Mixtures of Conditional Maximum Entropy Models

Dmitry Pavlov[*] PAVLOVD@ICS.UCI.EDU
Yahoo! Inc., 701 First Avenue, Sunnyvale, CA 94089 USA

Alexandrin Popescul POPESCUL@CIS.UPENN.EDU
Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

David M. Pennock[*] DAVID.PENNOCK@OVERTURE.COM
Overture Services Inc., 74 N. Pasadena Ave., 3rd floor, Pasadena, CA 91103 USA

Lyle H. Ungar UNGAR@CIS.UPENN.EDU
Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

Abstract

Driven by successes in several application areas, maximum entropy modeling has recently gained considerable popularity. We generalize the standard maximum entropy formulation of classification problems to better handle the case where complex data distributions arise from a mixture of simpler underlying (latent) distributions. We develop a theoretical framework for characterizing data as a *mixture of maximum entropy models*. We formulate a maximum-likelihood interpretation of the mixture model learning, and derive a generalized EM algorithm to solve the corresponding optimization problem. We present empirical results for a number of data sets showing that modeling the data as a mixture of latent maximum entropy models gives significant improvement over the standard, single component, maximum entropy approach.

Keywords: Mixture model, maximum entropy, latent structure, classification.

1. Introduction

Maximum entropy (maxent) modeling has a long history, beginning as a concept in physics and later working its way into the foundations of information theory and Bayesian statistics (Jaynes, 1979). In recent years, advances in computing and the growth of available data contributed to increased popularity of maxent modeling, leading to a number of successful applications, including natural language processing (Berger et al., 1996), language modeling (Chen & Rosenfeld, 1999), part of speech tagging (Ratnaparkhi, 1996), database querying (Pavlov & Smyth, 2001), and protein modeling (Buehler & Ungar, 2001), to name a few. The maxent approach has several attractive properties that have contributed to its popularity. The method is semi-parametric, meaning that the learned distribution can take on any form that adheres to the constraints. In this way, maxent modeling is able to combine sparse local information encoded in the constraints into a coherent global probabilistic model, without *a priori* assuming any particular distributional form. The method is capable of combining heterogeneous and overlapping sources of information. Under fairly general assumptions, maxent modeling has been shown to be equivalent to maximum-likelihood modeling of distributions from the exponential family (Della Pietra et al., 1997).

One of the more recent and successful applications of maxent modeling is in the area of classification (Jaakkola et al., 1999), and text classification in particular (Nigam et al., 1999). In this case, conditional maxent distributions (i.e., probabilities of the class labels given feature values) are learned from the training data and then used to automatically classify future feature vectors for which class membership is unknown.

Note that being maximally noncommittal can sometimes be a hindrance in cases where exploitable hidden

[*] Work conducted at NEC Laboratories America, 4 Independence Way, Princeton, NJ 08540 USA.

structure exists in the data beyond the expressed constraints. Many data sets with seemingly complex distributional structures can be seen as generated by several simpler latent distributions that are not directly observable. As an example, the distribution of text in a broad document collection may have a complex joint structure, but if broken up into meaningful topics, may be well modeled as a mixture of simpler topic-specific distributions. *Mixture models* (McLachlan & Basford, 1988) are designed to handle just such a case, where it is assumed the full distribution is composed of simpler components. In a sense, discovering the underlying structure in a data set can be thought of as an unsupervised learning subtask within a larger supervised learning problem.

In this paper, we generalize the maxent formalism to handle *mixtures of maxent models.* In cases where data can be decomposed into latent clusters, our framework leverages this extra structural information to produce models with higher out-of-sample log-likelihood and higher expected classification accuracy. We formulate a maximum-likelihood interpretation of the mixture model learning, and derive a generalized EM (GEM) algorithm (Dempster et al., 1977) to solve the corresponding optimization problem. We present empirical results on several publicly available data sets showing significant improvements over the standard maxent approach. On the data sets tested, our mixture technique never performs worse than standard maxent (within noise tolerance), and often performs significantly better.

In contrast to numerous dimensionality reduction techniques employed in supervised learning, which can be regarded as techniques exploiting latent structure in the space of features, the mixtures of conditional maximum entropy models directly exploit the latent structure in the space of original examples (e.g. documents, not words, in text classification).

