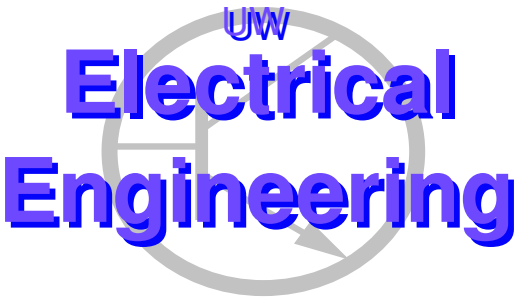
Necessary Intransitive Likelihood-Ratio Classifiers

Gang Ji, Jeff Bilmes
{gang,bilmes}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500



UWEE Technical Report
Number UWEETR-2002-0014
December 2002

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Necessary Intransitive Likelihood-Ratio Classifiers

Gang Ji, Jeff Bilmes
{gang,bilmes}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

University of Washington, Dept. of EE, UWEETR-2002-0014
December 2002

Abstract

In any pattern classification task, errors are introduced because of the difference between the true generative model and the one obtained via model estimation. One approach to solve this problem uses more training data and more accurate (but often more complicated) models. In previous work we address this problem differently, by trying to compensate (post log-likelihood ratio) for the difference between the true and estimated model scores. This was done by adding a bias term to the log likelihood ratio that was based on an initial pass on the test data, thereby producing an intransitive classifier. In this work, we extend the previous theory by noting that the correction term used before was sufficient but not necessary for perfect correction. We derive weaker (necessary) conditions that still lead to perfect correction, and therefore might be more easily attainable. We test a number of new schemes on an isolated-word speech recognition task as well as on the UCI machine learning data sets. Results show that by using the bias terms calculated in this new way, classification accuracy substantially improves over both the baseline and over our previous results.

1 Introduction

Statistical pattern recognition is often based on Bayes decision theory (Duda & Hart, 2000), which aims to achieve minimum error rate classification. Given an observation x of a random variable $X : \Omega \rightarrow \mathbb{R}^n$, for each classification decision, say class c', a loss function $L(c'|c)$ is associated with the decision if the true answer is class c. The goal is to find a class c^* that minimizes the conditional risk $R(c'|x)$,

$$c^* = \underset{c'}{\text{argmin}}\ R(c'|x) = \underset{c'}{\text{argmin}} \sum_c L(c'|c)P(c|x). \tag{1}$$

Among different loss functions, the zero-one loss function is the most commonly used $L(c'|c) = 1 - \delta_{c'c}$, which means that the loss for a correct decision is zero, and for any wrong decision the loss is 1, and δ_{ij} is the Krönecker delta function. With this $0/1$-loss function, the decision rule becomes

$$c^* = \underset{c}{\text{argmax}}\ P(c|x) = \underset{c}{\text{argmax}}\ P(x|c)P(c), \tag{2}$$

which is based on the posterior probability and is called the minimum-error-rate decision rule.

In previous work (Bilmes et al., 2001), we observed that multi-class Bayes classification can be viewed as a tournament style game, where the winner between "players" is decided using (log) likelihood ratios. Supposing the classes (players) are $\{c_1, c_2, \cdots, c_M\}$, and the observation (game) is x, the winner of each pair of classes is determined, with the assumption of equal priors, by the sign of the log likelihood ratio $L_{ij}(x) = \ln \frac{P(x|c_i)}{P(x|c_j)}$, in which case if $L_{ij} > 0$ class c_i wins and otherwise class c_j wins. A practical game strategy can be obtained by fixing a comparison order, $\{i_1, i_2, \cdots, i_M\}$, as a permutation of $\{1, 2, \cdots, M\}$, where class c_{i_1} plays class c_{i_2} , the winner plays class c_{i_3} , and so on until a final winner is ultimately found. This yields a transitive game (Luce & Raiffa, 1957) — assuming no ties, the ultimate winner is identical regardless of the comparison order.

To perform these procedures optimally, correct likelihood ratios are needed, which requires correct probabilistic models and sufficient training data. Given a finite amount of training data or the wrong model family, typical in practice, this is never the case. In previous work (Bilmes et al., 2001), we introduced a method to correct for the difference between the true and an approximate log likelihood ratio. In this work, we improve upon the correction method by using an expression that can still lead to perfect correction, but is weaker than what we used before. We show that this new condition achieves a significant improvement over baseline results, both on a medium vocabulary isolated-word automatic speech recognition task and on the UCI machine learning data sets. The paper is organized as follows: Section 2 describes the general scheme and describes past work. Section 3 discusses the weaker correction condition, and its approximations. Section 4 provides various experimental results on an isolated-word speech recognition task. Section 5 contains the experimental results on the UCI data. Finally, Section 6 concludes.

2 Background

A common problem in many probabilistic machine learning settings is the lack of a correct statistical model. In a generative pattern classification setting, this occurs because only an estimated quantity $\hat{P}(x|c)$[1] of a distribution is available, rather than the true class-conditional model $P(x|c)$. In the likelihood ratio decision scheme described above, only an imperfect log likelihood ratio is available for decision making $\hat{L}_{ij}(x) = \ln \frac{\hat{P}(x|c_i)}{\hat{P}(x|c_j)}$ rather than the true log likelihood ratio $L_{ij}(x) = \ln \frac{P(x|c_i)}{P(x|c_j)}$.

