Conditional Maximum Likelihood Estimation of Naive Bayes Probability Models Using Rational Function Growth Transform

Ciprian Chelba and Alex Acero
{chelba,alexac}@microsoft.com

April 5, 2004

Technical Report
MSR-TR-2004-33

We present a method for conditional maximum likelihood estimation of Naive Bayes models that employs a well known technique relying on a generalization of the Baum-Eagon inequality from polynomials to rational functions. The main advantage of the procedure is that it keeps the model parameter values (probabilities) properly normalized at each iteration. We apply the model trained under the maximum likelihood and conditional maximum likelihood criteria, respectively, to a text classification problem. A simple modification of the algorithm increases the convergence speed significantly over a straightforward implementation. The model trained under the conditional maximum likelihood criterion achieves a relative improvement of 40% in classification accuracy over its maximum likelihood counterpart on a text classification task.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

Naive Bayes models [1] have been successfully employed for a variety of classification tasks. Typically, the model parameters are estimated using a maximum likelihood criterion in conjunction with simple smoothing techniques. However, the maximum likelihood estimation criterion is not directly related to the performance of the model in terms of classification accuracy. A better training criterion is the conditional maximum likelihood one which is expected to be more correlated with the classification performance of the model.

We present a parameter estimation technique for Naive Bayes probability models that maximizes the conditional likelihood of a given training set. The algorithm employs a generalization of the Baum-Eagon inequality from polynomials defined over a domain of probability distributions to rational functions [2].

The main advantage of the technique over other re-estimation procedures such as gradient ascent is that the parameter values are guaranteed to be in the desired domain of probability distributions at each iteration.

The report is organized as follows: we first introduce conditional models that rely on a Naive Bayes assumption, mostly for fixing notation. Section 3 presents the re-estimation procedure under the conditional maximum likelihood criterion, including a few refinements that improve the convergence speed and stability. Section 4 presents text classification experiments on the Air Travel Information System (ATIS) [3] data.

2 Conditional Models Relying on a Naive Bayes Assumption

In many practical applications one seeks to model a conditional probability $P(y|x), y \in \mathcal{Y}, x \in \mathcal{X}$. A common situation is that in which we identify a set of features deemed relevant for building the model. Since we are interested in building a conditional model that relies on a Naive Bayes assumption, we will restrict our attention to features whose support is included in $\mathcal{X}$; the features are binary valued indicator functions $f(x) : \mathcal{X} \rightarrow \{0, 1\}$. Let $\mathcal{F} = \{f_k, k = 1 \ldots F\}$ be the set of features chosen for building a particular model $P(y|x)$.

For any given event (x, y) one constructs a binary valued feature vector listing the values each feature takes at this particular point:

$$\underline{f}(x) = (f_1(x), \ldots, f_F(x))$$

For convenience we denote $\overline{f_i(x)} = 1 - f_i(x)$.

Assuming a Naive Bayes model for the feature vector and the predicted variable $(\underline{f}(x), y)$,

$$P(\underline{f}(x), y) = \theta_y \prod_{k=1}^{F} \theta_{ky}{}^{f_k(x)} \overline{\theta}_{ky}{}^{\overline{f_k(x)}}$$

the conditional probability $P(y|x)$ can be calculated as:

$$P(y|x;\underline{\theta}) \quad = \quad Z(x;\underline{\theta})^{-1} \cdot \theta_y \prod_{k=1}^{F} {\theta_{ky}}^{f_k(x)} \overline{\theta}_{ky}^{\overline{f_k(x)}} \tag{1}$$

where:

- $\theta_y \geq 0, \forall y \in \mathcal{Y}, \sum_{y \in \mathcal{Y}} \theta_y = 1$;
- $\theta_{ky} \geq 0, \overline{\theta}_{ky} \geq 0, \theta_{ky} + \overline{\theta}_{ky} = 1, \forall k = \overline{1, F}, y \in \mathcal{Y}$;
- $Z(x;\underline{\theta})^{-1} = \sum_y P(\underline{f}(x), y)$

We note that $P(x, y) = P(\underline{f}(x), y)$ does not result in a proper probability model since different (x, y) values may map to the same $(\underline{f}(x), y)$ value. Also, since the model uses one free parameter θ_{ky} for each different $y \in \mathcal{Y}$, one could represent the model equivalently using the usual $f_k(x, y) = f_k(x) \cdot \delta(y)$ features.

