Comparison of Log-Linear Models and Weighted Dissimilarity Measures

Daniel Keysers[1], Roberto Paredes[2], Enrique Vidal[2], and Hermann Ney[1]

[1]Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen – University of Technology, D-52056 Aachen, Germany
{keysers, ney}@informatik.rwth-aachen.de

[2]Instituto Tecnológico de Informática
Departemento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, E-46022 Valencia, Spain
{rparedes, evidal}@iti.upv.es

Abstract. We compare two successful discriminative classification algorithms on three databases from the UCI and STATLOG repositories. The two approaches are the log-linear model for the class posterior probabilities and class-dependent weighted dissimilarity measures for nearest neighbor classifiers. The experiments show that the maximum entropy based log-linear classifier performs better for the equivalent of a single prototype. On the other hand, using multiple prototypes the weighted dissimilarity measures outperforms the log-linear approach. This result suggests an extension of the log-linear method to multiple prototypes.

1 Introduction

In this paper, we compare two classification algorithms that are both *discriminative*. Algorithms for classification of observations $x \in \mathbb{R}^D$ into one of the classes $k \in \{1, \ldots, K\}$ usually estimate some of their parameters in the training phase from a set of labeled training data $\{(x_n, k_n)\}$, $n = 1, \ldots, N$. The training procedure can take into account only the data from one class at a time or all of the competing classes can be considered at the same time. In the latter case the process is called discriminative. As discriminative training puts more emphasis on the decision boundaries, it often leads to better classification accuracy.

We examine the connection between two discriminative classification algorithms and compare their performance on three databases from the UCI and STATLOG repositories [5, 6].

The principle of maximum entropy is a powerful framework that can be used to estimate class posterior probabilities for pattern recognition tasks. It leads to log-linear models for the class posterior and uses the log-probability of the class posterior on the training data as training criterion. It can be shown that its combination with the use of first-order feature functions is equivalent to the discriminative training of single Gaussian densities with pooled covariance matrices [4].

The use of weighted dissimilarity measures, where the weights may depend on the dimension and class and are trained according to a discriminative criterion, has shown high performance on various classification tasks [9]. Also for this method, a strong connection to the use of Gaussian densities can be observed if one prototype per class is used. For more than one prototype per class, the similarity leads to a mixture density approach. These connections to the Gaussian classifier are used to compare the two discriminative criteria.

2 Classification Framework

To classify an observation $x \in \mathbb{R}^D$, we use the Bayesian decision rule

$$x \longmapsto r(x) = \operatorname*{argmax}_k \{p(k|x)\} = \operatorname*{argmax}_k \{p(k) \cdot p(x|k)\} .$$

Here, $p(k|x)$ is the class posterior probability of class $k \in \{1, \ldots, K\}$ given the observation x, $p(k)$ is the a priori probability, $p(x|k)$ is the class conditional probability for the observation x given class k and $r(x)$ is the decision of the classifier. This decision rule is known to be optimal with respect to the number of decision errors, if the correct distributions are known. This is generally not the case in practical situations, which means that we need to choose appropriate models for the distributions.

If we denote by Λ the set of free parameters of the distribution, the maximum likelihood approach consists in choosing the parameters $\hat{\Lambda}$ maximizing the log-likelihood on the training data:

$$\hat{\Lambda} = \operatorname*{argmax}_{\Lambda} \sum_n \log p_\Lambda(x_n|k_n) \tag{1}$$

Alternatively, we can maximize the log-likelihood of the class posteriors,

$$\hat{\Lambda} = \operatorname*{argmax}_{\Lambda} \sum_n \log p_\Lambda(k_n|x_n) \ , \tag{2}$$

which is also called discriminative training, since the information of out-of-class data is used. This criterion is often referred to as mutual information criterion in speech recognition, information theory and image object recognition [2, 8].

Discriminative training was used in [9] to learn the weights of a weighted dissimilarity measure. This weighted measure was used in the nearest neighbor classification rule improving significantly the accuracy of the classifier in comparison to other distance measures, for which the parameters were not estimated using discriminative training.