One approach to correct for this inaccuracy is to use richer class conditional likelihoods, more complicated parametric forms of $L_{ij}(x)$ itself, and/or more training data. In previous work (Bilmes et al., 2001), we proposed a different approach that requires no change in generative models, no increase in free parameters, and no additional training data but still yields improved accuracy. The key idea is to compensate for the difference between $L_{ij}(x)$ and $\hat{L}_{ij}(x)$ using a *bias*[2] term $\alpha_{ij}(x)$ computed from test data such that:

$$L_{ij}(x) - \alpha_{ij}(x) = \hat{L}_{ij}(x). \tag{3}$$

If it is assumed that a single bias term is used for all data, so that $\alpha_{ij}(x) = \alpha_{ij}$, we found that the best α_{ij} is as follows:

$$\alpha_{ij} = \frac{1}{2}\left(D(i\|j) - D(j\|i)\right) - \frac{1}{2}\left(\hat{D}(i\|j) - \hat{D}(j\|i)\right), \tag{4}$$

where $D(i\|j) = E_{P(x|c_i)} \ln L_{ij}(x)$ is the Kullback-Leibler (KL) divergence (Cover & Thomas, 1991) between $P(x|c_i)$ and $P(x|c_j)$ and $\hat{D}(i\|j) = E_{P(x|c_i)}\hat{L}_{ij}(x)$ is its estimation. Under the assumption of symmetric KL-divergence for the true model (e.g., equal covariance matrices in the Gaussian case), the bias term can be solved explicitly as

$$\alpha_{ij} = -\frac{1}{2}\left(\hat{D}(i\|j) - \hat{D}(j\|i)\right). \tag{5}$$

We saw how the augmented likelihood ratio $S_{ij}(x) = \hat{L}_{ij}(x) + \alpha_{ij}$ can lead to an intransitive game (Luce & Raiffa, 1957; Straffin, 1993), since $S_{ij}(x)$ can specify intransitive preferences amongst the set $\{1, 2, \cdots, M\}$. We therefore investigated a number of intransitive game playing strategies. Moreover, we observed that if the correction was optimal, the true likelihood ratios would be obtained which are clearly transitive. We therefore hypothesized and experimentally verified that the existence of intransitivity was a good indicator of the occurrence of a classification error.

This general approach can be improved upon in several ways. First, better intransitive strategies can be developed (for detecting, tolerating, and utilizing the intransitivity of a classifier); second, the assumption of symmetric KL-divergence could be relaxed; and third, the above criterion is stricter than required to obtain perfect correction. In this work, we advance on the latter two of the above three possible avenues for improvement.

3 Necessary Intransitive Scheme

An $\alpha_{ij}(x)$ that solves Equation 3 is a sufficient condition for a perfect correction of the estimated likelihood ratio since given such a quantity, the true likelihood ratio would be attainable. This condition, however, is stricter than required

[1] In this paper, we use "hatted" letters to describe estimated quantities.

[2] Note that by *bias*, we do not mean standard parameter bias in statistical parameter estimation.

because it is only the sign of the likelihood ratio that is needed to decide the winning class. We therefore should ask for a condition that corrects only for the discrepancy in sign between the true and estimated ratio, i.e., we want to find a function $\alpha_{ij}(x)$ that minimizes

$$J[\alpha_{ij}] = \int_{\mathbb{R}^n} \left\{ \text{sgn}\left[L_{ij}(x) - \alpha_{ij}(x)\right] - \text{sgn}\hat{L}_{ij}(x) \right\}^2 \cdot P(x)\, dx.$$

Clearly the $\alpha_{ij}(x)$ that minimizes $J[\alpha_{ij}]$ is the one such that $\forall x \in \text{supp}P = \overline{\{x : P(x) \neq 0\}}$,

$$\text{sgn}\left[L_{ij}(x) - \alpha_{ij}(x)\right] = \text{sgn}\hat{L}_{ij}(x). \tag{6}$$

As can be seen, this condition is weaker than Equation 3, weaker in the sense that any solution to Equation 3 solves Equation 6 but not vice versa. Note also that Equation 6 provides *necessary* conditions for an additive bias term to achieve perfect correction, since any such correction must achieve parity in the sign. Therefore, it might make it simpler to find a better bias term since Equation 6 (and therefore, set of possible α values) is less constrained. As will be seen, however, analysis of this weaker condition is more difficult. In the following sections, therefore we introduce several approximations to this condition.

