, we will use testing on the set B as a bridge between testing on single examples (as in $\hat{\varepsilon}^M_{-1}(\theta)$) and (hypothetical) testing on infinitely many examples (as in $\varepsilon^M(\theta)$).
- Show that (with high probability) training on A and testing on B produces a similar log likelihood estimate (call it $\tilde{\varepsilon}^{A,B}(\theta)$) to training on $A \cup B$ and testing from the distribution $\mathcal{D}$ itself (this latter value would be exactly $\varepsilon^M(\theta)$).
- Show that (with high probability) training on $A \cup B$ with testing performed by leave-one-out averaging over B produces a similar log likelihood estimate (call it $\hat{\varepsilon}^{A,B_{-1}}(\theta)$) to training on A and testing on B ($\tilde{\varepsilon}^{A,B}(\theta)$).
- Use the above two to get a high probability and low error link between training on $M = A \cup B$ with leave-one-out averaging over B ($\hat{\varepsilon}^{A,B_{-1}}(\theta)$) and $\varepsilon^M(\theta)$.
- Finally, consider disjoint ways of forming B in the first place, and average the previous result over them to prove the theorem.

Proof: Split M into A and B, with $|B| = m^t$.

- $\tilde{\varepsilon}^{A,B}(\theta) \approx \varepsilon^M(\theta)$**:** First note that B is fairly large, and so testing on B gives a good estimate of the true log likelihood after training on A. Mathematically, a high probability bound is obtained using the Chernoff Bound. Combining this with the result from Corollary 4 for $\tau > t/2$ and m much larger than $\log(nR)$, we get:

  $$P(|\tilde{\varepsilon}^{A,B}(\theta) - \varepsilon^M(\theta)| > 1/m^\tau) \quad < \quad T_6 R \frac{m^{t+\tau}}{a^{1-c}}$$

  where $a = m - m^t$ and T_6 is a constant. This gives a useful bound for $\tau < 1 - c - t$.
- $\hat{\varepsilon}^{A,B_{-1}}(\theta) \approx \tilde{\varepsilon}^{A,B}(\theta)$**:** The advantage here is that the tests are made on exactly the same elements - those from B. Choose any element in B, call it B_i. For the test on B_i, in one case parameters are estimated from $A \cup B \setminus \{B_i\}$ and in the other case from A. Consider the difference in log likelihood for B_i in these two cases. Corollary 4 gives a high probability bound on the difference, and also a bound on the maximum difference[6]. Thus the expected value is bounded by:

  $$E[|\hat{\varepsilon}^{A,B_{-1}}(\theta) - \tilde{\varepsilon}^{A,B}(\theta)|] \quad < \quad T_5 R \frac{m^{t+\tau}}{a^{1-c}} \cdot 2\eta R \log(T_4 n m^{2t}) + \frac{2\eta R}{m^\tau}$$

  As before, applying the Markov inequality gives a high probability bound:

  $$P(|\hat{\varepsilon}^{A,B_{-1}}(\theta) - \tilde{\varepsilon}^{A,B}(\theta)| > 1/m^\chi) \quad < \quad \underbrace{T_7 R^2 \frac{m^{t+\tau+\chi} \log(T_4 n m^{2t})}{a^{1-c}} + \frac{T_8 R}{m^{\tau-\chi}}}_{\text{call this } b}$$

  where $a = m - m^t$ and T_7, T_8 are constants. We get a useful bound when $\chi < 1 - c - t - \tau$ and $\chi < \tau$.

