The Learning Curve Method Applied to Clustering

Christopher Meek, Bo Thiesson, and David Heckerman
Microsoft Research
One Microsoft Way
Redmond, WA 98052
{meek,thiesson,heckerma}@microsoft.com

Abstract

We describe novel fast learning curve methods — methods for scaling inductive methods to large data sets – and their application to clustering. We describe the decision theoretic underpinnings of the approach and demonstrate significant performance gains on two real-world data sets.

Keywords: Learning Curve Method, Clustering, Segmentation, Scalability

1 Introduction

In situations where one has access to massive amounts of data, the cost of building a statistical model can be significant if not insurmountable. A common practice is to build the model on the basis of a sample of the data. However, the choice of the size of the sample to use is far from clear. In this paper, we describe the learning curve method, an approach to choosing the size of sample to use for training, and its application to the problem of clustering large data sets. Learning curve methods rely upon two basic observations; first, the computational cost increases as a function of the size of the training data and second, the performance/accuracy of a model has diminishing improvements as a function of the size of the training data. The curve describing the performance as a function of the size of the training data is often called the learning curve. The typical shape of a learning curve is concave with performance approaching some limiting behavior. This suggests that one can often significantly reduce the cost of training a model without significantly reducing the performance of the resulting model by simply reducing the amount of data used to train the model. The goal of a learning curve method is to balance the computational cost of training a model from data with the benefits of increases in accuracy.

We describe the learning curve method and its application to the problem of learning a clustering model. Unlike previous applications of this approach to scaling learning methods, our focus is on how one can adapt the training policy, the method by which the training algorithm is applied to subsets of the data. One idea that we investigate is the use of computationally fast but crude training methods to determine the size of the sample to use for training. For an iterative training method such as the Expectation-Maximization algorithm, one can run the algorithm a fixed number of iterations or run the algorithm to a convergence threshold at which the statistical model is only partially trained. Additionally, we consider using the results of the training algorithm obtained on smaller data sets as the initialization of the training algorithm for larger data sets. Using these basic ideas, we provide several simple efficient methods for choosing the amount of data for training a clustering model. We demonstrate significant computational performance gains on two real-world data sets obtaining, roughly, a 5 to 20 fold speedup when using these methods.

2 The Learning Curve Method

The basic idea of a learning curve method is to iteratively apply a training algorithm to larger and larger subsets of the data until the future expected costs outweigh the future expected benefits associated with the training. There are three main components of a learning curve method. The first component is the *data policy*; the schedule by which one uses portions of the data set to train a model. The second component is the *training policy*, which defines how one applies a training algorithm to the data. The final component is the *convergence criterion*, which is how one determines that the marginal cost associated with training exceeds the marginal benefit of improved performance. Each of these will be discussed in more detail below.

2.1 Data Policy

Two types of fixed data policies have been considered. John and Langly (1996) consider incrementally adding a fixed number of data points and Provost, Jensen, and Oates (1999) consider incrementally adding a geometrically increasing number of data points. As argued by Provost et al., when one does not have an accurate guess as to the "correct" number of data points to achieve the proper cost/benefit tradeoff, the method of incrementally adding a fixed number of data points can require an unreasonable number of iterations when a large number of data points is needed. In contrast, when using a geometric schedule, one can quickly reach an appropriate number of data points. For instance, if the cost of training is roughly linear in the number of data points, then using a geometric schedule to train on data sets of size $k * 2^0, k * 2^1, \ldots, k * 2^i$ until we reach some data set of size $k * 2^i$ ($N < k * 2^i < 2N$) will require only a constant factor more computation than simply applying the training method to the data set of N data points.

An alternative approach is to adaptively choose the number of data points for consideration by modeling the shape of the learning curve.

In this work we evaluate a geometric data policy. We label the successive data sets $D_1, \ldots, D_n$ where $D_i \subset D_j$ if $i < j$.

2.2 Training Policy

The training policy is the method used when evaluating the subsets of training data, $D_1, \ldots, D_n$. By carefully choosing this method it is possible to gain significant increases in performance, that is, one can significantly reduce the amount of time it takes to identify the number of data points N_{lc} needed to adequately train the model and, thus, reduce the amount of time needed to train the model. Note that the training policy used while determining N_{lc} might not correspond to the training policy used to obtain the final model using the N_{lc} data points. If alternative training policies yield similar learning curves then one can choose a computationally efficient policy to select a number of data points for training which would be similar to the number chosen by a computationally more expensive policy.