Note that as in previous work, we henceforth assume $\alpha_{ij}(x) = \alpha_{ij}$ is a constant. In this case, the equation providing the best α_{ij} values is:

$$E_{P_{ij}}\left\{\text{sgn}\left[L_{ij}(x) - \alpha_{ij}\right]\right\} = E_{P_{ij}}\left\{\text{sgn}\hat{L}_{ij}(x)\right\}. \tag{7}$$

3.1 The Problem with Sign

The main problem in trying to solve for α_{ij} in Equation 7 is the existence of a discontinuous function. In this section, therefore, we work towards obtaining an analytically tractable approximation. The sign function $\text{sgn}(z)$ is defined as

$$\text{sgn}(z) = 2u(z) - 1 = \begin{cases} 1 & \text{if } z > 0, \\ 0 & \text{if } z = 0, \\ -1 & \text{if } z < 0, \end{cases}$$

where $u(z)$ is the Heaviside step function. We obtain an approximation via a Taylor expansion as follows:

$$\begin{aligned} \text{sgn}(z+\epsilon) &= \text{sgn}(z) + \epsilon\text{sgn}'(z) + o(\epsilon) \\ &= \text{sgn}(z) + 2\epsilon\delta(z) + o(\epsilon), \end{aligned} \tag{8}$$

where $\delta(z)$ is the Dirac delta function (Kevorkian, 2000). It can be defined as the derivative of the Heaviside step function $u'(z) = \delta(z)$, and it satisfies the sifting property $\int_{\mathbb{R}} f(z)\delta(z - z_0) = f(z_0)$. Therefore, it follows that (Jones, 1966, page 263)

$$\int_{\mathbb{R}^n} f(z)\delta[g(z)]\, dz = \int_{Z_g} \frac{f(z)}{|\nabla g(z)|} \cdot d\mu,$$

where ∇g is the gradient of the field g and $Z_g = \{z \in \mathbb{R}^n : g(z) = 0\}$ is the zero set of g with Lebesgue measure μ (Rao, 1987).

Of course, the Taylor expansion is valid only for a differentiable function, otherwise the error terms can be arbitrarily large. If, however, we find and use a suitable continuous and differentiable approximation rather than the discrete sign function, the above expansion becomes more appropriate. There exists a trade-off, however, between the quality of the sign function approximation (a better sign function should yield a better approximation in Equation 6) and the error caused by the $o(\epsilon)$ term in Equation 8 (a better sign function approximation will have a greater error when the higher-order Taylor terms are dropped). We therefore expect that ideally there will exist an optimal balance between the two.

Retaining the first-order Taylor term, and applying this to the left side of Equation 6, we get

$$\text{sgn}\left[L_{ij}(x) - \alpha_{ij}\right] \approx \text{sgn}L_{ij}(x) - 2\alpha_{ij}\delta\left[L_{ij}(x)\right].$$

The distribution under which the expectation in Equation 7 is taken can also influence our results. If it is known that the true class of x is always c_i, the c_i-conditional distribution should be used, i.e., $P_{ij}(x) = P(x|c_i)$, yielding a class-conditional correction term $\alpha_{ij}^{(i)}$, and a class-conditional likelihood-ratio correction $S_{ij}^{(i)}(x) = \hat{L}_{ij}(x) + \alpha_{ij}^{(i)}$. The symmetric case arises when x is of class c_j. If, on the other hand, neither c_i nor c_j are the true classes (i.e., x is sampled from some other class-conditional distribution, say $P(x|c_k)$), it does not matter which distribution for $P_{ij}(x)$ is used since, for a given comparison order in a game playing strategy, either winner will ultimately play using the true class distribution $P(x|c_k)$ of x. It is therefore valid to consider the case only when either $x \in C_i$ (i.e., x is of class c_i) or $x \in C_j$.

In practice, however, we do not know which of the two are correct. The ideal choice in either case can be expressed using indicators as follows:

$$A_{ij} = \alpha_{ij}^{(i)} \mathbf{1}_{\{x \in C_i\}} + \alpha_{ij}^{(j)} \mathbf{1}_{\{x \in C_j\}}.$$

Taking the expected value of A_{ij} with respect to $p(x|c_i \cup c_j)$ yields

$$\alpha_{ij} = E_{p(x|c_i \cup c_j)}[A_{ij}] = \frac{\alpha_{ij}^{(i)} P(c_i) + \alpha_{ij}^{(j)} P(c_j)}{P(c_i) + P(c_j)}.$$

This results in a single likelihood correction $S_{ij}(x) = \hat{L}_{ij}(x) + \alpha_{ij}$ that is obtained simply by integrating in Equation 7 with respect to the average distribution over class c_i and c_j, i.e.,

$$P_{ij}(x) = \frac{P(c_i)P(x|c_i) + P(c_j)P(x|c_j)}{P(c_i) + P(c_j)}.$$

With these assumptions, and supposing the zero set $Z_{L_{ij}} = \{x \in \mathbb{R}^n : P(x|c_i) = P(x|c_j)\}$ of $L_{ij}(x)$ is Lebesgue measurable with measure μ, we get:

$$\begin{aligned} & \int_{\mathbb{R}^n} \{\text{sgn} L_{ij}(x) - 2\alpha_{ij}\delta\left[L_{ij}(x)\right]\} P_{ij}(x)\, dx \\ = & \int_{\mathbb{R}^n} \text{sgn} L_{ij}(x) P_{ij}(x)\, dx - 2K(P_i, P_j)\alpha_{ij}, \end{aligned}$$

where

$$K(P_i, P_j) = \int_{Z_{L_{ij}}} \frac{P_{ij}(x)}{|\nabla L_{ij}(x)|} \cdot d\mu. \tag{9}$$