The rest of the paper is organized as follows. In Section 2 we discuss some of the related work. In Section 3 we review the definition of a standard maxent model. Section 4 presents a definition of the mixture of maxent models and the main update equations for the GEM algorithm used to fit the mixture. Experimental results are discussed in Section 5. In Section 6 we draw conclusions and describe directions for future work.

2. Related Work

The latent maximum entropy principle was introduced in a general setting by Wang et al. (2002). In particular, they gave a motivation for generalizing the standard Jaynes maximum entropy principle (Jaynes, 1979) to include latent variables and formulated a convergence theorem of the associated EM algorithm. In this paper, we present a derivation of the EM algorithm for a specific mixture model latent structure as well as describe and discuss empirical results of such an approach.

Modeling of the latent structure in document space was previously employed in a classification setting by Nigam et al. (2000, unpublished commercial project), where *distributional clustering* (Baker & McCallum, 1998; Pereira et al., 1993) and maxent modeling were combined to improve document classification accuracy. The fundamental distinction between their approach and ours is that our latent structure mixture modeling is fully integrated into an EM algorithm designed to maximize a single objective function.

The generalized linear mixed models (Wolfinger & O'Connell, 1993) widely used in marketing research are similar to our approach. However, our model mixture components are non-linear, and as such, potentially more powerful. For binary classification problems the conditional maximum entropy models can be shown to be equivalent to logistic regression models, however for multi-class problems such relationship does not hold. Mixtures of multinomial logistic regression models have also been studied in the past (McFadden & Train, 1997; David & Kenneth, 1998).

3. Conditional Maxent Model

Consider a problem of estimating the distribution $p(c|d)$ of a discrete-valued class variable c for a given vector of observations d in the presence of constraints on the distribution. To define constraints, we represent each vector d as a set of (in general real-valued) *features*. Typically, we allow each class to be characterized by a separate set of features. For a given vector of observations d and class label c, we set to 0 all features from classes other than c. A formal definition of the feature s in class c' is as follows (Nigam et al., 1999):

$$F_{s,c'}(c,d) \quad = \quad \begin{cases} 0 & \text{if } c \neq c' \\ d_s & \text{otherwise,} \end{cases}$$

where d_s is the value of the s-th component of the vector d. For example, in a text classification task, d is a document, s could be the word "*surgery*", d_s—a frequency of the word "*surgery*" in the document d and $c' =$ "*Medicine*" the class label for d. In what follows, we omit c' to simplify the notation but emphasize that there might be a separate set of features for each class.

A constraint based on the feature F_s prescribes that the empirically observed frequency of this feature should be equal to its expectation with respect to the model $p(c|d)$:

$$\sum_d \sum_c p(c|d) F_s(c,d) \quad = \quad \sum_d F_s(c(d),d), \qquad (1)$$

where $s = 1, \ldots, S$ runs across all features, and $c(d)$ is the class label of the vector d. The left-hand side of Equation 1 represents the expectation (up to a normalization factor) of the feature $F_s(c,d)$ with respect to the distribution $p(c|d)$ and the right-hand side is the expected value (up to the same normalization factor) of this feature in the training data.

The set of features supplied with maximum entropy as an objective function can be shown to lead to the following form of the conditional maxent model (Jelinek, 1998)

$$p(c|d) \quad = \quad \frac{1}{Z_\lambda(d)} exp[\sum_{s=1}^{S} \lambda_{sc} F_s(c,d)], \qquad (2)$$

where $Z_\lambda(d)$ is a normalization constant ensuring that the distribution sums to 1. In what follows we drop the subscript in Z_λ to simplify notation.

There exist efficient algorithms for finding the parameters $\{\lambda\}$ from the set of Equations 1 (e.g., generalized iterative scaling (Darroch & Ratcliff, 1972) or improved iterative scaling (Della Pietra et al., 1997)).

4. Mixture of Conditional Maxent Models

As we pointed out above, it might be advantageous to assume that the data points are generated from a set of K clusters, with each cluster described by its own distribution:

$$p(c|d) \quad = \quad \sum_{k=1}^{K} p(c|d,k)\alpha_k, \qquad (3)$$

where $\alpha_k = p(k)$ is a prior probability of cluster k, $\sum_k \alpha_k = 1$ and for each $k = 1, \ldots, K$, $p(c|d,k)$ has a maximum entropy form

$$p(c|d,k) \quad = \quad \frac{1}{Z_k(d)} exp[\sum_{s=1}^{S} \lambda_{sck} F_s(c,d)].$$

We derive the generalized EM algorithm for finding parameters λ and α in Appendix A. Here we present the update equations for the maximum likelihood estimates of parameter values. The treatment of the MAP estimates (obtained by imposing a Gaussian prior (Chen & Rosenfeld, 1999)) is given in Appendix A.

