

Using Simulated Pseudo Data To Speed Up Statistical Predictive Modeling From Massive Data Sets

*Ramesh Natarajan and Edwin Pednault**

1 Introduction

Predictive modeling techniques are now being used in application domains where the training data sets are potentially enormous. For example, certain marketing databases that we have encountered contain millions of customer records with thousands of attributes per record. The development of statistical modeling algorithms for such massive data sets is important for emerging data mining application areas such as database marketing, targeted advertising, fraud detection, credit/risk modeling, and information retrieval.

One widely-used approach to this problem is to reduce the amount of data prior to the statistical modeling procedure. For example, given a data set with N records and J features per record, sub-sampling may be used to select some $N' \ll N$ records. Also, a preliminary feature selection or feature transformation may be used to select some $J' \ll J$ features. The reduced data set with N' records and J' features can be used to perform the required statistical modeling in a computationally tractable way. However, sub-sampling can be unsatisfactory for heterogenous and high-dimensional data sets, and the resulting model estimates will have a high degree of variability. Similarly, the preliminary feature selection or feature transformation may be unsatisfactory when it inadvertently excludes features that may be critical for the accuracy of the eventual statistical models.

In this paper, we describe a heuristic approach for massive data sets that is

*I.B.M. T. J. Watson Research Center, Yorktown Heights, NY 10598

based on constructing a simple probability model from the data, and using this model to generate simulated pseudo data that can be used for some aspects of the overall statistical modeling procedure. In order to illustrate this approach, we consider the specific problem of a Naive Bayes model with feature selection. This model is widely used in the machine learning community for predicting multivariate categorical response variables. The usual feature selection algorithm requires numerous data scans to find the Naive Bayes model with the optimum feature subset. However, our heuristic approach uses just two data scans to obtain models of comparable accuracy.

2 Notation and Background

Let y denote the categorical response variable that takes on K values denoted by $1, 2, \dots, K$ respectively. Let the set of J covariate predictors for y be denoted by X_1, X_2, \dots, X_J respectively. The corresponding vector of covariates is denoted by

$$\mathbf{X} = [X_1 \quad X_2 \quad \dots \quad X_J]^T. \quad (1)$$

We now let

$$\theta_k(\mathbf{x}) = P(y = k | \mathbf{X}), \quad (2)$$

denote the conditional probability that y takes the value k (note that $\sum_{k=1}^K \theta_k(\mathbf{X}) = 1$).

In many cases, the objective of the predictive modeling is to classify the categorical response based on the measured covariates. The optimal Bayes classification is given by

$$l(\mathbf{X}) = \arg \max_k \theta_k(\mathbf{X}). \quad (3)$$

The estimates $\hat{\theta}_k(\mathbf{X})$ for $\theta_k(\mathbf{X})$ in (3) are obtained from the training data set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ by some statistical estimation procedure (e.g., maximum likelihood). These estimates are used in (3) to obtain corresponding estimates $\hat{l}(\mathbf{X})$ for $l(\mathbf{X})$. Previous work (e.g., Domingos and Pazzani [8], Friedman [9]) has shown that estimates for $\hat{l}(\mathbf{X})$ with low classification error rates can be obtained even if low-accuracy estimates $\hat{\theta}_k(\mathbf{X})$ are used in (3). However, aside from the classification problem, there are other applications where accurate estimates for $\hat{\theta}_k(\mathbf{X})$ may be required. One example is targeted marketing, where customers are ranked in the order of the estimated probability of responding to a particular promotional offering.

We consider the empirical expected negative log-likelihood of the N training data points $\{\mathbf{x}_i, y_i\}_{i=1}^N$, given by

$$\mathcal{L}_{TR} = -\frac{1}{N} \sum_{i=1}^N \log \theta_{y_i}(\mathbf{x}_i). \quad (4)$$

The function \mathcal{L}_{TR} plays an essential role in predictive modeling, and most algorithms are based in one way or the other on the minimisation of this function.

The feature selection problem is based on finding the optimal subset of features of cardinality $\hat{J} \leq J$ from the data, for which the corresponding model has smallest

generalization error. The selection of this right-sized model is based on constructing a sequence of nested models using feature subsets of increasing cardinality (we discuss the details of how this sequence of models is constructed in Section (4)). The response probability estimate for the model of size j in this sequence is denoted by $\hat{\theta}_k^{(j)}(\mathbf{x})$.