Therefore,

$$\alpha_{ij} = \frac{1}{2K(P_i, P_j)} \int_{\mathbb{R}^n} \left[\text{sgn} L_{ij}(x) - \text{sgn}\hat{L}_{ij}(x)\right] P_{ij}(x)\, dx.$$

The quantity $K(P_i, P_j)$ is a form of probabilistically weighted smoothness measure of $L_{ij}(x)$ along the c_i/c_j decision boundary. When $P(x|c_i)$ and $P(x|c_j)$ are 1-dimensional Gaussian distributions with means μ_i and μ_j, identical variances σ^2, and equal priors, the zero set is $Z_{L_{ij}} = \{\frac{\mu_i + \mu_j}{2}\}$. Therefore,

$$K(P_i, P_j) = \frac{\sigma e^{-\frac{(\mu_i - \mu_j)^2}{8\sigma^2}}}{\sqrt{2\pi}\,|\mu_i - \mu_j|}.$$

On the one hand, when the two means are close, $K(P_i, P_j)$ is large thereby forcing α_{ij} to be small (i.e., little decision boundary shift is allowed). On the other hand, when the means are distant, $K(P_i, P_j)$ is small allowing for a large decision boundary shift. Similar such results can be derived for multi-variate Gaussians with differing means and covariances.

Unfortunately, it is quite difficult to explicitly evaluate $K(P_i, P_j)$ without knowing the true probability distributions. In this work, therefore, we make the bold assumption that a reasonable nominal value for this factor is unity. As will be seen, this assumption yields a likelihood-ratio adjustment that is similar in form to our previous KL-divergence based adjustment. More practically, the assumption significantly simplifies the derivation and still yields good empirical results. Under this assumption, expression for α_{ij} becomes:

$$\alpha_{ij} = \frac{1}{2} E_{P_{ij}(x)}[\text{sgn} L_{ij}(x)] - \frac{1}{2} E_{P_{ij}(x)}[\text{sgn}\hat{L}_{ij}(x)]. \tag{10}$$

The left term on the right of the equality is quite similar to the left difference on the right of the equality in the KL-divergence case (Equation 4). Again, because we have no information about the true class conditional models, we assume the left term in Equation 10 to be zero (denote this as assumption B). Comparing this with the corresponding assumption for the KL-divergence case (assumption A, Equation 4), it can be shown that 1) they are not identical in general, and 2) in the Gaussian case, A implies B but not vice versa, meaning B is weaker than A.

Under assumption B, an expression for the resulting α_{ij} can be derived using the weak law of large numbers yielding:

$$\alpha_{ij} \approx \frac{1}{2(N_i + N_j)} \left(\sum_{x \in c_i} \operatorname{sgn} \ln \frac{\hat{P}(x|c_j)}{\hat{P}(x|c_i)} - \sum_{x \in c_j} \operatorname{sgn} \ln \frac{\hat{P}(x|c_i)}{\hat{P}(x|c_j)} \right), \tag{11}$$

where $x \in C_i$ and $x \in C_j$ corresponds to the samples as they are classified in a previous recognition pass; N_i and N_j are number of samples from model c_i and c_j respectively. Like in (Bilmes et al., 2001), since the true classes are unknown, we perform a previous classification pass (e.g., using the original likelihood ratios) to get estimates and use these in Equation 11.

Note that there are three potential sources of error in the analysis above. The first is the $K(P_i, P_j)$ factor that we neglected. The second is assumption B, that (since weaker) can be less severe than in the corresponding KL-divergence case. The third is the error due to the discontinuity of the sign function. To address the third problem, rather than using the sign function in Equation 11, we can approximate it with a continuous differential function with the goal of balancing the trade-off mentioned above. In Figure 1 (top), we show three possible sign-function approximations: hyperbolic tangent, arctangent, and a shifted sigmoid function. In Figure 1 (bottom), the shifted sigmoid is presented with several values of its free parameter *beta*(β). In the following, we derive expressions for α_{ij} for each of these sign approximations.

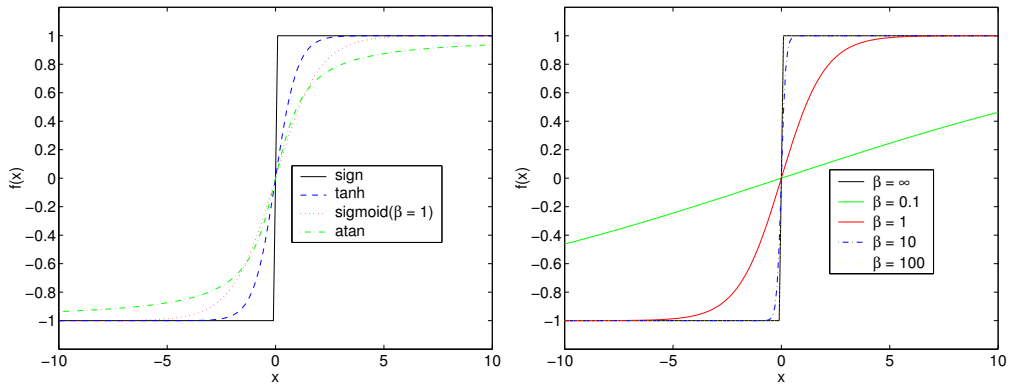


Figure 1: Top: Several approximating sign functions. Bottom: Shifted sigmoid with different β values.