In the E-step, we find the posterior distribution over the clusters:

$$P_k \triangleq P(cluster = k|c(d),d) \quad = \quad \frac{p(c(d)|d,k)\alpha_k}{\sum_{k=1}^{K} p(c(d)|d,k)\alpha_k}.$$

In the M-step, we maximize the likelihood by finding the new values of parameters using the cluster memberships obtained in the E-step:

$$\begin{aligned} \alpha_k^{new} &= \frac{1}{|D|} \sum_{d \in D} P(cluster = k|c(d),d); \\ \delta_{s'c'k'} &= \sum_{d \in D} P_{k'} F_{s'}(c',d)[I(c(d) = c') - p(c'|d,k')]; \\ \lambda_{s'c'k'}^{new} &= \lambda_{s'c'k'}^{old} + \epsilon \delta_{s'c'k'}, \end{aligned}$$

where ϵ is a small step in the direction of the gradient of the log-likelihood, ensuring that the likelihood increases, and $I()$ is the indicator function. As we discuss in Appendix A, finding exact values of parameters λ that maximize the likelihood is difficult since it requires solving the system of non-linear equations. However, for the GEM algorithm to converge, it is sufficient that the likelihood only increases in the M-step (McLachlan & Krishnan, 1997). We employ this form of the generalized EM algorithm and do a single step of the gradient ascent for parameters λ in the M-step.

The worst-case time complexity of the algorithm per iteration in an N_C-class problem is $O(KSN_C|D|)$. The worst-case is achieved on the computation of $\delta_{s'c'k'}$, and a straightforward speed-up can be gained by noticing that some of the feature values $F_{s'}(c',d)$ can be equal to 0, thus making the corresponding terms on the right-hand side of the update equation for $\delta_{s'c'k'}$ vanish. As we show in Section 5, on sparse data the improvements can be quite significant. Further speedups might be achieved by employing the recent work by (Goodman, 2002), though we have not explored this direction at present.

5. Experimental Results

We ran experiments on several publicly available data sets. The names and parameters of the data sets are given in Table 1. Among the parameters we report the number of classes N_C, the number of features S and the number of data records $|D|$. Note that as we mentioned in the end of the previous section, the product of

Table 1. Parameters of the data sets used in experiments. S is the number of features (attributes), $|D|$ is the number of training records, N_C is the number of classes, $SN_C|D|$ is the product of the 3 previous columns and represents the major factor in the time complexity (reported on the log_{10} scale) and *Sparsity* is the percent of data entries with 0 values (out of total $|D| \otimes S$). Experiments on the WebKB data were conducted with subsets of attributes, containing 50, 200, 500, 1000 most frequent attributes; sparsity index for each subset is reported.

Name	S	$\|D\|$	N_C	$\log_{10}(SN_C\|D\|)$	*Sparsity*, %
WebKB	50	1919	6	5.76	41.51
WebKB	200	1919	6	6.36	65.99
WebKB	500	1919	6	6.76	80.60
WebKB	1000	1919	6	7.06	87.74
Letter recognition	16	10000	26	6.62	2.61
Yeast	8	732	10	4.76	12.43
MS Web	294	16000	2	6.97	99.08
Vehicle	18	593	4	4.63	1.18
Vowel	11	726	11	4.94	3.39
Cover	54	11340	7	6.63	77.99
Segmentation	19	1155	7	5.18	10.79

these three quantities factors in along with the number of mixture components into the worst-case time complexity of the algorithm. We report the logarithm of the product in the fifth column to show the anticipated order of magnitude of the complexity. The last column shows the *sparsity* of the data set that also affects the time complexity. If one imagines the data organized as a matrix with S columns corresponding to features and $|D|$ rows corresponding to data records, then *sparsity* reports the percentage of 0 entries in this matrix. As we mentioned above, the higher the sparsity, the more time-efficient the algorithm is.

The **WebKB** data (Craven et al., 1998) contains a set of Web pages gathered from university computer science departments. We used all classes but **others** and different numbers (up to 1000) of the most frequent words. The **Letter recognition**, **Yeast**, **MS Web**, **Vehicle** and **Vowel** data sets were downloaded from the UC Irvine machine learning repository (Blake & Merz, 1998). In the **MS Web** data set, we predicted whether a user visited the "free downloads" web page, given the rest of his navigation on `microsoft.com`. For all remaining data sets, we solved the classification task as posted on the UC Irvine web site.