The generalization error with each model $\hat{\theta}_k^{(j)}(\mathbf{x})$ in this sequence can be estimated from the empirical negative log-likelihood for a separate test data set. This estimate is denoted by $\mathcal{L}_{TE}(\hat{\theta}_k^{(j)}(\mathbf{x}))$, and the optimal value \hat{J} is then given by

$$\hat{J} = \arg \min_j \mathcal{L}_{TE}(\hat{\theta}_k^{(j)}(\mathbf{x})). \quad (5)$$

In many cases, there is insufficient data for an independent test data set, and an alternative estimate for the generalization error for each model size is obtained from n -fold cross-validation (where n is typically chosen to be 5 or 10). Here the original training data set is divided into n roughly-equal segments, with segment k containing N_k points (note that $N = \sum_{k=1}^n N_k$). For each k , the $N - N_k$ data points are used as training data to obtain a nested sequence of models $\hat{\theta}_k^{(k,j)}(\mathbf{x})$ in the usual way, and the remaining N_k data points are used as test data for evaluating $\mathcal{L}_{TE}(\hat{\theta}_k^{(k,j)}(\mathbf{x}))$. This procedure is repeated for each value of k and the results are averaged to obtain the n -fold cross-validation estimate of the generalization error, which in turn yield the optimum value \hat{J} as

$$\hat{J} = \arg \min_j \left[\frac{1}{N} \sum_{k=1}^n N_k \mathcal{L}_{TE}(\hat{\theta}_k^{(k,j)}(\mathbf{x})) \right]. \quad (6)$$

Both the test set and the cross-validation estimates of the generalization error require additional computation which may be too expensive in certain applications. In that case, a simpler approach is to obtain \hat{J} from a penalized form of \mathcal{L}_{TR} in the form

$$\hat{J} = \arg \min_j \left[\mathcal{L}_{TR}(\hat{\theta}_k^{(j)}(\mathbf{x})) + \frac{1}{2} j \alpha \right]. \quad (7)$$

Here a penalty term involving the model complexity is used in order to avoid overfitting to the training data. A suitable value for the regularization parameter α can be specified *a priori*. A widely-used choice in regression problems is the Bayes information criterion $\alpha_{BIC} = \log N$, where N is the number of training data points (Schwarz [23]).

3 Naive Bayes Model

The conditional probability (2) can be expressed using Bayes rule as

$$\theta_k(\mathbf{x}) = \frac{\pi_k P(\mathbf{X} = \mathbf{x} | y = k)}{\sum_{k'=1}^K \pi_{k'} P(\mathbf{X} = \mathbf{x} | y = k')} \quad (8)$$

The priors π_k may either be specified extrinsically, or may be estimated using the response frequencies in the training data as

$$\pi_k = \frac{N_k + \lambda_k}{N + \sum_{k'=1}^K \lambda_{k'}} \quad , \quad (9)$$

where $\sum_k N_k = N$. The frequency counts in (9) have been smoothed using the parameters $\{\lambda_k\}_{k=1}^K$ to avoid overfitting the data for small sample sizes. The extent of smoothing is controlled by the parameters λ_k , which in the Bayesian framework may be regarded as parameters of a Dirichlet conjugate prior for the multinomial probability distribution (Agresti [2]). The special case of the uniform Dirichlet prior $\lambda_k = 1$ in (9) leads to the well-known Laplace smoothing correction.

A simple and widely-used model for the covariate dependencies on $\theta_k(\mathbf{x})$ in (8) is provided by the Naive Bayes assumption, where

$$P(\mathbf{X} = \mathbf{x}|y = k) = \prod_{j=1}^J P(X_j = x_j|y = k). \quad (10)$$

This states that the covariates $\{X_j\}_{j=1}^J$ in \mathbf{X} are all conditionally independent given the response y . With this assumption, the relevant conditional probabilities in (8) can be estimated from a set of frequency counts that can be obtained in a single training data scan.

Any continuous covariates X_j can be treated in (10) by fitting a simple parametric or non-parametric model to the corresponding univariate marginal distributions $P(X_j = x_j|y = k)$ from the training data. The univariate gaussian is frequently used, but more elaborate kernel density models have also been used by John, Kohavi and Pfelger [15]. In our experiments, we have uniformly pre-discretized the continuous variables and binned the values into 8 bins. For simplicity, we have not considered non-uniform discretization nor have we tuned the number of bins. Hsu, Kuang and Wong ([14]) have shown that even this simple uniform discretization can be as effective as more elaborate non-uniform schemes (Kohavi and Sahimi [17]) with respect to the eventual Naive Bayes model accuracy.

We assume that the intrinsic or derived categorical feature X_j takes on M_j values denoted by $1, 2, \dots, M_j$ respectively. Then, the estimate P_{jmk} for $P(X_j = m|y = k)$ in (10) is given by

$$P_{jmk} = \frac{N_{jmk} + \lambda_{jmk}}{N_k + \sum_{m'=1}^{M_j} \lambda_{jm'k}} \quad , \quad (11)$$

where $\sum_{m=1}^{M_j} N_{jmk} = N_k$. Here also, the frequency counts have been smoothed to avoid overfitted estimates for the probabilities, and the comments following (9) apply here as well. In particular, setting the smoothing parameters $\lambda_{jmk} = 1$ corresponds to Laplace smoothing. The frequency counts in (9) and (11) are the sufficient statistics for estimating π_k and P_{jmk} , and these can be accumulated in a single pass over the data.