3.2 Hyperbolic tangent

In the hyperbolic tangent case:

$$\operatorname{sgn} z \approx \tanh z = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

which yields:

$$\alpha_{ij} \approx \frac{1}{2} \int_{\mathbb{R}^n} \frac{\hat{P}^2(x|c_j) - \hat{P}^2(x|c_i)}{\hat{P}^2(x|c_i) + \hat{P}^2(x|c_j)} P_{ij}(x)\, dx$$

or again using the law of large numbers,

$$\alpha_{ij} \approx \frac{1}{2(N_i+N_j)} \left(\sum_{x \in c_i} \frac{\hat{P}^2(x|c_j) - \hat{P}^2(x|c_i)}{\hat{P}^2(x|c_i) + \hat{P}^2(x|c_j)} - \sum_{x \in c_j} \frac{\hat{P}^2(x|c_i) - \hat{P}^2(x|c_j)}{\hat{P}^2(x|c_i) + \hat{P}^2(x|c_j)} \right). \tag{12}$$

3.3 Arctangent

We can also replace the sign function by arctangent.

$$\text{sgn}z \approx \frac{2}{\pi} \tan^{-1} z.$$

In this case,

$$\alpha_{ij} \approx \frac{1}{\pi(N_i+N_j)} \left(\sum_{x \in c_i} \tan^{-1} \ln \frac{\hat{P}(x|c_j)}{\hat{P}(x|c_i)} - \sum_{x \in c_j} \tan^{-1} \ln \frac{\hat{P}(x|c_i)}{\hat{P}(x|c_j)} \right). \tag{13}$$

3.4 Shifted Sigmoid

The sigmoid function is the solution of the differential equation $y' = y(1-y)$ and has the form $f(z) = \frac{1}{1+e^{-\beta z}}$, where the free parameter β (interpreted as inverse temperature) determines how well the curve will approximate a discontinuous function (see Figure 1 (bottom)). Using the sigmoid function, we can approximate the sign function as

$$\text{sgn}z \approx \frac{2}{1+e^{-\beta z}} - 1.$$

Note that the approximation improves as β increases. Hence,

$$\alpha_{ij} \approx \frac{1}{2(N_i+N_j)} \left[\sum_{x \in c_i} \left(1 - \frac{2}{1 + \frac{\hat{P}^\beta(x|c_j)}{\hat{P}^\beta(x|c_i)}} \right) - \sum_{x \in c_j} \left(1 - \frac{2}{1 + \frac{\hat{P}^\beta(x|c_i)}{\hat{P}^\beta(x|c_j)}} \right) \right]. \tag{14}$$

4 Speech Recognition Evaluation

Similar to previous work (Bilmes et al., 2001), we implemented this technique on NYNEX PHONEBOOK (Bilmes, 1999; Pitrelli et al., 1995), a medium vocabulary isolated-word speech corpus. A Gaussian mixture hidden Markov model (HMM) is used to calculate the probability scores $\hat{P}(x|c_i)$ where in this case x is a matrix of feature values (one dimension as MFCC features and the other as time frames), and c_i is a given spoken word. The HMM models use four hidden states per phone, and 12 Gaussian mixtures per state. This yields approximately 200k free model parameters in total. In our experiments, the steps are as follows: 1) calculate $\hat{P}(x|c_i)$ using full HMM inference (no Viterbi approximation) for each test case and for each class (word); 2) classify the examples using just the log likelihood ratios $\hat{L}_{ij} = \ln \frac{\hat{P}(x|c_i)}{\hat{P}(x|c_j)}$; 3) calculate the bias term using one of the techniques described above; 4) classify again using the new improved likelihood ratio $S_{ij} = \hat{L}_{ij} + \alpha_{ij}$. In this case, since the procedure is no longer transitive, we run 1000 random tournament-style games (as in (Bilmes et al., 2001)) and choose the most frequent winner as the ultimate winner. The results are shown in Table 1.

Table 1: WERs with various sign approximations.