For all data sets, we experimented with 1, 3, 5, 7, 9, 11, 13 and 15 components.[1] We split the data into three sets: training data, held-out data, and test data. Held-out data was used to determine when to stop training, and to choose the best number of components. When training the mixture model we stopped the GEM algorithm when the relative increase in the log-likelihood on held-out data became less than 0.0005. We made five random restarts of GEM to reduce the influence of starting point initialization.[2] The best model for each of the starts was selected based on the classification performance on the held-out data. Neither the mixture model nor the standard maximum entropy model were smoothed. Performance statistics measured include the classification accuracy and the log-likelihood on the test set, and the time taken to learn the model. For smaller **Vehicle** and **Vowel** data sets we also performed respectively 10 and 15 fold cross-validation and averaged the results.

Table 2 reports results for the WebKB data set. The first column shows the number of top most frequent attributes used for training; the remaining columns are labeled with the number of mixture components. Within each row block of the table, "A" reports classification accuracy on the test data, "L" the log-likelihood on the test data and "T" the time taken to train the model. The boxed number represents the classification accuracy of the best model selected based on the held-out data.

Notice that the classification accuracy of the best mixture model (boxed) is better than the accuracy of the standard maxent across all selected attribute subsets. However, as the size of the attribute subset increases not only the accuracy of the models increases but also the improvement provided by the mixture becomes smaller. Notice also that for larger attribute subset sizes, the log-likelihood scores of the mixture model are

[1] We used the same algorithm to fit one component models.

[2] Note that unlike the optimization problems for fitting the standard (one component) maxent, the likelihood surface for mixture may have several local maxima.

Table 2. WebKB data set: Performance of the mixture of maxent models on 3, 5, 7, 9, 11, 13 and 15 components compared to the standard (1-component) maxent model. "A" stands for accuracy on the test data, "L" for log-likelihood score on the test data and "T" for time taken to learn the model (in seconds). Different row blocks in the table correspond to selecting the top 50, 200, 500 and 1000 attributes in the data set with respect to their frequency. The boxed number is the classification accuracy of the best mixture model selected according to the held-out data.

N. Attributes		1	3	5	7	9	11	13	15
	A	49.69	53.34	$\boxed{51.69}$	53.34	53.08	52.56	51.95	52.38
50	L	-1.383	-1.408	-1.375	-1.358	-1.395	-1.383	-1.374	-1.366
	T	55	161	219	341	481	570	597	840
	A	68.05	69.87	69.87	70.57	69.79	$\boxed{71.01}$	70.74	70.65
200	L	-0.947	-0.938	-0.954	-0.961	-0.904	-0.958	-0.927	-0.909
	T	78	305	480	485	677	1380	1205	1750
	A	80.81	$\boxed{81.51}$	79.94	80.03	80.20	80.73	80.47	79.86
500	L	-0.686	-0.691	-0.664	-0.698	-0.679	-0.685	-0.665	-0.674
	T	151	388	692	804	1254	1474	2157	2100
	A	81.85	81.85	81.68	$\boxed{82.03}$	82.03	82.11	82.29	82.20
1000	L	-0.689	-0.669	-0.663	-0.669	-0.633	-0.637	-0.624	-0.628
	T	225	723	1411	1723	2122	2873	3382	4038

typically slightly better than for the standard maxent model; however, this does not necessarily translate into improvement in classification. We have also observed a similar phenomenon on several other data sets, results for which we report below.

The time taken to train the mixture grows roughly linearly with the number of mixture components. However, the times are still manageable, and as our experiments suggest, uncovering potential mixture structure and obtaining the improvement in classification, could well be worth spending the extra time. Furthermore, one could employ recent advances in speeding up maximum entropy learning (Goodman, 2002) to alleviate the complexity associated with the learning time. Recall that in Table 1 we demonstrated that the sparsity of the WebKB data increases with the growth of the size of the attribute set used in learning. Table 2 in turn shows that sparse data often leads to sublinear complexity growth. For instance, one might expect the time complexity of fitting a three component mixture on 200 attributes to be roughly 4 times higher than on 50 attributes; however, the actual number is roughly twice as high (305 seconds for 200 attributes versus 161 seconds for 50 attributes), due to the inherent sparsity of the data and our ability to take advantage of it.