The Naive Bayes model (8)-(10) can be written in the form of a logistic model

$$\theta_k(\mathbf{X}) = \frac{e^{f_k(\mathbf{X})}}{\sum_{k'=1}^K e^{f_{k'}(\mathbf{X})}} \quad , \quad (12)$$

where $f_1(\mathbf{X})$ is set to zero for model identifiability (Agresti [2]), by setting

$$f_k(\mathbf{x}) = \log\left(\frac{\pi_k}{\pi_1}\right) + \sum_{j=1}^J \sum_{m=1}^{M_j} \log\left(\frac{P_{jmk}}{P_{jm1}}\right) \delta_{x_j,m} \quad , \quad (13)$$

where $\delta_{x_j,m}$ denotes the Kronecker delta function. This is a simplified logistic model for categorical covariates in which nonlinear and interaction effects are omitted (see Agresti [2], pp. 92-93).

The empirical negative log-likelihood for the training data is obtained from (8), (10) and (11) as

$$\begin{aligned} \mathcal{L}_{TR} = & \underbrace{- \sum_{k=1}^K \frac{N_k}{N} [\log \pi_k + \sum_{j=1}^J \sum_{m=1}^{M_j} \frac{N_{jmk}}{N_k} \log P_{jmk}]}_{\mathcal{A}} \\ & + \underbrace{\frac{1}{N} \sum_{i=1}^N \log(\sum_{k'} \{\prod_{j=1}^J P_{jx_{j,i}k'}\} \pi_{k'})}_{\mathcal{B}} \quad , \quad (14) \end{aligned}$$

where $x_{j,i}$ denotes the value of x_j for the i 'th training data point. Note that the term denoted by \mathcal{A} in (14) can be evaluated exactly using the estimates of $\pi_k(t)$ and $P_{jmk}(t)$ from (9) and (11). However, the term denoted by \mathcal{B} in (14) can only be evaluated by a separate pass over the training data, with the contribution of each data point being evaluated and summed. Similarly, the feature selection algorithm will require \mathcal{L}_{TR} to be evaluated for the Naive Bayes models obtained with various subsets of features. In each such case, the corresponding \mathcal{A} term can be evaluated exactly from the frequency counts alone; however the evaluation of the corresponding \mathcal{B} term requires a separate data scan.

4 Empirical Study of Feature Selection

The empirical utility of feature selection in a Naive Bayes classifier can be evaluated using standard data sets in the machine learning literature (see Table (1)). Table (2) shows that the Naive Bayes model with feature selection has lower error rates than CART (Breiman, Friedman, Olshen and Stone, [6]). The only notable exceptions are glass, tictactoe, and vehicle. Table (2) also shows that feature selection is beneficial in the Naive Bayes model since in most cases the optimum model used only a subset of the original features. Similar results have been reported by Langley, Iba and Thomson [20] (see also Langley and Sage [19]). The results in Table (2) are obtained using data sets where the features were likely to have been selected based on their predictive utility. This suggests that feature selection would be even more useful in general ‘‘data mining’’ data sets.

Langley, Iba, and Thomson [20] use a greedy forward selection procedure for finding the optimum feature subset, where features are sequentially added to the

Nome	records	features	response classes
balance ³	625	4	3
breast(Wisconsin) ^{3*}	683	9	2
digit ¹	200	7	10
flea ³	74	6	3
german ²	1000	20	2
glass ³	214	10	6
heart ²	270	13	2
ionosphere ^{3*}	351	34	2
iris ³	150	4	3
letter ³	20000	16	26
mushroom ^{3*}	5644	22	2
segment ²	2310	19	7
tictactoe ³	958	9	2
vehicle ²	846	18	4
votes ³	435	16	2
waveform ³	5000	21	3

Table 1. Description of data sets obtained from (1) Breiman, Friedman, Olshen and Stone [6], (2) Brazdil and Gama [5], and (3) Blake, Keogh and Merz [3]. The datasets with the superscript * contain missing data records, which were removed before model building (although the algorithms can handle missing values).

model based on the maximum induced decrease in the cross-validated misclassification error. The latter quantity usually achieves a minimum for some subset of features, which then corresponds to the optimum feature subset. Their approach uses a slightly different scoring function and subset selection strategy compared to this paper.

Our feature selection procedure is also a greedy forward selection procedure, but is based on sequentially adding features based on the maximum induced decrease in \mathcal{L}_{TR} in (14). The optimal model size is then obtained from the 10-fold cross-validation estimate of the generalization error for each model size as outlined in (6). This procedure was used to obtain the results in Table (2).

However, cross-validation is computationally expensive for large data sets, and it is of interest to see how the alternative BIC penalized-likelihood approach in (7) would perform. Figures (1)-(3) show that in practically all cases, the 10-fold cross-validation and the BIC penalized-likelihood approaches yield very similar results for the optimum model size.