3 Maximum Entropy, Gaussian and Log-Linear Models

The principle of maximum entropy has origins in statistical thermodynamics, is related to information theory and has been applied to pattern recognition tasks

such as language modeling [1] and text classification [7]. Applied to classification, the basic idea is the following: We are given information about a probability distribution by samples from that distribution (training data). Now, we choose the distribution such that it fulfills all the constraints given by that information (more precisely: the observed marginal distributions), but otherwise has the highest possible entropy. (This inherently serves as regularization to avoid overfitting.) It can be shown that this approach leads to log-linear models for the distribution to be estimated.

Consider a set of so-called feature functions $\{f_i\}, i = 1, \ldots, I$ that are supposed to compute 'useful' information for classification:

$$f_i \quad : \quad \mathbb{R}^D \times \{1, \ldots, K\} \longrightarrow \mathbb{R} \quad : \quad (x, k) \longmapsto f_i(x, k)$$

It can be shown that the resulting distribution that maximizes the entropy has the following log-linear or exponential functional form:

$$p_\Lambda(k|x) = \frac{\exp\left[\sum_i \lambda_i f_i(x, k)\right]}{\sum_{k'} \exp\left[\sum_i \lambda_i f_i(x, k')\right]}, \quad \Lambda = \{\lambda_i\}. \tag{3}$$

Interestingly, it can also be shown that the stated optimization problem is convex and has a unique global maximum. Furthermore, this unique solution is also the solution to the following dual problem: Maximize the log probability (2) on the training data using the model (3).

A second desirable property of the discussed model is that effective algorithms are known that compute the global maximum of the log probability (2) given a training set. These algorithms fall into two categories: On the one hand, we have an algorithm known as generalized iterative scaling [3] and related algorithms that can be proven to converge to the global maximum. On the other hand, due to the convex nature of the criterion (2), we can also use general optimization strategies as e.g. conjugate gradient methods.

The crucial problem in maximum entropy modeling is the choice of the appropriate feature functions $\{f_i\}$. In general, these functions depend on the classification task considered.

The straight forward way to define feature functions for classification purposes is to directly use the features provided for the specific task. Consider therefore the following first-order feature functions for log-linear classification:

$$f_{k,i}(x, k') = \delta(k, k')\, x_i \ ,$$
$$f_k(x, k') = \delta(k, k') \ ,$$

where $\delta(k, k') := 1$ if $k = k'$, and 0 otherwise denotes the Kronecker delta function. The Kronecker delta is necessary here to distinguish between the different classes. It can be shown that maximum entropy training using first-order features is equivalent to the discriminative training of single Gaussian densities with globally pooled covariance matrices using the criterion (2) [4]. Furthermore, we may also consider products of feature values for the feature functions (second-order features) by including

$$f_{k,i,j}(x, k') = \delta(k, k')\, x_i x_j \ , \quad i \geq j \ .$$

In this case, the maximum entropy training is equivalent to the discriminative training of single Gaussian densities with full, class-specific covariance matrices, where the constraint on the covariance matrices to be positive (semi-) definite is relaxed [4]. The correspondences can be derived by observing that the functional form of the class posterior

$$p(k|x) = \frac{p(k)\,\mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{k'} p(k')\,\mathcal{N}(x|\mu_{k'}\Sigma_{k'})}$$

also leads to a log-linear expression like (3) for the appropriate choice of feature functions. These correspondences to Gaussian models with one prototype justify the classification of the log-linear approach to be a 'one-prototype' approach.