For the application of the learning curve methods to clustering we consider two aspects of the training policy. First, we consider alternative convergence thresholds and alternative fixed numbers of iterations of an iterative learning algorithm. In this paper, we use the Expectation-Maximization (EM) algorithm. Second, we consider the reuse of parameter estimates from previous stages of processing. We denote the parameters obtained from training a model on subset D_i by $\theta(D_i)$.

2.3 Convergence Policy

The convergence policy is the method by which we decide that we have identified the number of data points needed to adequately train the statistical model. It is natural to view this component from a decision theoretic perspective. Given a fixed training and data policy, how does one balance the tradeoff between the cost of training and the benefit of improved performance.

In the case of clustering, it is natural to measure the expected cost of training in terms of the expected time it will take to train on the next data set. Alternatively, when using the EM algorithm, the time is roughly linear in the size of the data set and, thus, one can measure cost in terms of the size of the next data set. We assume that the cost is linear in the size of the data set (i.e. roughly linear in time). Thus, after evaluating data set D_n the cost to evaluate/train on the next data set would be $|D_{n+1}|$.

Again, in the case of clustering, it is natural to evaluate the benefit in terms of the performance on holdout data. We use the log-likelihood of the model on holdout data, $l(D_{ho}|\theta(D_i))$. There are a variety of natural measures of expected benefit. For our analysis, we assume that the expected benefit is linear in the relative improvement in holdout score between two most recent data sets and the improvement in holdout score between the most recent data set and a baseline model, $\theta_{base}(D_1)$.

Thus, under these assumptions, we choose to terminate the learning curve method after evaluating data set D_n when the *learning curve convergence measure*, the ratio of benefit over cost, drops below the (learning curve) convergence threshold, λ, that is,

$$\frac{l(D_{ho}|\theta(D_n)) - l(D_{ho}|\theta(D_{n-1}))}{l(D_{ho}|\theta(D_n)) - l(D_{ho}|\theta_{base}(D_1))} \frac{1}{|D_{n+1}|} < \lambda. \quad (1)$$

When the ratio of the benefit over cost drops below this convergence threshold we say that the (learning curve) convergence criterion is satisfied.

In our experiments we choose the baseline model to be a model in which all of the features are mutually independent. Alternative policies have been described by John and Langley (1996) and Provost et al. (1999). Our policy is simple but potentially sensitive to local variations in the learning curve. Fortunately, our experiments, described below, suggest that learning curves for clustering models are usually smooth. In situations where the learning curves are not smooth, the alternative policies suggested by John and Langly and Provost et al. may be useful.

MSNBC

method	Sample size 40000	80000	160000	320000	497971
fixed-1	0.002363	0.000472	0.000099	0.000017	0.000003
fixed-3	0.001281	0.000299	0.000065	0.000016	0.000003
fixed-5	0.001060	0.000244	0.000051	0.000017	0.000002
fixed-10	0.001134	0.000289	0.000056	0.000010	0.000005
thres-0.1	0.001577	0.000350	0.000078	0.000018	0.000005
thres-0.01	0.000372	0.000241	0.000046	0.000015	0.000003
thres-0.001	0.001855	0.000210	0.000070	0.000022	0.000002
thres-0.0001	0.002332	0.000399	0.000085	0.000027	0.000003
naive	0.002342	0.000425	0.000131	0.000019	0.000000

MS.COM

method	Sample size 40000	80000	160000	320000	640000	1280000	1838877
fixed-1	0.026042	0.003178	0.000371	0.000141	0.000036	0.000007	0.000003
fixed-3	0.006356	0.001501	0.000347	0.000081	0.000024	0.000003	0.000002
fixed-5	0.003913	0.001363	0.000336	0.000067	0.000021	0.000003	0.000001
fixed-10	0.002515	0.000998	0.000319	0.000073	0.000020	0.000004	0.000001
thres-0.1	0.006356	0.001501	0.000347	0.000081	0.000024	0.000003	0.000002
thres-0.01	0.002628	0.000576	0.000237	0.000099	0.000020	0.000005	0.000001
thres-0.001	0.001421	0.001017	0.000351	0.000061	0.000016	0.000004	0.000001
thres-0.0001	0.001449	0.000910	0.000373	0.000081	0.000026	0.000002	0.000001
naive	0.002307	0.000857	0.000393	0.000094	0.000029	0.000004	0.000001

Table 1: Values for the learning curve convergence measure at sample sizes given by the data policy.