5 Computational Heuristics

A careful implementation of the greedy forward selection procedure using the penalized-likelihood approach described in Section (4) requires $O(J)$ data scans for evaluating Naive Bayes models with J covariate features. In each step of the greedy forward

Name	CART	Naive Bayes	
	Error rate	Error rate	# features selected
balance	0.216 ± 0.002	0.083 ± 0.011	4/4
breast(Wisconsin)	0.053 ± 0.001	0.040 ± 0.007	4/9
digit	0.315 ± 0.033	0.305 ± 0.032	7/7
flea	0.068 ± 0.003	0.040 ± 0.023	5/6
german	0.251 ± 0.014	0.247 ± 0.013	14/20
glass	0.290 ± 0.031	0.364 ± 0.033	8/9
heart	0.219 ± 0.025	0.185 ± 0.024	10/13
iris	0.128 ± 0.017	0.088 ± 0.015	12/34
letter	0.073 ± 0.021	0.033 ± 0.015	2/4
mushroom	0.000 ± 0.000	0.000 ± 0.000	12/16
segment	0.054 ± 0.004	0.087 ± 0.006	12/22
tictactoe	0.051 ± 0.007	0.266 ± 0.014	5/9
vehicle	0.300 ± 0.016	0.393 ± 0.017	8/18
votes	0.048 ± 0.010	0.044 ± 0.009	5/16
waveform	0.231 ± 0.006	0.204 ± 0.0017	11/21

Table 2. A comparison of classification errors for CART and Naive Bayes with feature selection. The optimum CART model used the 1-SE rule. The optimum number of features in the Naive Bayes model was estimated in this case by 10-fold cross-validation. For, the Naive Bayes results, the continuous covariates were all discretized into 8 uniform bins.

algorithm, a separate data scan can be used to evaluate the \mathcal{L}_{TR} values for all possible models arising from each new feature that can potentially be added to the existing set; therefore, the overall feature selection procedure requires $O(J)$ data scans.

We consider an alternative Monte Carlo heuristic for evaluating \mathcal{L}_{TR} that requires only two data scans to perform the feature selection, but yields models that are quite comparable in quality to the exact evaluation.

This heuristic is used to estimate the term \mathcal{B} in (14) using simulated pseudo-data generated from the probability models (8) and (10). Note that these probability models are completely specified from the frequency counts along with (9) and (11). Each sampled data point is obtained by first generating a y value from (9), and with this value generating all the X_j values from (11). This procedure only requires straightforward sampling from a univariate multinomial distribution.

We consider a random sample of m pseudo-data points $\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^m$, from which the term \mathcal{B} in (14) can be approximated as

$$\mathcal{B} \approx \frac{1}{m} \sum_{i=1}^m b_i \equiv \frac{1}{m} \sum_{i=1}^m \log \left(\sum_{k'=1}^K \{ \prod_{j=1}^J P_{j\tilde{x}_j, ik'} \} \pi_{k'} \right). \quad (15)$$

Here b_i is the contribution from the corresponding pseudo-data point to the sum-

mation in (15) above. The resulting Monte Carlo estimate of \mathcal{L}_{TR} in (14), and the variance of this estimate are denoted by $\tilde{\mathcal{L}}_m$ and \tilde{v}_m respectively. As new pseudo-data points b_m are generated, these estimates can be incrementally updated using the recurrences

$$m\tilde{\mathcal{L}}_m = (m-1)\tilde{\mathcal{L}}_{m-1} + b_m + \mathcal{A}, \quad (16)$$

$$(m-1)\tilde{v}_m = (m-2)\tilde{v}_{m-1} + b_m^2 + (m-1)\tilde{\mathcal{L}}_{m-1}^2 - m\tilde{\mathcal{L}}_m^2 - 2(m-1)\tilde{\mathcal{L}}_{m-1}\mathcal{A} + 2m\tilde{\mathcal{L}}_m\mathcal{A} - \mathcal{A}^2. \quad (17)$$

For large m , the Monte Carlo estimate for $\tilde{\mathcal{L}}_{TR}$ is given by $\tilde{\mathcal{L}}_m \pm \sqrt{\tilde{v}_m}$, where the bounds represent the one-standard error confidence interval (Hammersley [13]) on the asymptotic value. Given the inherent approximation in replacing $\tilde{\mathcal{L}}_{TR}$ by a simulated value, it seems excessive to obtain a very tight estimate for $\tilde{\mathcal{L}}_m$. In our experiments, we have simply used $m = N$ pseudo-data points, since that is the number of data points in the actual training data.

However, the possibility is that small number of $m \ll N$ pseudo-data points might suffice is of interest, since this would make this procedure far less expensive than the I/O cost of a separate training data scan. Furthermore, these m pseudo-data points are generated by sampling a probability distribution fitted to the entire training data. Hence, the resulting estimates will have less sampling variability than an equivalent set of m data points sampled uniformly directly from the training data.

The overall feature selection procedure using this heuristic can be implemented as follows. In the first data scan, the frequency counts in (10) and (11) are collected. Then, the Monte Carlo heuristic (16) is used to obtain estimates for \mathcal{L}_{TR} that are required in the usual greedy forward selection algorithm leading to a sequence of nested models, as described in Section (4). A second data scan is then carried out in which \mathcal{L}_{TR} is evaluated exactly using (14) for each model in this sequence. Finally the BIC penalized-likelihood criterion (7) is used to identify the optimal model size as in Section (4).