2.1 Relationship with Log-Linear Models

A simple re-parameterization of the conditional model presented above results in a log-linear model.

First let's note that:

$$P(\underline{f}(x), y) = \theta_y \cdot \prod_{k=1}^{F} {\theta_{ky}}^{f_k(x)} \cdot {\overline{\theta}_{ky}}^{\overline{f_k(x)}}$$

can be rewritten as:

$$P(\underline{f}(x), y) = \theta_y \cdot \prod_{k=1}^{F} \overline{\theta}_{ky} \cdot \prod_{k=1}^{F} \left[\frac{\theta_{ky}}{\overline{\theta}_{ky}} \right]^{f_k(x)}$$

Setting:

- $f_k(x, y) = f_k(x) \cdot \delta(y)$
- $\lambda_{ky} = \log(\frac{\theta_{ky}}{\overline{\theta}_{ky}})$;
- $\lambda_{0y} = log(\theta_y \cdot \prod_{k=1}^{F} \overline{\theta}_{ky})$;
- $f_0(x, y) = f_0(y)$

we have:

$$P(y|x;\underline{\lambda}) \quad = \quad Z(x;\underline{\lambda})^{-1} \cdot exp(\sum_{k=0}^{F} \lambda_{ky} f_k(x, y)) \tag{2}$$

which is the familiar log-linear model arrived at in maximum entropy probability estimation [4]. Consequently, the estimation procedure presented in the next section is applicable to this class of log-linear models as well, resulting in an alternative to the usual estimation techniques employed for log-linear models.

3 Conditional Maximum Likelihood Estimation of Naive Bayes Models

It is desirable to estimate the model parameters $\underline{\theta} = \{\theta_y, \theta_{ky}, \overline{\theta_{ky}}, \forall y \; and \; k\}$ such that the conditional likelihood $H(\mathcal{T}; \underline{\theta}) = \prod_{j=1}^{T} P(y_j | x_j)$ assigned by the model to a set of training samples $\mathcal{T} = \{(x_1, y_1) \ldots (x_T, y_T)\}$ is maximized:

$$\underline{\theta}^* = \arg \max_{\underline{\theta}} H(\mathcal{T}; \underline{\theta}) \tag{3}$$

It is easy to note that $H(\mathcal{T}; \underline{\theta})$ is a ratio of two polynomials with real coefficients, each defined over a set $\times$ of probability distributions:

$$\times = \{\underline{\theta} : \theta_y \geq 0, \forall y \in \mathcal{Y} \; and \; \sum_y \theta_y = 1; \theta_{ky} \geq 0, \overline{\theta}_{ky} \geq 0 \; and \; \theta_{ky} + \overline{\theta}_{ky} = 1, \forall y \in \mathcal{Y}, \forall k = 1 \ldots F\}$$

Following the development in [2] one can iteratively estimate the model parameters using a growth transform for rational functions on the domain $\times$. The re-estimation equations take the form:

$$\begin{aligned}
\widehat{\theta_y} &= N^{-1} \theta_y \left(\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_y} + C_{\underline{\theta}} \right) && (4) \\
N &= C_{\underline{\theta}} + \sum_y \theta_y \frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_y} \\
\widehat{\theta_{ky}} &= N_y^{-1} \theta_{ky} \left(\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_{ky}} + C_{\underline{\theta}} \right) && (5) \\
\widehat{\overline{\theta}_{ky}} &= N_y^{-1} \overline{\theta}_{ky} \left(\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \overline{\theta}_{ky}} + C_{\underline{\theta}} \right) \\
N_y &= C_{\underline{\theta}} + \theta_{ky} \frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_{ky}} + \overline{\theta}_{ky} \frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \overline{\theta}_{ky}}
\end{aligned}$$

where $C_{\underline{\theta}} > 0$ is chosen such that

$$\begin{aligned}
\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_y} + C_{\underline{\theta}} &> \epsilon, \forall y && (6) \\
\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \theta_{ky}} + C_{\underline{\theta}} &> \epsilon, \forall k \; and \; y \\
\frac{\partial \log H(\mathcal{T}; \underline{\theta})}{\partial \overline{\theta}_{ky}} + C_{\underline{\theta}} &> \epsilon, \forall k \; and \; y
\end{aligned}$$

with $\epsilon > 0$ suitably chosen, see [2] for details.