3 Methods, Models, and Experimental Results

In this section, we evaluate several different learning curve methods for the problem of clustering large data sets. As described above, each of the methods utilizes the geometric fixed data policy and continues to evaluate larger data sets until the learning curve convergence criterion is satisfied. Each of the learning curve methods is distinguished only on the basis of the training policy and not the convergence or data policies.

We investigate a simple but widely used class of models for clustering, namely finite mixture models, where each component is defined by a log-linear model with only main effects for all variables in the data set. These models can alternatively be viewed as naive-Bayes models with a hidden class variable, also known as AutoClass models (Cheeseman and Stutz, 1995).

We use the EM algorithm to train the mixture models. We initialize the algorithm by estimating the parameters for the baseline model and then randomly perturb parameter values by a small amount to obtain a parameterization for each mixture component. See Thiesson, Meek, Chickering, and Heckerman (1999) for further details. The convergence criterion that we use to terminate the EM algorithm is the following. We converge when the relative improvement in log-likelihood of the training data between successive EM iterations relative to the total improvement in log-likelihood over the initial model is less than the EM convergence threshold γ. Typically when running the EM algorithm, one runs the algorithm to a convergence threshold that is quite low. In our experiments we use $\gamma_{final} = 10^{-5}$ when we train the mixture model after having used a learning curve method to determine the adequate number of data points N_{lc} to be used in the final training of the model.

Our benchmark learning curve method is the LC_{naive} method which runs the EM algorithm to EM convergence threshold γ_{final} on each data set using the same initial parameterization until the (learning curve) convergence criterion λ is satisfied. The LC_{fixed} methods runs the EM algorithm for a fixed number of iterations using the same initial parameterization. The LC_{thres} methods runs the EM algorithm to a EM convergence threshold $\gamma_{lc} > \gamma_{final}$ on each of the data sets D_i using the same initial parameterization for each data set. The LC_{naive} method corresponds to LC_{thres} with threshold 10^{-5}. Our final types of method are the $\mathrm{LC}^{reuse}_{fixed}$ and $\mathrm{LC}^{reuse}_{thres}$ methods which are similar to the LC_{fixed} and LC_{thresh} methods. They differ in the following two ways. First, parameter values $\theta(D_{n-1})$ (except those parameter values associated with component mixture weights which are set to be uniform) obtained from the previous iteration in the learning curve method are used to initialize the EM algorithm for data set D_n. Second, when reusing parameter values, some cluster components loose all of their support due to the size of the initial data sets. To alleviate this premature component starvation we identify components that have little or no support (less than one case) and reset the component parameterization to its initial parameterization.

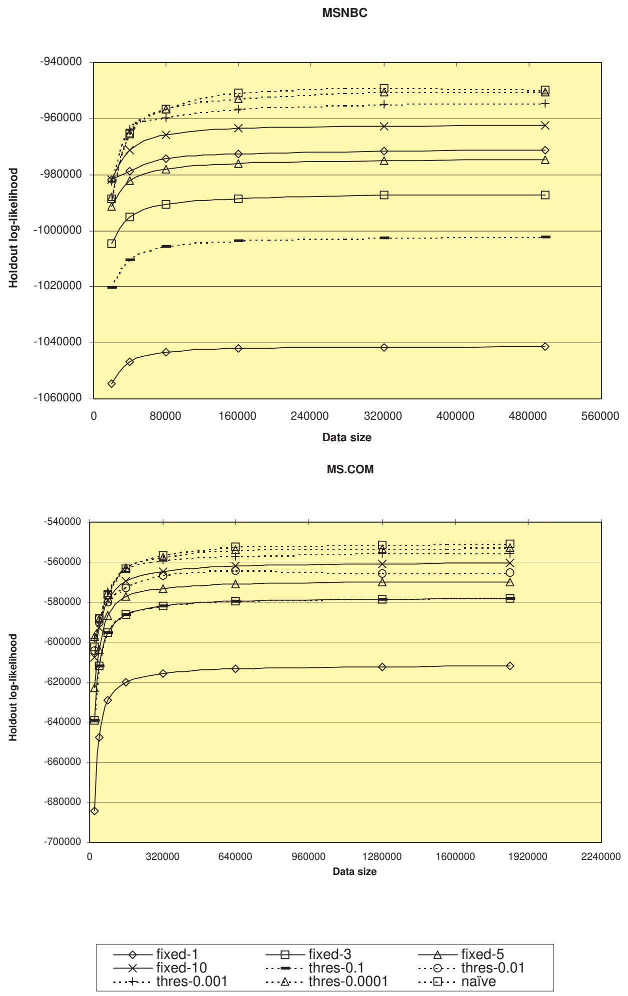

Figure 1: LC_{fixed} and LC_{thres} learning curves for the MSNBC and MS.COM data sets.