In Table 3 we present results similar to that of Table 2 on the data sets other than WebKB. Again, we can clearly see the improvement provided by the mixture in comparison with the standard maxent both in the log-likelihood scores and the classification accuracy on the test data. This suggests that in most cases the mixture model is a more adequate model for the data than the standard maxent model since the former does a better job capturing the structure contained in the data. The improvement varies depending on the data set and for the classification accuracy ranges from fractions of percent on the Vehicle data set to almost 9 percent on the Vowel data set. The average accuracy improvement of selected models (boxed values in Tables 2 and 3) over the one-component model is 2.94%. The 95% confidence interval of improvement percentage for Table 3 according to a statistical t-test is $[0.10\%, 5.78\%]$, so the improvement we observe is significant at a greater than 0.95 confidence level according to this test.[3] We have focused solely on presenting the improvements resulting from the introduction of mixtures over a single component maxent model. A common conclusion in large-scale comparison studies, e.g. (King et al., 1995; Lim et al., 2000), is that there is no single best algorithm across different datasets; their relative merits depend on the characteristics of a particular dataset. The same studies report that logistic regression, which is equivalent to maxent in binary classification case, is often quite successful. We are not aware of reported comparisons between maxent and (polytomous) logistic regression for more general multi-class problems.

This set of experiments also confirms our previous observation that the actual time complexity strongly depends on the sparsity of the data. By looking only at the complexity terms of Table 1, one could expect

[3] We also reported a probabilistic measure of generalization performance—log-likelihood of the test data; its average improvement over one-component models is also significant at a greater than 0.95 confidence level.

Table 3. Performance of the mixture of maxent models on 3, 5, 7, 9, 11, 13 and 15 components compared to the standard (1-component) maxent model on various data sets from the UCI repository. "A" stands for accuracy on the test data, "L" for log-likelihood score on the test data and "T" for time taken to learn the model (in seconds). The boxed number is the classification accuracy of the best mixture model selected according to the held-out data.

Name		1	3	5	7	9	11	13	15
Letter recognition	A	72.42	74.65	$\boxed{76.62}$	76.47	76.45	76.35	76.07	—
	L	-1.069	-0.974	-0.862	-0.868	-0.840	-0.870	-0.914	—
	T	3004	11992	20800	29540	40801	45913	—	—
Yeast	A	51.67	$\boxed{54.00}$	53.33	50.00	50.67	52.00	55.67	54.00
	L	-1.264	-1.228	-1.248	-1.280	-1.256	-1.268	-1.237	-1.255
	T	33	76	128	188	259	310	376	434
MS Web	A	72.37	75.09	$\boxed{75.62}$	75.44	75.62	75.61	75.38	75.73
	L	-0.528	-0.504	-0.492	-0.491	-0.488	-0.487	-0.490	-0.485
	T	25	126	233	239	358	423	538	585
Vehicle	A	71.11	70.65	71.47	$\boxed{71.35}$	71.59	71.24	70.76	71.01
	L	-0.771	-0.767	-0.719	-0.736	-0.748	-0.754	-0.749	-0.743
	T	15	30	51	67	89	91	122	144
Vowel	A	43.03	49.89	52.12	49.29	51.31	$\boxed{52.02}$	49.39	50.10
	L	-1.665	-1.448	-1.465	-1.460	-1.471	-1.418	-1.451	-1.485
	T	9	49	80	83	105	116	113	240
Cover	A	57.55	57.23	56.79	56.88	57.46	57.97	57.99	$\boxed{59.13}$
	L	-1.037	-1.016	-0.993	-1.021	-0.982	-0.984	-0.983	-0.957
	T	575	637	1563	1845	2410	3542	3758	4554
Segmentation	A	$\boxed{89.75}$	90.18	90.18	89.89	90.47	89.89	89.89	90.33
	L	-0.298	-0.291	-0.282	-0.287	-0.289	-0.288	-0.295	-0.293
	T	63	137	245	276	345	455	588	595

that time performance on the *Letter Recognition* and *Cover* data sets would be roughly the same. However, the *Cover* data set is substantially more sparse and this results in an order of magnitude decrease in actual training time difference.