The main advantage of the growth transform re-estimation procedure is that the model parameters are renormalized at each iteration, maintaining them in the parameter space $\times$; using traditional gradient ascent techniques would require extra care.

The technique is widely used in discriminative training of Hidden Markov models for speech recognition [5], [6].

3.1 Smoothing and Initialization

When experimenting with the model we found that smoothing improves the performance of the model. The model parameters $\underline{\theta}$ are initialized using maximum likelihood estimates smoothed using MAP:

$$\begin{aligned}
\theta_y &= \frac{\sum_{i=1}^{T} \delta(y, y_i) + \alpha \frac{1}{|\mathcal{Y}|}}{T + \alpha} && (7) \\
\theta_{ky} &= \frac{\sum_{i=1}^{T} f_k(x_i)\delta(y, y_i) + \alpha \frac{1}{2}}{\sum_{i=1}^{T} \delta(y, y_i) + \alpha} && (8) \\
\overline{\theta}_{ky} &= \frac{\sum_{i=1}^{T} \overline{f_k(x_i)}\delta(y, y_i) + \alpha \frac{1}{2}}{\sum_{i=1}^{T} \delta(y, y_i) + \alpha}
\end{aligned}$$

The optimal value for the MAP weight α is determined such that it maximizes classification accuracy on cross validation data.

As for reestimating the model parameters using the above smoothing scheme, it is convenient to note that Eq. (8) can be equivalently written as linear interpolation between the relative frequency estimate for $\theta_{ky}, \overline{\theta}_{ky}$ and the uniform distribution ($\frac{1}{2}$).

If we denote counts of various events by #(), we can rewrite Eq. (8) as:

$$\begin{aligned}
\theta_{ky} &= \lambda_y \cdot \frac{\#(k, y)}{\#(y)} + \overline{\lambda_y} \cdot \frac{1}{2} && (9) \\
\lambda_y &= \frac{\#(y)}{\#(y) + \alpha} \\
\overline{\lambda_y} &= \frac{\alpha}{\#(y) + \alpha}
\end{aligned}$$

Based on Eq. (1) the model can be re-parameterized as:

$$P(y|x; \underline{\theta}) = Z(x; \underline{\theta})^{-1} \cdot \theta_y \prod_{k=1}^{F} (\lambda_y \cdot \theta_{ky} + \overline{\lambda_y} \cdot \frac{1}{2})^{f_k(x)} (\lambda_y \cdot \overline{\theta}_{ky} + \overline{\lambda_y} \cdot \frac{1}{2})^{\overline{f_k(x)}} \quad (10)$$

where:

- $\theta_y \geq 0, \forall y \in \mathcal{Y}, \sum_{y \in \mathcal{Y}} \theta_y = 1$;
- $\theta_{ky} \geq 0, \overline{\theta}_{ky} \geq 0, \theta_{ky} + \overline{\theta}_{ky} = 1, \forall k = \overline{1, F}, y \in \mathcal{Y}$;
- $\lambda_y \geq 0, \overline{\lambda}_y \geq 0, \lambda_y + \overline{\lambda}_y = 1, \forall y \in \mathcal{Y}$;
- $Z(x; \underline{\theta})^{-1} = \sum_y P(\underline{f}(x), y)$

We note that under the parameterization for $P(y|x)$ in Eq. (10), the conditional likelihood $H(\mathcal{T}; \underline{\theta}) = \prod_{j=1}^{T} P(y_j|x_j)$ is still a ratio of polynomials as required by [2].