3.1 Data Sets

We evaluated the learning curve methods on two real-world data sets. The *MSNBC* data set, is derived from web logs for one day in 1998 for the MSNBC website. It records which of the 303 most popular stories on that day each of the visitors read. The *MS.COM* data set is derived from the web logs for one day in 2000 for the microsoft.com web site. It records which areas or "vroots" of the site each of the user visited among the 775 most popular sites. In each of the data sets, users correspond to cases and items possible viewed correspond to variables. Both data sets are sparse in the sense that on average a user only views a few items. The MSNBC data set contains 597,971 users and the MS.COM data set contains 1,938,877 users. The data sets were partitioned into training and holdout sets at random. In both cases, we used 100,000 users for the holdout set and the remaining users for the training set. The holdout set is used during both the evaluation of the learning curve convergence criterion and the evaluation the final holdout score after the EM algorithm is run to EM convergence threshold of γ_{final} for the selected number of cases N_{lc}.

We have investigated mixture models with 25, 50, and 100 components. All experiments were qualitatively similar and demonstrated the same trends. Hence, in this paper, we only report results for the 25 component mixture model.

3.2 Results

We first concentrate on the LC_{fixed} and LC_{thres} methods. Figure 1 shows learning curves for these methods on the MSNBC and MS.COM experiments. The number of iterations used for the LC_{fixed} and the threshold used for the LC_{thres} are indicated to the right of 'fixed' and 'thres' in the legend for the figure. For the MS.COM data set, the thres-0.1 curve is identical to the fixed-3 curve and can therefore not be distinguished in the figure.

We see that all LC_{fixed} methods display same behavior as the LC_{naive} method. In particular, we notice that the one-step LC_{fixed} method shows this behavior, which implies that the learning curve convergence can be detected efficiently for clustering models. Being able to quickly evaluate the adequacy of alternative subsets of training data allows one to more easily use other convergence policies such as the LRLS policy suggested in Provost, Jensen, and Oates (1999).

From Figure 1 we also notice that the learning curves for LC_{thres} methods and the LC_{naive} method are similar in shape. In our experience, the LC_{thres} curves are noisier and do not track the LC_{naive} curve as well as the LC_{fixed} methods. The explanation for this is that EM convergence is only evaluated after a complete pass through the data and successive steps of the LC_{thres} may run the EM algorithm a different number of iterations. This difference can have a dramatic effect on the resulting parameterization (especially when the EM convergence level γ_{lc} is high) and, hence, a dramatic effect on the log-likelihood score for the holdout set. Since our convergence policy is based on local tests, this behavior may occasionally force the algorithm to terminate early and choose a sample size that is too low. Alternative convergence policies might alleviate this difficulty with the LC_{thres} methods.

It is useful to compare the learning curve convergence measures for the alternative learning curve methods. If the convergence measures for the alternative methods follow the convergence measures of the LC_{naive} method for different sample sizes, then the methods will likely choose identical sample sizes to be adequate for training, that is, the N_{lc} chosen by the methods will be similar. Table 1 demonstrates that the convergence policy in Equation (1) has this property. For our two data sets, there are many values of the convergence threshold for which all of the LC_{thres} and LC_{fixed} learning curve methods will select the same sample size, and some thresholds for which the adequate sample size varies by only one step in the data policy. For instance, for LC convergence level $\lambda = 0.0005$, all but one method agree on 80,000 cases for MSNBC, and for all MS.COM experiments the adequate sample size is 160,000 cases.

To present performance results for the learning curve methods we introduce the following additional notation. Let *EM-full* denote the method which runs EM to EM convergence level γ_{final} on the full data set. The elapsed time to run a learning curve method to convergence is the time needed to choose the number of data points N_{lc} plus the time needed to run EM to convergence level γ_{final} on those N_{lc} data points. On the final run of EM we use the parameters obtained during the last step of the learning curve method as the initial values for the EM algorithm. We compute the *speedup factor* as the runtime for EM-full to reach convergence divided by the elapsed time for a learning curve method to reach convergence. To compare the quality of the learned models we also compute the *holdout score*: $\log p(D_{ho}|\hat{\theta}_{lc})$, where $\hat{\theta}_{lc}$ denote the estimate obtained by the particular method. Methods that choose identical sample sizes N_{lc} will yield identical holdout scores. Finally, to measure the cost of using the learning curve method we compute the *overhead ratio* as the elapsed time to run the method to convergence divided by the runtime for the standard EM algorithm when run on the adequate sample size N_{lc} to an EM convergence level of γ_{final}.