The results using this heuristic approach often yield the same results for the optimum model size as the more exact approach used for the results in Table (2). Surprisingly, for some data sets, e.g. digit, the Monte Carlo heuristic gives the same estimates for \mathcal{L}_{TR} as the exact evaluation. However, even when the two estimates differ, it is often the case that the sequence in which features are added in the greedy forward selection is quite similar. This is particularly true when the number of features is less than the eventual optimum number of features, as ascertained from the second training data scan. This may be explained by the fact that the estimated values of \mathcal{L}_{TR} tend to level off as conditionally correlated features are introduced into the Naive Bayes model as covariates. This is also precisely the point at which the training data, which contains these conditionally correlated features displays properties that cannot be obtained from the simulated pseudo data of the generative Naive Bayes probability model, however large the simulation sample size. Then the values of \mathcal{L}_{TR} in (16) cannot be reliably estimated from the Monte Carlo pseudo data in that case, but since this divergence occurs only beyond the value of the optimum model size, the subsequent ordering of the features produced by the heuristic is not important.

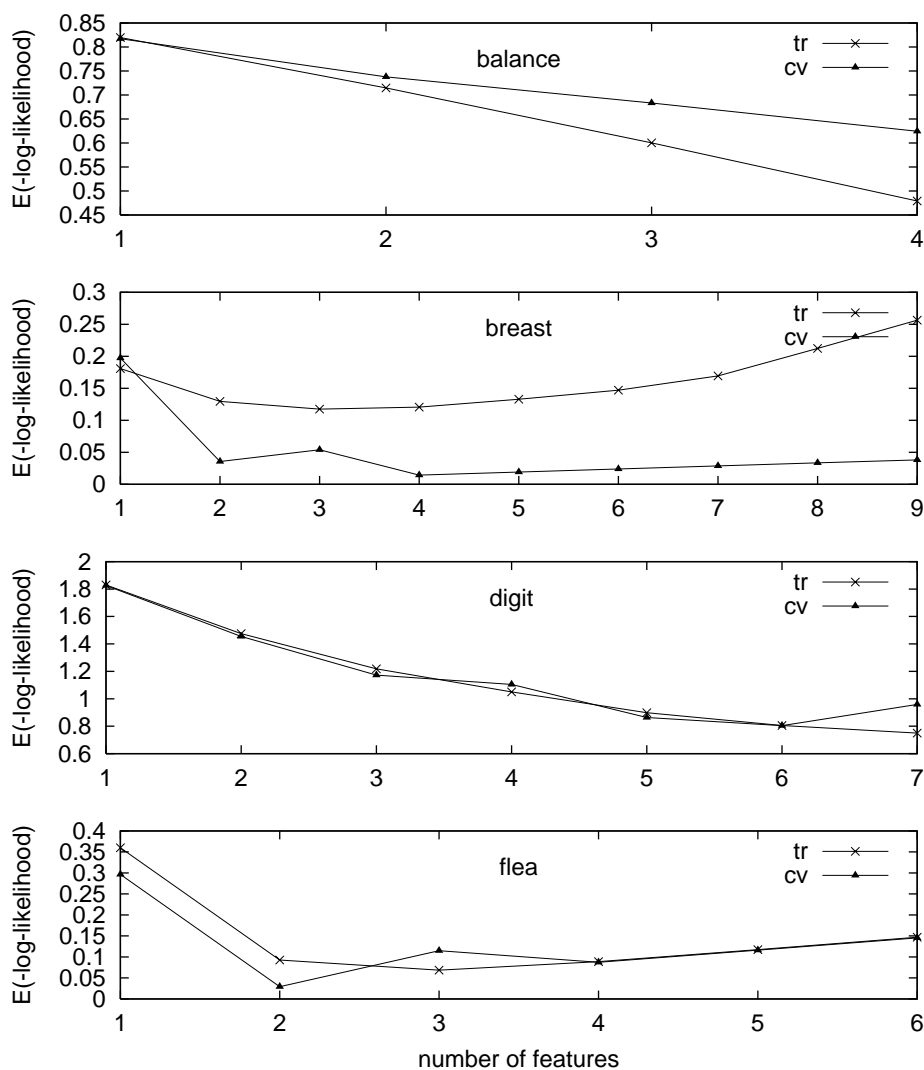


Figure 1. The empirical negative log-likelihood estimates for Naive Bayes models for the balance, breast, digit and flea data sets from the training data including training data with penalization (tr) and 10-fold cross-validation (cv).