SIZE	ORIG	SIGN	TANH	ATAN	SIG(1)
75	2.3358	1.7584	1.7584	1.7581	1.7584
150	3.3107	2.8258	2.8382	2.8269	2.8258
300	5.2251	4.7524	4.7492	4.6984	4.7524
600	7.3927	6.6383	6.6109	6.5972	6.6383

Table 2: WERs, shifted sigmoid, different β values

SIZE β	0.1	1.0	10	100	KLD
75	1.8228	1.7584	1.5581	1.5708	1.9147
150	2.6502	2.8258	2.6523	2.4664	2.7228
300	4.7448	4.7524	4.2855	3.9502	4.2893
600	6.6581	6.6383	6.0409	5.6980	5.9144

In the table, the first column gives the vocabulary size (number of different classes) in the test data; the second column shows the baseline word error rate (WER) using only $\hat{L}_{ij}$; the remaining columns are the results using the various bias terms. In the shifted sigmoid case, $\beta = 1$. As can be seen, in all cases the new methods yield significant accuracy improvement over the baseline, while the various sign function approximations perform about the same.

The shifted sigmoid function can be fine-tuned to approximate the sign function by changing β. This function is particularly useful since it allows us to investigate the trade-off mentioned in Section 3.1. The results are shown in Table 2 for $\beta = \{0.1, 1.0, 10, 100\}$. From the results we can see that the overall performance increases as we increase the inverse of temperature, β. This is because when β increases, the shifted sigmoid curve is a better approximation to the sign function. For $\beta = 100$, the results here show an improvement over the comparable KL-divergence results reported in (Bilmes et al., 2001) (shown in the right-most column in the table). We are currently investigating larger β values to determine when the inaccuracies due to the Taylor error term start affecting the results (note, however, that for the 75 word case, it seems this has occurred at $\beta = 100$).

5 UCI Dataset Evaluation

In order to show that our methodology is general beyond isolated-word speech recognition, we also evaluated this technique on the UCI machine learning repository (Murphy & Aha, 1995). In our experiments, baseline classifiers are built using the Matlab neural network toolbox with feed-forward 3-layer MLPs with different number of hidden units, and the Levenberg-Marquardt algorithm (Press et al., 1992) is used for training. Table 3 shows properties of the UCI data sets. The first column shows the data set name. The second column shows the number of attributes in each the data set, which can be either discrete or continuous. The third column gives the number of classes, and the fourth gives the total number of data set samples (that must be divided into training and test sets). Finally, the last column shows the number of hidden units used in the corresponding feed-forward baseline neural network. Note that a neural-network was used even if attributes were missing in the data set (i.e., hidden features). In such a case, those samples were simply removed from consideration since a neural-network is unable to marginalize out a missing attribute as can a Bayesian network.

Note that not all of the UCI data sets were used in our experiments. This was because either 1) a missing attributes in each sample meant no training data was available for the neural network, or 2) we simply could not find the data in the UCI repository or the data matching previous descriptions (Friedman et al., 1997; Friedman et al., 1998). In general, we applied our technique and report results for as many data sets as was possible.

All experimental results use 5-fold cross-validation using randomly selected chunks. All results show mean (and standard deviation) of performance. Logistic sigmoid is used for the hidden layers, and the soft-max function is used for the output layers, making the network outputs interpretable as posterior probabilities $P(c|x)$, where x is the sample and c is the class. Note this is different from the hidden Markov model (HMM) classifiers above that

Table 3: UCI data description (number attributes, number classes, total samples, number hidden units)

DATASET	ATTR	CLS	INST	HIDDEN
AUTO	25	7	159	7
BALANCE-SCALE	3	3	625	30
BREAST	9	2	630	30
CHESS	36	2	2130	10
CLEVE	13	2	296	10
FLARE	10	2	1066	10
GLASS	9	7	214	40
HAYES-ROTH	5	3	132	5
HEART	13	2	270	10
HEPATITIS	19	2	80	15
IONOSPHERE	32	2	351	10
IRS	4	3	150	7
LETTER	16	26	15000	100
LIVER-DISORDER	6	2	345	10
PIMA	8	2	768	5
VOTE	16	2	435	40
WAVEFORM-21	21	3	300	3

produce likelihoods $P(c|x)$. But the same procedures can be used, in this case the posteriors are divided by the priors giving the relation $P(c|x)/P(c) = P(x|c)/p(x)$ (i.e., scaled likelihoods) which when used in a likelihood ratio produces the standard $L_{ij}(x)$ values. Note also that on these data sets, so far we have only tried one random tournament game to decide the winner. The experimental procedure is as follows for each of the five data folds: 1) train the neural network on 4/5-sized training partition; 2) produce scaled likelihoods for the 1/5-sized partition using the neural network outputs and priors; 3) compute the bias correction terms (i.e., KL-divergence based or sign based) using initial classification hypotheses made in step 2; 4) re-do classification using the corrected likelihood-ratios with a single random game.