3.2 Model Parameter Updates

Calculating the partial derivatives in Eq. (5) we obtain:

$$\widehat{\theta_{ky}} = N_{ky}^{-1} \cdot \theta_{ky} \cdot \left\{ 1 + \beta_{\underline{\theta}} \frac{\lambda_y}{\lambda_y \cdot \theta_{ky} + \overline{\lambda_y} \cdot \frac{1}{2}} \sum_{i=1}^{T} f_k(x_i)[\delta(y, y_i) - p(y|x_i; \underline{\theta})] \right\} \quad (11)$$

$$\widehat{\overline{\theta}_{ky}} = N_{ky}^{-1} \cdot \overline{\theta}_{ky} \left\{ 1 + \beta_{\underline{\theta}} \frac{\lambda_y}{\lambda_y \cdot \theta_{ky} + \overline{\lambda_y} \cdot \frac{1}{2}} \sum_{i=1}^{T} \overline{f_k(x_i)}[\delta(y, y_i) - p(y|x_i; \underline{\theta})] \right\}$$

where N_{ky}^{-1} is a normalization constant that ensures that $\widehat{\theta_{ky}} + \widehat{\overline{\theta}_{ky}} = 1$, $\beta_{\underline{\theta}} = 1/C_{\underline{\theta}}$ and $\delta()$ is the Kronecker delta operator, $\delta(y, y_i) = 1 \; for \; y = y_i, 0 \; otherwise$.

3.2.1 Model Parameter Initialization

The θ_y, θ_{ky} parameters are initialized to their ML values (relative frequency). The MAP weight α, common for all classes y, is calculated by line search such that the conditional likelihood on cross-validation data is maximized. The resulting interpolation weight λ_y for each class y is fixed to the value determined using Eq. (9).

Only the $\theta_{ky}, \overline{\theta}_{ky}$ parameters are re-estimated using the RFGT transform. The interpolation weights λ_y and the class priors θ_y are not re-estimated since the model performance is not very sensitive to their values.

3.3 Refinements

The choice of $\beta_{\underline{\theta}} = 1/C_{\underline{\theta}}$ where $C_{\underline{\theta}}$ satisfies Eq. (6) is problematic:

- the correct value of ϵ that will ensure monotonic increase in conditional likelihood $H(\mathcal{T}; \underline{\theta})$ is hard to determine, see [2]. Large values for ϵ will slow down convergence whereas small values may result in non-monotonic increase of the conditional likelihood.

- assuming that we choose $\beta_{\underline{\theta}}$ such that Eq. (6) is satisfied for a small non-negative value of ϵ, that value for $\beta_{\underline{\theta}}$ may still be very small. In practice we noticed that typically there exists a pair of (k, y) values that will require a very small $\beta_{\underline{\theta}}$ value which will make the updates for other (k, y) pairs very small, slowing down convergence. Fortunately, as pointed out in [2], one can use different $\beta_{\underline{\theta}}$ values for different probability distributions in ×, in our case making it sensitive to the (k, y) value: $\beta_{\underline{\theta}} = \beta_{\underline{\theta}}(k, y)$. We have exploited this flexibility as explained later.

3.3.1 Dependency on Training Data Size

Since the sums in the above equations run over the entire training data, it is useful to normalize with respect to training data size, namely use $\beta_{\underline{\theta}} = \frac{\gamma_{\theta}}{T}$ in Eq. (11)

3.3.2 Dependency on Class Count

It can be easily noted that the update Eqns. (11) could be made dimensionally correct if one let $\beta_{\underline{\theta}} \cdot \sum_{i=1}^{T} f_k(x_i)[\delta(y, y_i) - p(y|x_i; \underline{\theta})]$ have the dimension of a probability θ_{ky}. We can achieve this by using $\beta_{\underline{\theta}} = \frac{\zeta_{\underline{\theta}}(y)}{\#(y)}$. In practice we noticed that this parameterization results in a considerable convergence speedup.

We note that setting $\zeta_{\underline{\theta}}(y) = \frac{\#(y)}{T} \cdot \gamma_{\underline{\theta}}$ would result in the exact same update equations under either the γ or the ζ updates. Since $\frac{\#(y)}{T} = \theta_y < 1.0 \forall y$, the speed-up observed in practice can thus be partly attributed to using a higher step-size in the ζ-based updates. However, in our experimental setup the prior distribution $\theta_y = \frac{\#(y)}{T}$ is highly skewed towards one particular class ($\theta_{FLIGHT} = 0.74$) so we believe that not all of the convergence speed improvement comes from the difference in step size (see Fig. 1) although a direct comparison of the relative performance of the two parameter update equations is difficult.