6 Summary Remarks

The scale-up of statistical modeling algorithms to massive data sets is of great interest to the data mining community, and several recent studies have focused on various aspects of this problem. For example, Graefe, Fayyad and Chaudhuri [12], and Moore and Lee [21] have considered efficient algorithms for evaluating sufficient statistics for statistical modeling algorithms. Bradley, Fayyad, and Reina

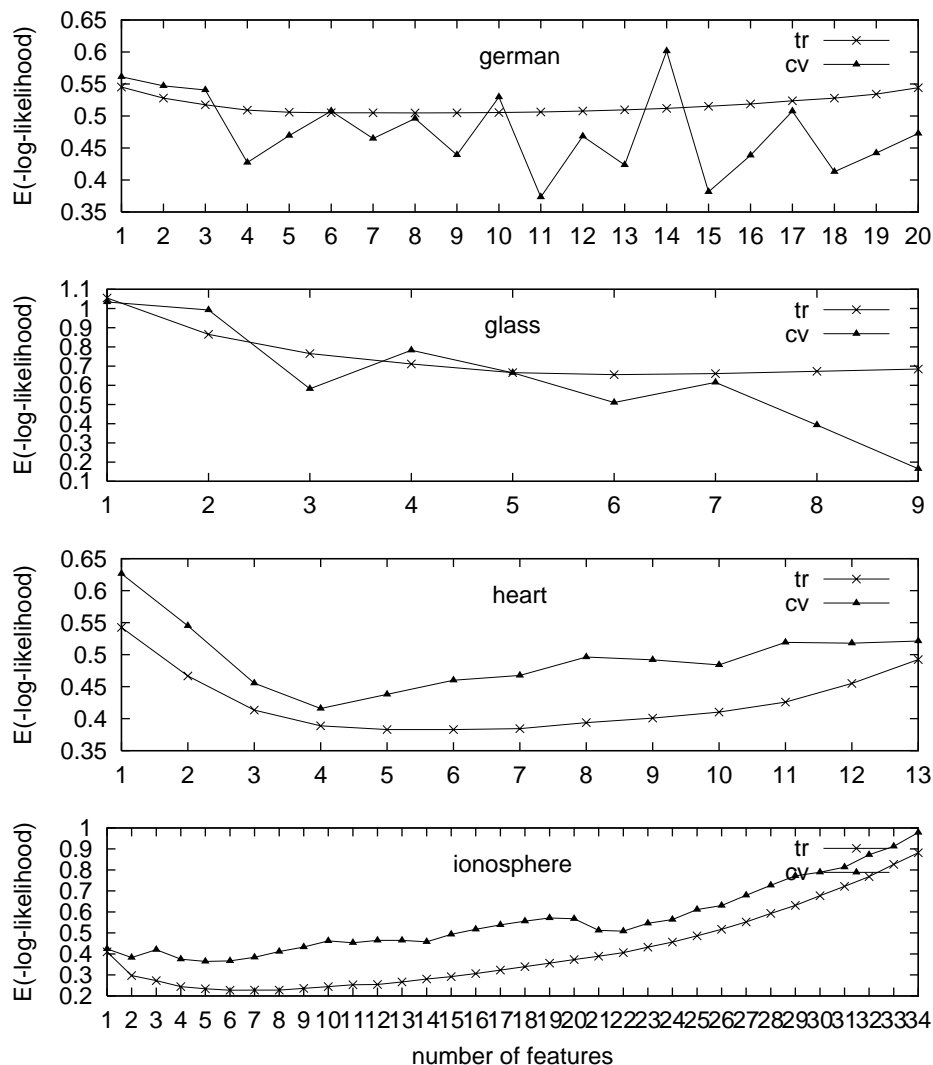


Figure 2. The empirical negative log-likelihood estimates for Naive Bayes models for the german, glass, heart, ionosphere data sets from the training data including training data with penalization (tr) and 10-fold cross-validation (cv).

[4], and DuMouchel et. al [7] have considered methods that use compressed data representations that are better than the usual random sub-sampling for producing statistical models.

Our approach is similar in spirit to these previous studies, and is based on constructing a simple probabilistic model for which sufficient statistics can be easily obtained from the massive training data set. This model is then used to generate