Overall, we conclude that the mixture of maximum entropy models provides a valuable modeling tool with a power exceeding that of the regular maxent. The mixture is capable of better capturing the underlying latent structure of the data if such a structure exists. The increased modeling power comes at the expense of higher time needed to fit the model. The actual CPU times in our experiments are still manageable and can further be reduced by employing recently published speed-up techniques for maximum entropy (Goodman, 2002).

6. Conclusions and Future Work

We presented a methodology for classification that exploits the latent structure in the data using a mixture of maxent models. We defined a mixture of maximum entropy models and derived a generalized EM algorithm for solving the corresponding optimization problem. Our experiments on several publicly available data sets suggest that the mixture of maxent models can provide a significant improvement over the standard maximum entropy model. We also presented update equations for the GEM algorithm for the case of the mixture of maximum entropy models smoothed with a Gaussian prior.

The idea of employing the mixture of maximum entropy models to uncover and exploit the latent structure in the data can be easily generalized to other domains, such as sequence prediction (Pavlov & Pennock, 2002), chemical naming, Internet user disambiguation and others.

References

Baker, L. D., & McCallum, A. (1998). Distributional clustering of words for text classification. *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR-98)* (pp. 96–103). ACM Press, New York.

Berger, A., Della Pietra, S., & Della Pietra, V. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics, 22*, 39–72.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

Buehler, E. C., & Ungar, L. H. (2001). Maximum entropy methods for biological sequence modeling. *BIOKDD* (pp. 60–64).

Chen, S., & Rosenfeld, R. (1999). *A Gaussian prior for smoothing maximum entropy models* (Technical Report CMUCS-99-108). Carnegie Mellon University.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of 15th Conference of the American Association for Artificial Intelligence (AAAI-98)* (pp. 509–516). AAAI Press, Menlo Park.

Darroch, J. N., & Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics, 43*, 1470–1480.

David, R., & Kenneth, T. (1998). Mixed logit with repeated choices: Households' choices of appliance efficiency level. *Review of Economics and Statistics, 80*, 1—11.

Della Pietra, S., Della Pietra, V., & Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, 380–393.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1—38.

Goodman, J. (2002). Sequential conditional generalized iterative scaling. *Association for Computational Linguistics Annual Meeting.*

Jaakkola, T., Meila, M., & Jebara, T. (1999). *Maximum entropy discrimination* (Technical Report MIT AITR-1668).

Jaynes, E. T. (1979). Where do we stand on maximum entropy? *The Maximum Entropy Formalism* (pp. 15—118). Cambridge MA: MIT Press.

Jelinek, F. (1998). *Statistical methods for speech recognition.* Cambridge. MA:MIT Press.

King, R., Feng, C., & Sutherland, A. (1995). StatLog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence, 9(3)*, 289–333.

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning, 40*, 203–228.

McFadden, D., & Train, K. (1997). *Mixed MNL models for discrete response* (Technical Report Department of Economics, UC Berkeley.).

McLachlan, G., & Basford, K. (1988). *Mixture models.* Marcel Dekker, New York.

McLachlan, G., & Krishnan, T. (1997). *The EM algorithm and extensions.* John Wiley and Sons, New York.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using maximum entropy for text classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering* (pp. 61–67).

Nigam, K., Popescul, A., & McCallum, A. (2000, unpublished commercial project). Using latent structure of a document collection to improve text classification. *Whizbang! Labs, Pittsburgh.*

Pavlov, D., & Pennock, D. (2002). A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. *Proceedings of Neural Information Processing Systems (NIPS-2002), to appear.*

Pavlov, D., & Smyth, P. (2001). Probabilistic query models for transaction data. *Proceedings of Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 164–173). New York, NY: ACM Press.

Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of English words. *Meeting of the Association for Computational Linguistics* (pp. 183–190).

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 133–142. Somerset, New Jersey: Association for Computational Linguistics.

Wang, S., Rosenfeld, R., Zhao, Y., & Shuurmans, D. (2002). The latent maximum entropy principle. *IEEE International Symposium on Information Theory (ISIT).*

Wolfinger, R., & O'Connell, M. (1993). Generalized linear mixed models: A pseudo-likelihood approach. *Journal of Statistical Computation and Simulation, 48*, 233—243.

A. Appendix: EM algorithm for mixture of maxent models

The log-likelihood L of the training data D generated by N_c classes, each represented by n_i, $i = 1, \ldots, N$ vectors of observations, is

$$L \triangleq \log p(D|\{\lambda\}, \{\alpha\}) = \sum_{d \in D} \log p(c(d)|d), \quad (4)$$

where $p(c(d)|d)$ is given by Equation 3.