3.3.3 Choosing $\gamma_{\underline{\theta}}, \zeta_{\underline{\theta}}$

As explained in [2] one can use a separate value $\gamma_{\underline{\theta}}$ for each probability distribution in ×. Our scheme for determining $\gamma_{\underline{\theta}}(k, y)$ — one for each (k, y) pair — is as follows:

- for each pair (k, y)
  - set $\gamma_{\underline{\theta}}(k, y) = \gamma_{min}$
  - for the (k, y) values where the previous choice does not meet Ineq. (6), set $\gamma_{\underline{\theta}}(k, y)$ lower such that we have equality in (6)
  - $\forall(k, y)$ set $\gamma_{\underline{\theta}}(k, y) = (1 - e^{-7}) \cdot \gamma_{\underline{\theta}}(k, y)$ such that the margin by which the (k, y) pairs satisfy Ineq. (6) for $\epsilon = 0$ is increased. We note that choosing $\epsilon = 0$ or very small doesn't guarantee monotonic increase of the conditional log-likelihood on training data, as outlined in [2] and confirmed by our experiments

The value γ_{min} as well as the number of re-estimation iterations are chosen such that they maximize classification accuracy on cross validation data.

A similar procedure is followed in the case of parameter updates based on $\zeta_{\underline{\theta}}$.

4 Experiments

We applied the Naive Bayes classifier to the problem of text classification in the ATIS domain. The Naive Bayes classifier was trained under both the maximum likelihood (ML) and conditional maximum likelihood (CML) criteria. We have compared the CML NB classifier against a MaxEnt one [7].

4.1 Experimental Setup

We have extracted the class A sentences from the ATIS II and ATIS III corpora [3]. As class labels we have used the SQL queries that accompany each sentence to extract class labels:

```
<s.G01011SX.SQL.0> class=flight
<s> show me the one way flights from detroit to westchester county </s>

( SELECT DISTINCT flight.flight_id FROM flight WHERE (
flight.from_airport IN ( SELECT airport_service.airport_code FROM
airport_service WHERE airport_service.city_code IN ( SELECT
city.city_code FROM city WHERE city.city_name = 'DETROIT' ))
AND ( flight.to_airport IN ( SELECT airport_service.airport_code
FROM airport_service WHERE airport_service.city_code IN ( SELECT
city.city_code FROM city WHERE city.city_name = 'WESTCHESTER COUNTY'
)) AND flight.flight_id IN ( SELECT flight_fare.flight_id FROM
flight_fare WHERE flight_fare.fare_id IN ( SELECT fare.fare_id FROM
fare WHERE ( fare.round_trip_required = 'NO' AND 1 = 1 )))))) ;
```

As training data we have used all the class A sentences in ATIS II and ATIS III, a total of 5822 sentences. As test data we have used the class A sentences from the 1993 and 1994 ATIS evaluation test sets, a total of 914 sentences.

The class vocabulary contained 14 classes extracted as described above. From the training data we constructed a word vocabulary containing all the 780 words seen in the training data.

The features $f_k(x)$ were taken to identify whether a given word w_k in the vocabulary is present or not in a given sentence x. We have then built a conditional Naive Bayes model for estimating the probability that a sentence x belongs to a class y, $P(y|x)$. The model parameters were estimated using both the maximum likelihood (ML) and conditional maximum likelihood (CML) criteria described above.

Each test sentence x was assigned to the most likely class, $y^* = \arg\max_y P(y|x)$.

The training data was randomly split into main (70% of train) and check (30% of train). The check data was used for determining the MAP smoothing weight α, see Eq.(7-8), the γ_{min}, ζ_{min} values as well as the number of CML training iterations. Once these are fixed to the values that yield maximum classification accuracy on check data, the entire training data is used to estimate the final parameter values for both ML and CML.

4.2 Convergence Speed Experiments

We have studied the effect of the choice for γ_{min}, ζ_{min} on the convergence of the CML training algorithm. Figures (1-2) show the gain in convergence speed brought by using the re-estimation equations using the ζ rather than γ updates — first 300 training iterations are shown.