[6]We require adding only $(m^t - 1)$ new elements instead of the m^t in Corollary 3, so this case can only be better

- Combining the above two results, we can relate the true log likelihood with the complete training set M to the log likelihood estimated by leave-one-out averaging on the part B.
- Finally, consider the m^{1-t} disjoint ways of choosing B. Our overall leave-one-out measure is simply the average of the leave-one-out averages over all these choices of B. Again, bound the expected value, and apply the Markov inequality to get:

  $$P(|\varepsilon^M(\theta) - \hat{\varepsilon}^M_{-1}(\theta)| > 1/m^\nu) \quad < \quad b \cdot m^\nu \cdot 2\eta R \log(T_4 n m^{2t}) + \frac{2\eta R}{m^{\chi-\nu}}$$

  where we want $\nu < \chi$, $\chi + \nu < \tau$ and $\theta = 1 - c - t - \tau - \chi - \nu > 0$ to get useful bounds. Now, we finally choose the values as, for example, $\nu = 1/20$, $\chi = 1/10$, $\tau = 1/5$, $t = 1/20$ and $c = 1/20$ to get a bound with $\nu = \theta = 1/20$.

□

Our proof is only intended to show the nature of dependence of the error on m, n, $|\mathcal{W}|$ and R. The exact numbers can surely be improved by a tighter analysis.

Theorem 6 (Uniform Convergence) *Suppose the Naive Bayes parameters are estimated on a training set M. If we restrict the R parameters such that $||\theta||_\infty \leq \eta$, then the true log likelihood and the leave-one-out log likelihood asymptotically converge for any parameter setting θ. Mathematically,*

$$P\left(\exists\, \theta \text{ s.t. } |\varepsilon^M(\theta) - \hat{\varepsilon}^M_{-1}(\theta)| > T_1 \frac{Rlog(nm + |\mathcal{W}|)}{m^{\theta/2R}}\right) \quad < \quad T_2 \frac{(2\eta+1)^R \cdot R^3 \log^2(mn)}{m^{\theta/2}}$$

for some constants $T_1, T_2, \theta > 0$.

Proof: Divide the $[-\eta, \eta]$ range for each parameter into fragments of length γ to get a grid with $n_g = ((2\eta+1)/\gamma)^R$ grid points . The previous result can be applied at every grid point θ_g. Combining all the grid points using a union bound, we get:

$$P(\exists \text{ grid point } \theta_g \text{ s.t. } |\varepsilon^M(\theta_g) - \hat{\varepsilon}^M_{-1}(\theta_g)| > 1/m^\nu) \quad < \quad n_g \cdot T_2 \frac{R^3 \log^2(mn)}{m^\theta}$$

for some constant T_2. Suppose we now have some parameters θ that do no lie on a grid point. However, we can always find a grid point (say θ_G) such that $||\theta - \theta_G||\infty < \gamma$. Similar to Lemma 3, it can be shown that:

$$|\frac{d\varepsilon^M(\theta)}{d\theta_r}| \quad \leq \quad |\log P(x_i^r|y=1)| \ \leq \ \log(nm + |\mathcal{W}|)$$

Using such a bound on the derivative, the maximum change in the log likelihood can be bounded in both cases.

$$\begin{aligned} |\varepsilon^M(\theta) - \varepsilon^M(\theta_G)| &\quad < \quad \gamma \log(nm + |\mathcal{W}|)R \\ |\hat{\varepsilon}^M_{-1}(\theta) - \hat{\varepsilon}^M_{-1}(\theta_G)| &\quad < \quad \gamma \log(nm + |\mathcal{W}|)R \end{aligned}$$

Combining the above bounds and choosing $\nu > \theta/2R$ and $\gamma = m^{-\theta/2R}$ gives the required result. □

Theorem 7 (Sample Complexity) *Suppose we want to ensure, with probability at least $(1-\delta)$, that $|\varepsilon^M(\theta) - \hat{\varepsilon}^M_{-1}(\theta)| < \epsilon$ for all parameters θ such that $||\theta||_\infty \leq \eta$. Then it suffices to choose $m = O(poly(1/\delta, 1/\epsilon, \log n, \log|\mathcal{W}|, R, \eta)^R)$ where $poly(\ldots)$ denotes a function polynomial in its arguments.*

Proof: Follows from Theorem 6. □