Table 2 shows the adequate sample sizes, test scores, speedup ratios, and overhead ratios that we obtain for MSNBC and MS.COM when training the 25 component mixture models. Results shown are for the learning curve convergence threshold $\lambda = 0.0005$. Of course, a higher convergence threshold will tend to select a smaller N_{lc} and provide more significant speedups; the choice of λ is our cost/benefit tradeoff. As suggested by Table 1 all methods (approximately) agree on the adequate sample size N_{lc}.

The speedup factor for a learning curve method depends on both the size and other features of the data set. By choosing larger data sets one can arbitrarily improve the speedup factor for the full EM algorithm comparison for a fixed λ. Hence, the speedup numbers in the table do not express the obtainable computational benefit from using learning curves methods, but provide us with a way to compare the different methods. Aside from difference in the selected size of N_{lc}, the one-step method is the most efficient of the LC_{fixed} and LC_{thres} methods. Each of the methods provides a significant speedup.

MSNBC

method	N_{lc}	holdout score	speedup factor	overhead ratio
fixed-1	80000	-956531	4.9	1.03
fixed-3	80000	-956531	4.8	1.05
fixed-5	80000	-956531	4.8	1.06
fixed-10	80000	-956531	4.6	1.10
thres-0.1	80000	-956531	4.9	1.04
thres-0.01	40000	-965492	10.4	1.08
thres-0.001	80000	-956531	4.0	1.28
thres-0.0001	80000	-956531	3.3	1.52
naive	80000	-956531	3.0	1.66
EM-full	497971	-949953	1.0	1.00

MS.COM

method	N_{lc}	holdout score	speedup factor	overhead ratio
fixed-1	160000	-563194	19.7	1.02
fixed-3	160000	-563194	19.5	1.04
fixed-5	160000	-563194	19.2	1.05
fixed-10	160000	-563194	18.6	1.09
thres-0.1	160000	-563194	19.5	1.04
thres-0.01	160000	-563194	18.4	1.10
thres-0.001	160000	-563194	15.9	1.27
thres-0.0001	160000	-563194	13.9	1.45
naive	160000	-563194	10.6	1.90
EM-full	497971	-550914	1.0	1.00

Table 2: Adequate sample sizes, holdout scores, speedups, and overheads for the LC_{fixed} and LC_{thres} learning curve methods.

The overhead ratios in Table 2 provides us with a guide to the overhead of applying the learning curve method to clustering. The ratios show that several of the LC methods evaluated in this paper have very little overhead. The overhead ratio is sensitive to the choice of γ_{final}. By choosing γ_{final} to be larger, the final run of EM would likely run fewer iterations and the relative amount of time that is spent determining N_{lc} would increase making the overhead ratio larger. For $\gamma_{final} = 10^{-5}$, the LC_{fixed} method using a single EM iteration has a overhead ratio close to one. Despite the sensitivity to γ_{final}, the impressive overhead ratios are due, in part, to the effective use of a training policies to identify the adequate number of data points. This can be seen in the large difference between the overhead ratios of the LC_{naive} method and the other LC_{fixed} and LC_{thres} methods. The relative importance of alternative training policies can also be seen in the large difference between speedup factors between the LC_{naive} and the other LC_{fixed} and LC_{thres} methods.

Now we consider the $\mathrm{LC}^{reuse}_{fixed}$ and $\mathrm{LC}^{reuse}_{thres}$ methods. The results for these methods are not as regular as compared to the other methods. In particular, both $\mathrm{LC}^{reuse}_{fixed}$ and $\mathrm{LC}^{reuse}_{thres}$ methods skew the selection of the adequate sample size towards larger sample sizes — sometimes significantly larger — than the one we obtain from the LC_{naive} method. For the same convergence threshold $\lambda = 0.0005$, as used for the experiments reported above, all but one of the reuse LC methods selects an adequate sample size of 320,000 for MSNBC and all methods select a sample size of 640,000 as adequate for MS.COM. This indicates that the learning curves for the reuse methods have a significantly different shape than the LC_{naive} learning curve.