Table 4 shows results using the just the KL-divergence-based corrections, and compares with results from the literature. Again, the first column is the data-set name. The second column shows baseline accuracy with the 5-fold standard derivations. The third column gives results after KL-divergence based bias corrections. The remaining columns show results from (Friedman et al., 1997; Friedman et al., 1998). Note that our baseline results use simply trained neural networks while those in (Friedman et al., 1997; Friedman et al., 1998) use naïve Bayes (NB) classifiers, and tree augmented naïve Bayes (TAN) classifiers (Friedman et al., 1997), or discrete TAN classifiers (Friedman et al., 1998). We attempted as much as possible to optimize our baseline results (i.e., by varying the number of hidden nodes, number of training epochs, and so on). As can be seen, in some cases our baselines are worse than the Bayesian network (e.g., "letter"), presumably because of a poor model match even given our attempt to optimize over number of hidden units. In other cases, our baselines are better (e.g., "chess", "glass", and "wave-form-21"). In any event, our goal is to compare our baselines with the likelihood ratio corrections our techniques can offer them. From the table, it can be seen that in all cases the correction terms calculated from the KL-divergence case *do* improve classification accuracy, and some improvements are very significant (e.g., "glass" and "liver-disorder").

We next report performance on these data sets using the sign-based corrections and their smooth approximations as shown in Table 5. The second column gives again the neural network baseline results. The remaining columns provide accuracy using different correction terms such as the discontinuous sign, sigmoid (with $\beta = 10, 100$), arctangent, and hyperbolic tangent. Similar to the effect of correction using the KL-divergence, the sign-based corrections also show improvements for all data sets. Note that in some of the cases (sigmoid with $\beta = 10$ and tanh) there are data sets that do slightly worse than the baseline. In *all* of the sigmoid($\beta = 100$), discontinuous sign, and atan cases, improvements (sometimes quite significant) can be seen.

Table 4: UCI data accuracy using neural network classifiers and KL-divergence correction compared to our neural network baseline, and to Bayesian network classifiers from † (Friedman et al., 1997) and $*$ (Friedman et al., 1998).

DATA SET	BASELINE	KLD	NB†	TAN†	DISC TAN*
AUTO	63.27 ± 7.92	64.11 ± 7.76			76.07 ± 8.57
BALANCE-SCALE	95.48 ± 1.21	96.16 ± 1.43			74.24 ± 7.56
BREAST	90.95 ± 1.83	91.51 ± 1.94	97.36 ± 0.50	95.75 ± 1.25	96.78 ± 1.69
CHESS	97.07 ± 1.08	98.09 ± 1.04	87.15 ± 1.03	92.40 ± 0.81	
CLEVE	79.80 ± 6.19	80.14 ± 6.47	82.76 ± 1.27	79.06 ± 0.65	81.08 ± 1.34
FLARE	78.04 ± 1.37	78.14 ± 1.87	79.46 ± 1.11	82.74 ± 1.60	82.37 ± 4.19
GLASS	75.97 ± 3.12	79.89 ± 2.74	69.66 ± 1.85	69.18 ± 2.64	69.65 ± 5.58
HAYES-ROTH	77.78 ± 2.58	79.46 ± 1.36			56.25 ± 4.42
HEART	79.93 ± 4.83	80.27 ± 5.92	81.48 ± 3.26	82.96 ± 2.51	83.43 ± 5.56
HEPATITIS	80.00 ± 7.12	81.25 ± 6.84	91.25 ± 1.53	85.00 ± 2.50	91.25 ± 3.42
IONOSPHERE	86.20 ± 4.40	88.05 ± 3.64			92.30 ± 2.62
IRS	92.33 ± 1.27	94.00 ± 1.49	93.33 ± 1.05	93.33 ± 1.05	96.00 ± 2.79
LETTER	41.41 ± 2.26	42.69 ± 2.31	74.96 ± 0.61	83.44 ± 0.53	
LIVER-DISORDER	64.51 ± 1.84	66.25 ± 1.67			58.54 ± 1.94
PIMA	75.56 ± 2.97	76.57 ± 2.27	75.51 ± 1.63	75.13 ± 1.36	75.13 ± 2.82
VOTE	93.82 ± 1.17	95.26 ± 0.92	90.34 ± 1.63	89.20 ± 1.61	
WAVEFORM-21	85.58 ± 1.27	86.64 ± 1.13	77.89 ± 0.61	75.38 ± 0.63	

6 Discussion

Extending upon previous work (Bilmes et al., 2001), we have introduce a new form of (necessary) intransitive likelihood ratio classifier, one in which only necessary corrections are made to likelihood ratio based classification. This was done by using sign-based corrections to likelihood ratios. We furthermore introduced a number of continuous differentiable approximations of the sign function in order to be able to vary the inherent trade-off in the error of a Taylor's approximation.

We have applied these techniques to both a speech recognition corpus and the UCI data sets, as well as applying previous KL-divergence based corrections to the latter data. Results on the UCI data sets confirm that our techniques generalize to data sets other than speech recognition. Results show that by adding the bias term under the different approximations, the error rates can significantly decrease. This suggests that the framework could be applied to many machine learning tasks. Of all the functions we used, the shifted sigmoid has the advantage that it has a free parameter that can be tuned to best balance the aforementioned trade-off.

In future work, will attempt to utilize higher order terms in the Taylor expansion, and will apply our methodology on additional data sets. We will also further investigate the $K(P_i, P_j)$ factor in Equation 9. Furthermore, since all of these methods are intransitive, we will further analyze why intransitivity occurs and how it can potentially be utilized. In particular, we plan to develop methods for explaining, tolerating, exploiting, removing, and modeling intransitivity.