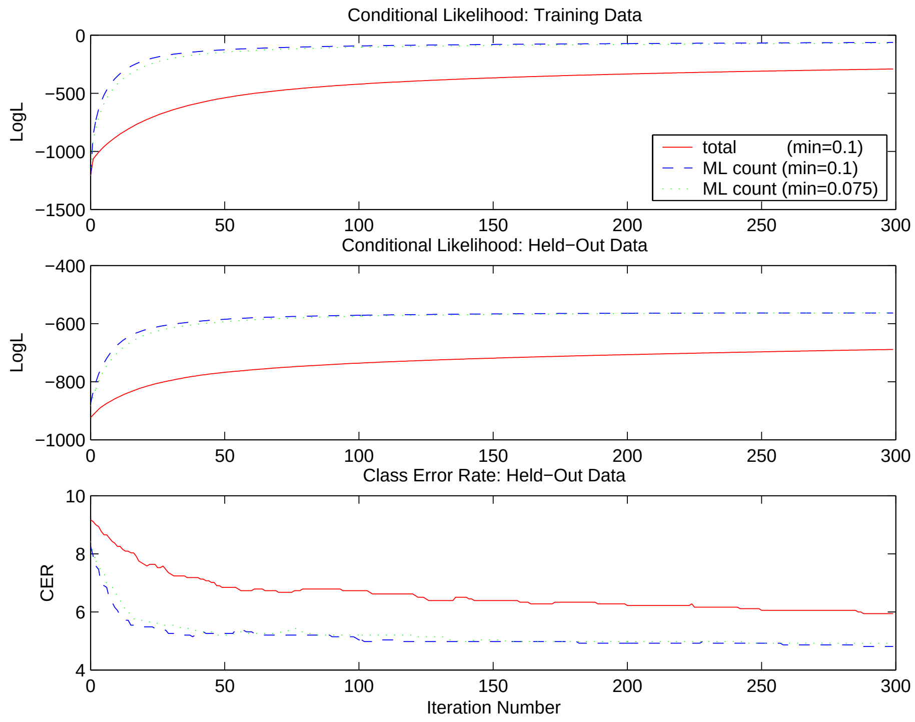

Figure 1: Convergence Speed Improvement $\gamma_{min}(-) = \zeta_{min}(--) = 0.1$ and $\zeta_{min}(\cdot) = 0.075$

As can be seen the improvement in convergence speed when measured as log-likelihood increase of training/held-out data for a given number of iterations is quite significant for both $\gamma_{min} = \zeta_{min} = 0.5$ and 0.1. When using $\zeta_{min} = 0.5$ the re-estimation is stopped after 92 iterations due the fact that the conditional log-likelihood on training data is no longer monotonically increasing.

A higher value for the $\gamma_{min} = \zeta_{min}$ parameter does lead to faster increase in conditional log-likelihood.

4.3 Classification Experiments

We have run classification experiments in the above training/test setup. The optimal α, ζ_{min} values as well as the number of CML training iterations were determined such that they yield maximum classification accuracy on check data, approximatively 30% of the training data. The value $\alpha = 0.14$ was found using line search 1:0.01:0; $\zeta_{min} = 0.1$ as well as the optimal number of CML iterations (275) were determined using line search 0.5:0.1:0 when running at most 500 training iterations.