4 Class-Dependent Weighted Dissimilarity Measures

In [9], a *class-dependent* weighted dissimilarity measure for nearest neighbor classifiers was introduced. The squared distance is defined as

$$d^2(x, \mu) = \sum_d \left(\frac{x_d - \mu_d}{\sigma_{k_\mu d}}\right)^2 \quad , \; \Lambda = \{\sigma_{kd}, \mu_d\},$$

where d denotes the dimension index and k_μ is the class the reference vector μ belongs to. The parameters Λ are estimated with respect to a discriminative training criterion that takes into account the out-of-class information and can be derived from the minimum classification error criterion:

$$\hat{\Lambda} = \operatorname*{argmin}_{\Lambda} \sum_n \frac{\min\limits_{\mu:k_\mu = k_n} d_\Lambda(x_n, \mu)}{\min\limits_{\mu:k_\mu \neq k_n} d_\Lambda(x_n, \mu)} \tag{4}$$

In other words, the parameters are chosen to minimize the average ratio of the distance to the closest prototype of the same class with respect to the distance to the closest prototype of the competing classes.

To minimize the criterion, a gradient descent approach is used and a leaving one out estimation with the weighted measure is computed at each step of the gradient procedure. The weights selected by the algorithm are those weights with the best leaving one out estimation instead of the weights with the minimum criterion value. In the experiments, only the weights $\{\sigma_{kd}\}$ were estimated according to the proposed criterion. The references $\{\mu_k\}$ were chosen as the means for the one-prototype approach and in the multiple-prototype approach the whole training set was used.

Also in this approach, we have a strong relation to Gaussian models. Consider the use of one prototype per class. The distance measure then is a *class-dependent* Mahalanobis distance with class-specific, diagonal covariance matrices

$$\Sigma_k = \operatorname{diag}(\sigma_{k1}^2, \ldots, \sigma_{kD}^2).$$

Table 1. Corpus statistics for the three databases used in the experiments from the UCI and STATLOG repositories, respectively.

corpus name	MONK	DNA	LETTER
# classes	2	3	26
# features	17	180	16
# training samples	124	2 000	15 000
# test samples	432	1 186	5 000

The decision rule is then equivalent to the use of single Gaussian models in combination with an additional factor to compensate for the missing normalization factor of the Gaussian. In the case of multiple prototypes per class, the equivalence is extensible to mixtures of Gaussian densities.

5 Connection between the two Classifiers

As discussed in the two previous sections, the two approaches are equivalent to the use of discriminative training for single Gaussian densities with some additional restrictions. This implies that the main difference between the classifiers is the criterion that is used to choose the class boundaries:

Gaussian densities: criterion: maximum likelihood (1); decision boundary: linear (pooled covariance matrices) or quadratic (class-specific covariance matrices)

log-linear model: criterion: maximum mutual information (maximum likelihood of the posterior) (2); decision boundary: linear (first-order feature functions) or quadratic (second-order feature functions)

weighted dissimilarity measures: criterion: intra-class distances versus inter-class distances (4); decision boundary: quadratic (one prototype per class) or piecewise quadratic (multiple prototypes per class)

6 Databases and Results

The experiments were performed on three corpora from the UCI and STATLOG database, respectively [5, 6]. The corpora were chosen to cover different properties with respect to the number of classes and features and with respect to the size. The statistics of the corpora are summarized in Table 1. MONK is an artificial decision task with categorical features also known as the monk's problem. For the experiments, the categorical features were transformed into binary features. For the DNA task, the goal is to detect gene intron/exon and exon/intron boundaries given part of a DNA sequence. Also for this task, the categorical features were transformed into binary features. Finally, the LETTER corpus consists of printed characters that were preprocessed and a variety of different features was extracted.

Table 2 shows a summary of the results obtained with the two methods. The figures show the following tendencies:

Table 2. Experimental results for the three databases used with different settings of the algorithms given as error rate (er) in %. The number of parameters (#param.) refers to the total number of parameters needed to completely define the classifier.