References

Bilmes, J. (1999). Burried markov models for speech recognition. *Proceedings of ICASSP 99*. Phoenix, AZ.

Bilmes, J., Ji, G., & Meilă, M. (2001). Intransitive likeilhood-ratio classifiers. *Proceedings of NIPS01*. Vancouver, BC, Canada.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. John Wiley and Sons, Inc.

Duda, R. O., & Hart, P. E. (2000). *Pattern classification and scene analysis*. New York: Wiley.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131–163.

Friedman, N., Goldszmidt, M., & Lee, T. J. (1998). Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting. *Fifteenth Inter. Conf. on Machine Learning (ICML)*.

Table 5: Accuracy on UCI data sets when applying other correction methods: (discontinuous) sign (Eq. 11), sigmoid with $\beta = 10, 100$ (Eq. 14), arctangent (Eq. 13), and hyperbolic tangent (Eq. 12).

DATA SET	BASELINE	SIGN	SIG(10)	SIG(100)	ATAN	TANH
AUTO	63.27 ± 7.92	64.74 ± 9.17	62.82 ± 9.49	64.78 ± 9.72	67.28 ± 7.92	57.24 ± 9.91
BALANCE-SCALE	95.48 ± 1.21	96.48 ± 1.21	96.48 ± 1.07	96.48 ± 1.07	96.48 ± 1.21	92.48 ± 1.55
BREAST	90.95 ± 1.83	92.98 ± 1.88	91.95 ± 1.61	91.95 ± 1.61	91.95 ± 1.84	91.95 ± 2.34
CHESS	97.07 ± 1.08	98.09 ± 1.04	98.12 ± 1.04	98.12 ± 1.04	98.09 ± 1.04	97.78 ± 1.08
CLEVE	79.78 ± 6.19	81.14 ± 5.55	80.80 ± 6.19	80.80 ± 6.19	80.80 ± 6.19	80.12 ± 4.26
FLARE	78.04 ± 1.37	80.11 ± 0.93	81.52 ± 1.09	81.52 ± 1.09	80.11 ± 0.93	81.14 ± 1.16
GLASS	75.97 ± 3.12	79.71 ± 3.53	74.20 ± 2.22	77.22 ± 2.22	77.22 ± 3.12	66.91 ± 3.22
HAYES-ROTH	77.78 ± 2.58	78.45 ± 2.76	78.79 ± 2.58	78.79 ± 2.58	78.79 ± 2.58	79.45 ± 2.90
HEART	79.93 ± 4.83	80.60 ± 6.18	80.60 ± 5.20	80.60 ± 5.20	80.27 ± 5.92	80.27 ± 6.58
HEPATITIS	80.00 ± 7.12	81.25 ± 7.65	85.00 ± 7.12	85.00 ± 7.12	81.25 ± 7.12	85.00 ± 7.65
IONOSPHERE	86.20 ± 4.40	86.63 ± 4.12	86.91 ± 4.22	86.91 ± 4.22	87.20 ± 4.40	88.06 ± 5.10
IRS	92.33 ± 1.27	94.00 ± 1.49	94.00 ± 1.49	94.00 ± 1.49	94.00 ± 1.49	93.33 ± 2.35
LETTER	41.41 ± 2.26	42.69 ± 2.31	41.80 ± 4.03	41.90 ± 5.81	42.85 ± 6.37	38.54 ± 1.65
LIVER-DISORDER	64.51 ± 1.84	66.25 ± 1.67	62.06 ± 1.92	62.28 ± 1.93	64.56 ± 1.83	62.91 ± 1.85
PIMA	75.56 ± 2.97	76.56 ± 1.56	76.30 ± 3.28	76.30 ± 3.28	76.57 ± 2.97	77.22 ± 3.18
VOTE	93.82 ± 1.17	94.83 ± 1.17	94.83 ± 1.17	94.83 ± 1.17	94.83 ± 1.17	93.55 ± 2.58
WAVEFORM-21	85.58 ± 1.27	86.28 ± 1.27	86.56 ± 1.21	86.58 ± 1.19	86.58 ± 1.27	84.26 ± 0.40

Jones, D. S. (1966). *Generalised functions*. McCraw-Hill Publishing Company Limited.

Kevorkian, J. (2000). *Partial differential equations : analytical solution techniques*. New York : Springer.

Luce, R. D., & Raiffa, H. (1957). *Games and decisions: introduction and critical survey*. Dover.

Murphy, P. M., & Aha, D. W. (1995). *UCI repository of machine learning database*.

Pitrelli, J., Fong, C., Wong, S. H., Spitz, J. R., & Lueng, H. C. (1995). Phonebook: a phonetically-rich isolated-word telephone-speech database. *Proceedings of ICASSP95*.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). *Numerical recipes in C: The art of scientific computing*. Cambridge University Press, Cambridge, England. second edition.

Rao, M. M. (1987). *Measure theory and integration*. John Wiley and Sons, Inc.

Straffin, P. D. (1993). *Game theory and strategy*. The Mathematical Association of America.