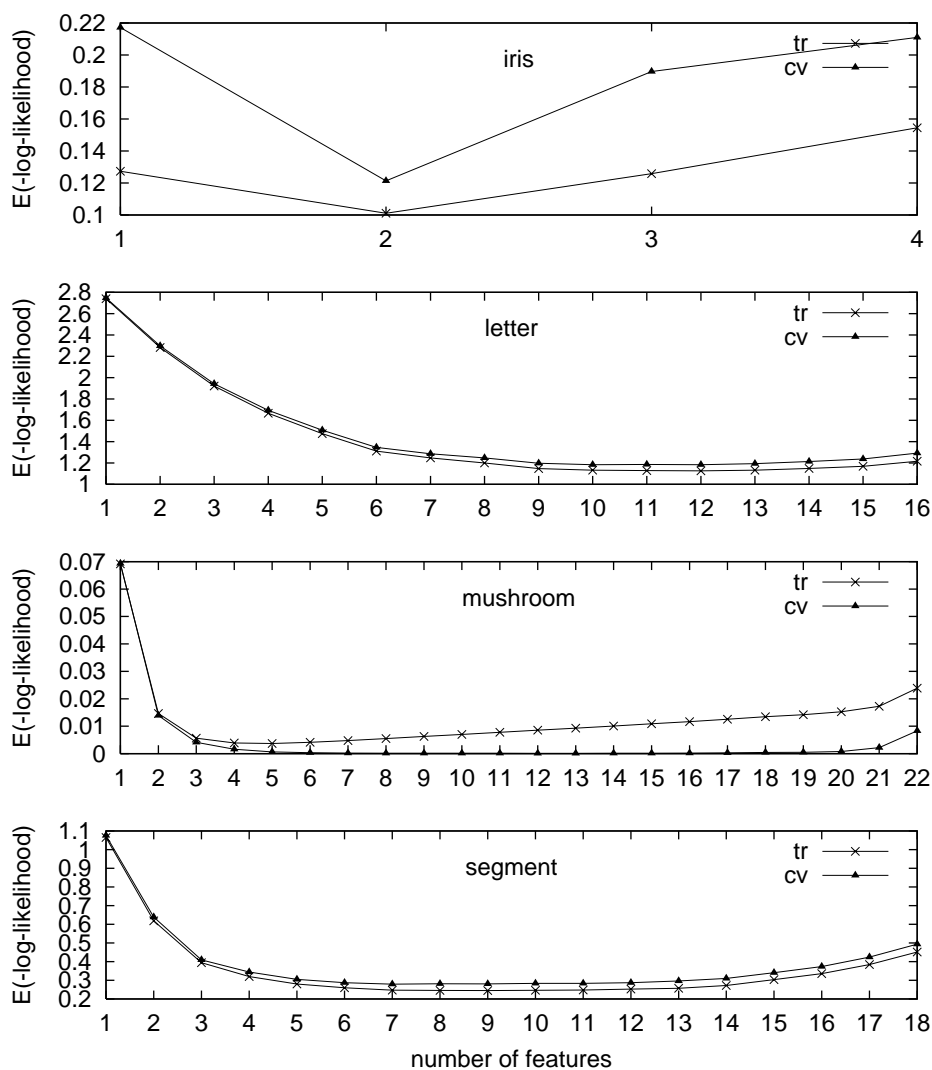


Figure 3. The empirical negative log-likelihood estimates for Naive Bayes models for the iris, letter, mushroom, segment data sets from the training data including training data with penalization (*tr*) and 10-fold cross-validation (*cv*).

simulated pseudo data for some parts of the overall statistical modeling procedure. This approach was illustrated for training a Naive Bayes model with feature selection. Here the usual algorithm would have required numerous training data scans, but using this heuristic it was possible to obtain results of comparable accuracy with only two data scans. The performance characterization of this heuristic depends significantly on several factors, including the assumptions used to build the