	MONK		DNA		LETTER	
method	er[%]	#param.	er[%]	#param.	er[%]	#param.
single Gaussian	28.5	51	9.5	720	41.6	432
log-linear, first-order	28.9	36	5.6	543	22.5	442
second-order	0.2	308	5.1	48 873	13.5	3 562
weighted dissimil., one prot.	16.7	68	6,7	1 080	24.1	832
multiple prot.	0.0	2 142	4.7	360 540	3.3	240 416
best other [5, 6]	0.0	-	4.1	-	3.4	-

- Considering the four approaches that can be labeled 'one-prototype' (single Gaussian, both log-linear models and the one-prototype weighted dissimilarity measure), the discriminative approaches generally perform better than the maximum likelihood based approach (single Gaussian).
- For the two log-linear approaches, the second-order features perform better than the first-order features.
- On two of the three corpora, the log-linear classifier with first-order features performs better than the one-prototype weighted dissimilarity measure using a smaller number of parameters.
- On all of the corpora, the log-linear classifier with second-order features performs better than the one-prototype weighted dissimilarity measure, but using a larger number of parameters.
- The weighted dissimilarity measures using multiple prototypes outperforms the other regarded ('one-prototype') approaches on all tasks and is competitive with respect to the best known results on each task.

Note that second-order features perform better here although estimation of full, class-specific covariance matrices is problematic for many tasks. This indicates a high robustness of the maximum entropy log-linear approach. Note further that both the one-prototype weighted dissimilarity classifier and the log-linear model with second-order features lead to quadratic decision boundaries, but the former does not take into account bilinear terms of the features, which is the case for the second-order features.

The high error rate of the log-linear model with first-order features on the MONK corpus was analyzed in more detail. As this task only contains binary features, also the one-prototype weighted dissimilarity classifier leads to linear decision boundaries here ($x^2 = x \Leftrightarrow x \in \{0, 1\}$). Therefore it is possible to infer the parameters for the log-linear model from the training result of the weighted dissimilarity classifier. This showed that the log-likelihood of the posterior (2) on the training data is lower than that resulting from maximum entropy training , which is not surprising as exactly this quantity is the training criterion for the log-linear model. But interestingly the same result holds for the *test* data as well. That is, the maximum entropy training result has higher prediction

accuracy on the average for the class posterior, but this does not result in better classification accuracy. This may indicate that on this corpus with very few samples the weighted dissimilarity technique is able to better adapt the decision boundary as it uses a criterion derived from the minimum classification error criterion.

7 Conclusion

A detailed comparison of two discriminative algorithms on three corpora with different characteristics has been presented. The discriminative approaches generally perform better than the maximum likelihood based approach.

A direct transfer of the maximum entropy framework to multiple prototypes is difficult, as the use of multiple prototypes leads to nonlinearities and the log-linear model cannot be directly applied any more.

The consistent improvements obtained with weighted dissimilarity measures and multiple prototypes in combination with the improvements obtained by using second-order features suggest possible improvements that could be expected from a combination of these two approaches.

References

1. A.L. Berger, S.A. Della Pietra, and V.J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March 1996.
2. J. Dahmen, R. Schlüter, and H. Ney. Discriminative training of Gaussian mixture densities for image object recognition. In *21. DAGM Symposium Mustererkennung*, pages 205–212, Bonn, Germany, September 1999.
3. J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.
4. D. Keysers, F.J. Och, and H. Ney. Maximum entropy and Gaussian models for image object recognition. In *Pattern Recognition, 24th DAGM Symposium*, pages 498–506, Zürich, Switzerland, September 2002.
5. C.J. Merz, P.M. Murphy, and D.W. Aha. *UCI Repository of Machine Learning Databases*. http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Department of Information and Computer Science, Irvine CA, 1997.
6. D. Michie, D.J. Spiegelhalter, C.C. Taylor (eds). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994. Available at http://www.amsta.leeds.ac.uk/∼charles/statlog/, datasets at http://www.liacc.up.pt/ML/statlog/datasets.html.
7. K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, Stockholm, Sweden, August 1999.
8. Y. Normandin. Maximum mutual information estimation of hidden Markov models. In C.H. Lee, F.K. Soong, and K.K. Paliwal, editors, *Automatic Speech and Speaker Recognition*, pages 57–81, Norwell, MA, Kluwer, 1996.
9. R. Paredes and E. Vidal. A class-dependent weighted dissimilarity measure for nearest-neighbor classification problems. *Pattern Recognition Letters*, 21(12):1027–1036, November 2000.