We have found that with learning curve convergence level $\lambda = 0.005$, all LC^{reuse}_{fixed} and LC^{reuse}_{thres} methods (approximately) select the same sample size as LC_{naive}. We currently do not have any insight about how the LC convergence threshold for the reuse LC methods scale with the convergence threshold for the naive method.

Table 3 shows the adequate sample sizes, test scores, speedups, and overheads that we obtain for MSNBC and MS.COM where we have used LC^{reuse}_{fixed} and LC^{reuse}_{thres} methods with learning curve convergence threshold $\lambda = 0.005$ to train models with 25 clusters.

MSNBC

method with reuse	N_{lc}	holdout score	speedup factor	overhead ratio
fixed-1	160000	-951241	1.5	1.13
fixed-3	80000	-954587	3.2	1.57
fixed-5	80000	-955439	4.8	1.06
fixed-10	80000	-957846	5.0	1.00
thres-0.1	160000	-950591	1.5	1.14
thres-0.01	80000	-955539	4.5	1.13
thres-0.001	80000	-960141	4.7	1.07
thres-0.0001	80000	-958500	4.2	1.21
thres-0000001	80000	-958252	3.7	1.38
EM-full	497971	-949953	1.0	1.00

MS.COM

method with reuse	N_{lc}	holdout score	speedup factor	overhead ratio
fixed-1	160000	-561997	20.9	0.96
fixed-3	160000	-565041	17.8	1.13
fixed-5	160000	-566180	14.5	1.39
fixed-10	160000	-566909	25.3	0.80
thres-0.1	160000	-564072	14.3	1.41
thres-0.01	80000	-580073	45.7	0.71
thres-0.001	160000	-568619	28.1	0.72
thres-0.0001	160000	-568656	18.2	1.11
thres-0.00001	160000	-568699	24.3	0.83
EM-full	497971	-550914	1.0	1.00

Table 3: Adequate sample sizes, holdout scores, speedups, and overheads for the LC^{reuse}_{fixed} and LC^{reuse}_{thres} learning curve methods.

Results for the reuse methods show improved speedup for most of the methods (aside from differences in the selected size of N_{lc}), and surprisingly, even compared to the one-step LC_{fixed} method. In most cases, the improved efficiency has, however, the cost of additional reduced log-likelihood scores on the holdout set, even for the same N_{lc}. This suggests that reusing the previous parameterizations can drive the algorithm towards convergence more quickly resulting in fewer iterations being needed for the final run of the EM on the N_{lc} data points. However, this gain comes at the cost of a final parameterization with lower overall holdout score — even when being smart about the initialization, as described in Section 3.

4 Related and Future Work

Learning curve methods are a natural way to improve the scalability of a learning algorithm. In this paper, we have described the application of learning curve methods to the problem of identifying good clusters of data for a fixed number of mixture components. There are many areas for future investigation. One interesting area for future work is to adapt these learning curve methods to simultaneously select the number of clusters in the model and the size of the data set. However, one might expect that, by increasing the size of the data set, one increases the need for additional clusters; with more data you might need more components.

In this paper, we have limited the application of decision theory to the convergence policy. It might be useful to consider decision theoretic approaches of controlling the data and training policies. In addition, alternative adaptive data and training policies should be investigated. Additional investigation of the connection between the learning curve convergence measures for the LC_{naive} and reuse LC methods are needed. Finally, our approach of using crude computationally efficient training methods for determining the appropriate number of data points to use for training should be evaluated for alternative iterative training methods (e.g. stochastic gradient descent, Newton-Raphson) and for alternative statistical models (e.g. classification and regression models).

References

Cheeseman, P., & Stutz, J. (1995). Bayesian classification (AutoClass): Theory and results. In Fayyad, U., Piatesky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 153–180. AAAI Press, Menlo Park, CA.

John, G., & Langley, P. (1996). Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 367–370. AAAI Press.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pp. 23–32. ACM.

Thiesson, B., Meek, C., Chickering, D., & Heckerman, D. (1999). Computational efficient methods for selectiong among mixtures of graphical models, with discussion. In Bernardo, J. M., Berger, J. O., Dawid, A. P., & Smith, A. F. M. (Eds.), *Bayesian Statistics 6: Proceedings of the Sixth Valencia International Meeting*, pp. 631–656. Oxford University Press.