Assuming for simplicity for now that there are no priors on parameters $\{\lambda\}$ and $\{\alpha\}$, the objective is to maximize the log-likelihood in Equation 4 subject to the constraint $\sum_k \alpha_k = 1$.

Setting up the Lagrange function and differentiating it with respect to $\lambda_{s'c'k'}$ yields the following:

$$\frac{\partial L}{\partial \lambda_{s'c'k'}} = \sum_{d \in D} \frac{\frac{\partial \alpha_{k'} p(c(d)|d,k')}{\partial \lambda_{s'c'k'}}}{\sum_{k=1}^{K} \alpha_k p(c(d)|d,k)}. \quad (5)$$

A standard trick in setting up the EM procedure is to introduce the posterior distribution over the clusters, i.e. define

$$P_k \triangleq P(cluster = k|c(d), d) = \frac{p(c(d)|d,k)\alpha_k}{\sum_{k=1}^{K} p(c(d)|d,k)\alpha_k}.$$

With this definition the derivative of the Lagrangian in Equation 5 can be rewritten as

$$\frac{\partial L}{\partial \lambda_{s'c'k'}} = \sum_{d \in D} P_{k'} \frac{\partial \log \alpha_{k'} p(c(d)|d,k')}{\partial \lambda_{s'c'k'}}. \quad (6)$$

Performing the differentiation of the second term under the summation in 6 yields:

$$\frac{\partial \log \alpha_{k'} p(c(d)|d,k')}{\partial \lambda_{s'c'k'}} = \quad (7)$$

$$-\frac{1}{Z_{k'}(d)} \frac{\partial Z_{k'}(d)}{\partial \lambda_{s'c'k'}} + F_{s'}(c(d), d) I(c(d) = c'),$$

where $I()$ is the indicator function. Using the definition of Z from Equation 2 results in the following expression for its derivative:

$$\frac{\partial Z_{k'}(d)}{\partial \lambda_{s'c'k'}} = F_{s'}(c', d) exp[\sum_{s=1}^{S} \lambda_{sc'k'} F_s(c', d)]. \quad (8)$$

Substituting the result of Equation 8 into Equation 7 we obtain

$$\frac{\partial \log \alpha_{k'} p(c(d)|d,k')}{\partial \lambda_{s'c'k'}} = \quad (9)$$

$$F_{s'}(c', d) \left[I(c(d) = c') - \frac{exp[\sum_{s=1}^{S} \lambda_{sc'k'} F_s(c', d)]}{Z_{k'}(d)} \right]$$

$$= F_{s'}(c', d)[I(c(d) = c') - p(c'|d, k')].$$

Substituting the result of Equation 9 in Equation 6 yields the system of equations for the critical points of the log-likelihood:

$$\sum_{d \in D} P_{k'} F_{s'}(c', d)[I(c(d) = c') - p(c'|d, k')] = 0.$$

Note that for the EM algorithm to converge it is sufficient to make a step in the direction of the gradient in the M-step and proceed to E-step (McLachlan & Krishnan, 1997). Thus, for sufficiently small ϵ and for all $s' = 1, \ldots, S$ (constraints/features) and $k' = 1, \ldots, K$ (classes) we can do gradient ascent as follows:

$$\delta_{s'c'k'} = \sum_{d \in D} P_{k'} F_{s'}(c', d)[I(c(d) = c') - p(c'|d, k')];$$

$$\lambda_{s'c'k'}^{new} = \lambda_{s'c'k'}^{old} + \epsilon \delta_{s'c'k'}.$$

For the case of the mixture model, one could also directly consider a lower bound B on $l(\Lambda + \Delta) - l(\Lambda) \geq B$ (Chen & Rosenfeld, 1999; Nigam et al., 1999) and set Δ so that $B > 0$. In this case, the derivation goes along the lines of (Chen & Rosenfeld, 1999) and result in the following update equation:

$$\delta_{s'c'k'} = \log \frac{\sum_{d \in D} P_{k'} F_{s'}(c(d), d) I(c(d) = c')}{\sum_{d \in D} P_{k'} F_{s'}(c', d) p(c'|d, k')}.$$

The derivation of the update equation for the mixture weights α_k, $k = 1, \ldots, K$, follows the steps above and results in the following rule:

$$\alpha_k^{new} = \frac{1}{|D|} \sum_{d \in D} P_k.$$