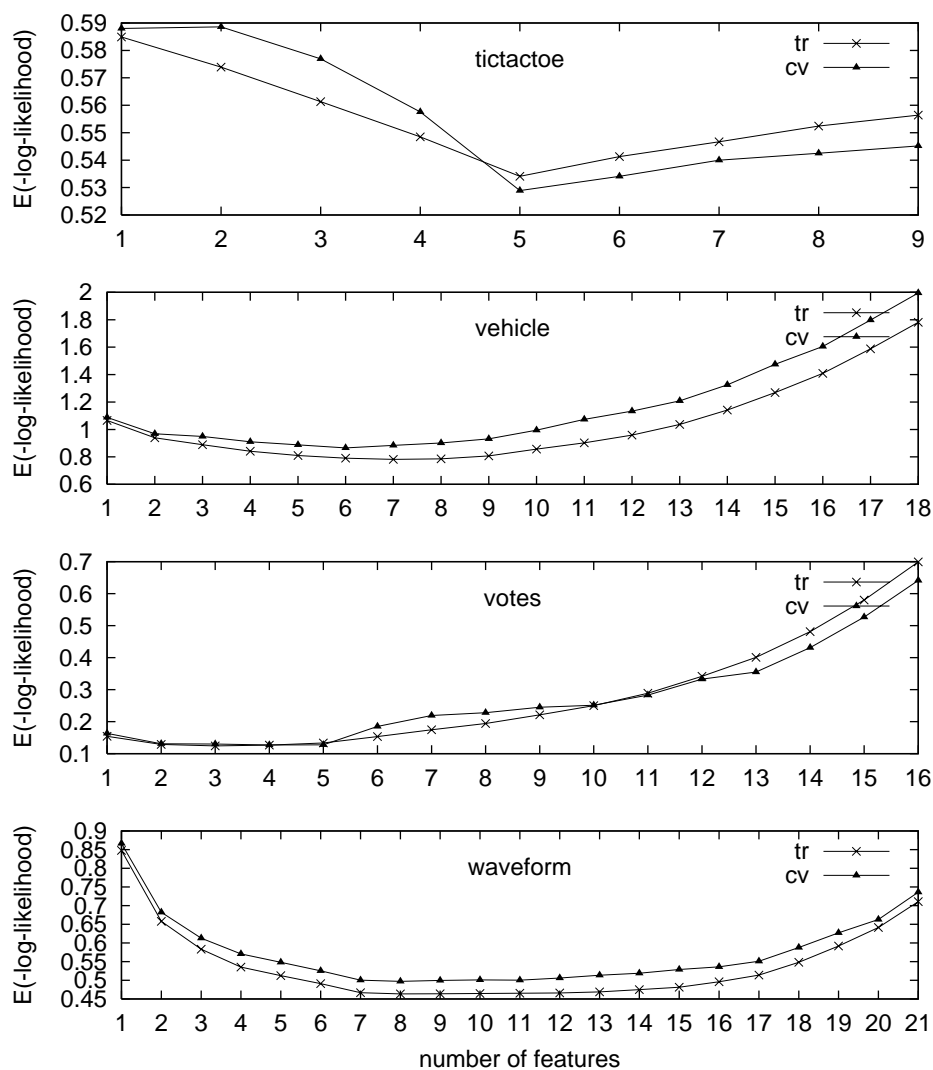


Figure 4. The empirical negative log-likelihood estimates for Naive Bayes models for the tictactoe, vehicle, voting, waveform data sets from the training data including training data with penalization (tr) and 10-fold cross-validation (cv).

simple probability model, the speed and simplicity of sampling from this model, and the number of sampled pseudo data points that are required to obtain reasonably accurate values for the estimates of interest. While we have not fully characterized this performance benefit in this paper, our experience with it in modeling certain large data sets has indicated that it is invaluable in reducing the computational cost without requiring *a priori* approximations that lead to grossly inaccurate models.

Bibliography

- [1] A. Blum and P. Langley, *Selection of Relevant Features and Examples in Machine Learning*, Artificial Intelligence, 97, pp. 245-271 (1997).
- [2] A. Agresti, *Categorical Data Analysis*, John Wiley, New York (1990).
- [3] C. Blake, E. Keogh and C. Merz, UCI repository of machine learning databases. (<http://www.ics.uci.edu/mllearn>).
- [4] P. S. Bradley, U. Fayyad and C. Reina, *Scaling Clustering Algorithms to Large Databases*, Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 9-15, (1998).
- [5] P. Brazdil and J. Gama, Statlog project datasets, <http://www.nccp.up.pt/liacc/ML/statlog>.
- [6] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*, Wadsworth, Belmont CA (1984).
- [7] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes and D. Pregibon, *Squashing Flat Files Flatter*, Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining, pp. 6-15, (1999).
- [8] P. Domingos and M. Pazzani, *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*, Machine Learning, Vol. 29, pp. 103-130 (1997).
- [9] J. H. Friedman, *On Bias, Variance, 0/1 - loss and the Curse-of-Dimensionality*, Machine Learning, J. Data Mining and Knowledge Discovery, 1, pp. 55 (1997).
- [10] N. Friedman, D. Geiger and M. Goldszmidt, *Bayesian Network Classifiers*, Machine Learning, Vol. 29, pp. 1-37 (1997).
- [11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Second Edition, Johns Hopkins University Press, Baltimore (1989).
- [12] G. Graefe, U. Fayyad and S. Chaudhuri, *On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases*, Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 204-208, (1998).

- [13] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, John Wiley, New York (1964).
- [14] C. N. Hsu, J. J. Kuang and T. T. Wong, *Why Discretization Works for Naive Bayesian Classifiers*, Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufman, San Francisco, pp. 399-406 (2000).
- [15] G. John, R. Kohavi and P. Pflieger, *Irrelevant features and the subset selection problem*, Proc. of the Eleventh International Conference on Machine Learning, Morgan Kaufman, San Francisco (1994).
- [16] G. John and P. Langley, *Estimating Continuous Distributions in Bayesian Classifiers*, Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338-345 San Mateo CA (1995).
- [17] R. Kohavi and M. Sahami, *Error-Based and Entropy-Based Discretization of Continuous Features*, Proc. of the Second International Conference on Knowledge Discovery and Data Mining, pp. 114-119 (1996).
- [18] I. Kononenko, *Semi-Naive Bayesian Classifier*, Proc. of the Sixth European Working Session on Learning, Pittman, Porto, Portugal, (1991).
- [19] P. Langley and S. Sage, *Induction of Selective Bayesian Classifiers*, Proc. of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, San Francisco pp. 399-406 (1994).
- [20] P. Langley, W. Iba and W. Thomson, *An analysis of Bayesian Classifiers*, Proc. of the Tenth National Conference on Artificial Intelligence, pp. 223-228, MIT Press, Cambridge, (1992).
- [21] A. W. Moore and M. S. Lee, *Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets*, Journal of Artificial Intelligence Research, p. 8 (1998).
- [22] M. Pazhani, *Searching for Attribute Independence in Bayesian Classifiers*, Proc. of the Eleventh Conference on Artificial Intelligence and Statistics, (1995).
- [23] G. Schwarz, *Estimating the dimension of a model*, The Annals of Statistics, 6 (1985), pp. 461-464.