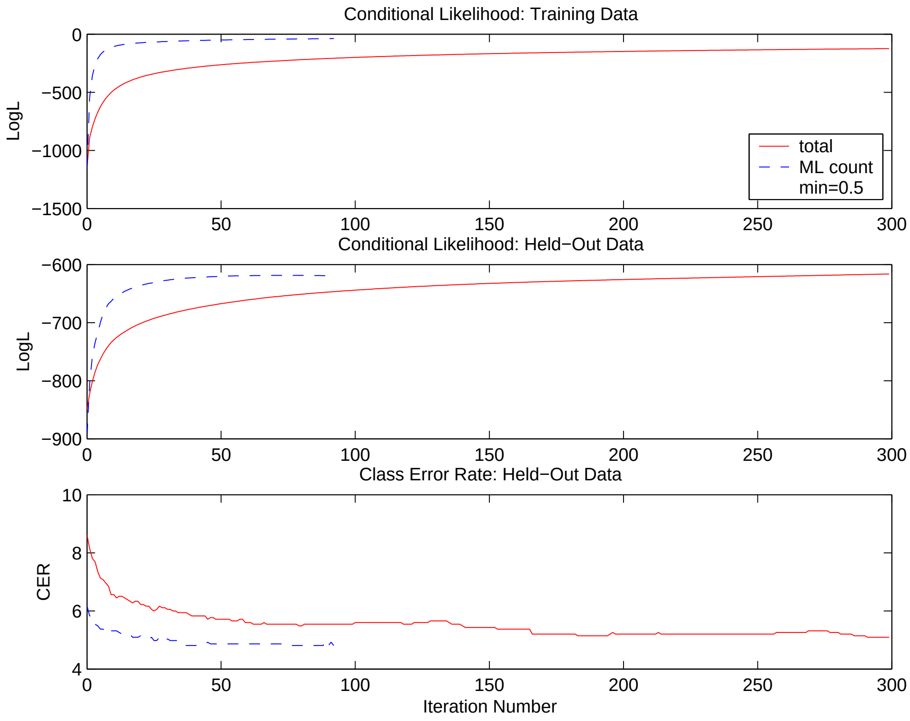

Figure 2: Convergence Speed Improvement $\gamma_{min}(-) = \zeta_{min}(--) = 0.5$

Table 1 shows the classification results. The CML trained Naive Bayes classifier reduces the classification error rate of the ML baseline by 40%. Although theoretically equivalent, MaxEnt and CML Naive Bayes do not perform equally well. A few reasons that could explain the performance gap are:

- the model parameterization is slightly different due to smoothing; in the case of the MaxEnt model smoothing results in a slightly modified objective function which has an extra term besides the conditional log-likelihood, see [8]
- for the log-linear (MaxEnt) model the objective function is convex in the model parameters (see Eq. 2) whereas this is not true in the Naive Bayes parameterization
- the RFGT estimation procedure has an extra free parameter that is set on cross-validation data (γ_{min} or ζ_{min}) whereas the GIS,IIS algorithms [9] typically used for estimating the MaxEnt model do not.

Training Objective Function	Smoothing	Class Error Rate (%)
ML	smoothed	11.3
CML	not smoothed	8.5
CML	smoothed	6.7
MaxEnt	smoothed	4.9

Table 1: Classification Error

Class	Frequency	ML CER (%)	CML CER (%)
FLIGHT	0.740	5.0	3.0
FARE	0.055	38.0	22.0
GROUND SERVICE	0.048	2.3	2.3
AIRLINE	0.046	28.6	16.7
AIRCRAFT	0.035	15.6	3.1
AIRPORT	0.024	22.7	22.7
FARE BASIS	0.014	7.7	23.1
AIRPORT SERVICE	0.010	11.1	11.1
CITY	0.009	50.0	87.5
FOOD SERVICE	0.008	57.1	0.0
CLASS OF SERVICE	0.004	100.0	50.0
RESTRICTION	0.004	0.0	0.0
DAYS	0.003	100.0	100.0
FLIGHT STOP	0	0.0	0.0

Table 2: Classification Error: Error Analysis

Table 2 compares the performance of the ML and the CML classifier on different classes, respectively. The most notable improvement is the reduction in error rate by 41% on the FLIGHT class, which occurs 74% of the test data. The CML training does not reduce the error rate across all classes.

5 Acknowledgments

Thanks to Milind Mahajan and Asela Gunawardana for useful comments and discussions.

References

[1] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, John Wiley and Sons, Inc., second edition, 2001.

[2] P. S. Gopalakrishnan, Dimitri Kanevski, Arthur Nadas, and David Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 107–113, January 1991.

[3] D. Pallet, J. Fiscus, W. Fisher, J. Garofolo, B. Lund, A. Martin, and M. Przybocki, "1993 benchmark tests for the ARPA spoken language program," in *Proceedings of the Human Language Technology Workshop*, C.J. Weinstein, Ed. Morgan Kaufmann, Plainsboro, NJ, March 1994.

[4] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–72, March 1996.

[5] Y. Normandin, *"Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem"*, Ph.D. thesis, McGill University, Montreal, 1991.

[6] P. Woodland and D. Povey, "Large scale discriminative training for speech recognition," in *ISCA ITRW Automatic Speech Recognition: Challenges for the Millenium*, Paris, 2000, pp. 7–16.

[7] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," in *Proceedings of ICASSP*, Hong Kong, April 2003, IEEE, vol. I, pp. 280–283.

[8] Stanley F. Chen and Ronald Rosenfeld, "A survey of smoothing techniques for maximum entropy models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 37–50, 2000.

[9] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," Tech. Rep. CMU-CS-95-144, School of Computer Science, Carnegie Mellon University, Pittsburg, PA, 1995.