Batch Classifications with Discrete Finite Mixtures

Petri Kontkanen, Petri Myllymäki, Tomi Silander, and Henry Tirri

Complex Systems Computation Group (CoSCo)
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland

Abstract. In this paper we study batch classification problems where multiple predictions are made simultaneously, in contrast to the standard independent classification case, where the predictions are made independently one at a time. The main contribution of this paper is to demonstrate how the standard EM algorithm for finite mixture models can be modified for the batch classification case. In the empirical part of the paper, the results obtained by the batch classification approach are compared to those obtained by independent predictions.

1 Introduction

In the standard classification approach, the model used to classify data is first constructed by using the available training data, and each classification problem is then solved independently with the produced model. In this paper, we extend this classification problem by allowing multiple predictions (classifications) to be made at the same time. In this *batch classification* case, all the classification problems are given simultaneously, and instead of dealing with a single vector to be classified, the task is to find a correct classification for a set of vectors.

The batch classification problem can be regarded as a missing data problem, where the missing data consists of the correct classifications of *query vectors*, the vectors to be classified. Intuitively, one could expect the batch classification to produce better results than independent classifications, since in the batch case the data available for making predictions consists not only of the original training data, but also of the set of all the query vectors. A closer look reveals, however, that the amount of missing data has also increased, making the missing data estimation problem more difficult. Therefore it is interesting to investigate the trade-off between the advantage of using the increased information available in the query batch, and the disadvantage of increased complexity in the search process. Similar work has been reported in [2], where the unclassified vectors were used as background knowledge for a conceptual-clustering algorithm.

In order to study this problem, we use the probabilistic model family of finite mixtures [3, 7], where the problem domain probability distribution is approximated as a finite, weighted sum of simple component distributions. The standard way to fix a finite mixture model is to estimate the values of the latent clustering

variable via the *Expectation Maximization (EM)* algorithm [1] (see, e.g., [5]), and then to choose the *maximum a posteriori probability (MAP)* parameter values.

The contribution of this paper is to demonstrate how the standard EM algorithm for finite mixtures is modified for the batch classification case, so that it can be used for estimating both the missing classification data, and the missing latent variable data at the same time. In other words, the same algorithm used normally for constructing the models from training data, is in our approach used also for making predictions. In the empirical part of the paper, we compare the results obtained by using the batch classification with the modified EM algorithm to the results obtained by the standard approach, where each query vector is classified independently.

2 Discrete finite mixtures

In the following, the problem domain is modeled by using m discrete random variables $X_1, \ldots, X_m$ (continuous variables are assumed to be discretized by quantization). A *data instantiation* $\boldsymbol{d}$ is a vector in which all the variables X_i have been assigned a value, $\boldsymbol{d} = (X_1 = x_1, \ldots, X_m = x_m)$, where $x_i \in \{x_{i1}, \ldots, x_{in_i}\}$. A *random sample* $D = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N)$ is a set of N i.i.d. data instantiations, where each $\boldsymbol{d}_j$ is sampled from the joint distribution of the variables $X_1, \ldots, X_m$.

In the discrete variable case, the *finite mixture* [3, 7] distribution for a data instantiation $\boldsymbol{d}$ can be written as

$$P(\boldsymbol{d}) = \sum_{k=1}^{K} \left(P(Y = y_k) \prod_{i=1}^{m} P(X_i = x_i | Y = y_k) \right).$$

where Y denotes a latent *clustering random variable*, the values of which are not given in the data D, K is the number of possible values of Y, and the variables $X_1, \ldots, X_m$ are assumed to be independent, given the value of the clustering variable Y.

In the following, we assume both the cluster distribution $P(Y)$ and the intra-class conditional distributions $P(X_i|Y = y_k)$ to be multinomial. Thus a finite mixture model can be defined by first fixing K, the model class (the number of the mixing distributions), and then by determining the values of the model parameters $\Theta = (\alpha, \Phi), \Theta \in \Omega$, where $\alpha = (\alpha_1, \ldots, \alpha_K)$ and $\Phi = (\Phi_{11}, \ldots, \Phi_{1m}, \ldots, \Phi_{K1}, \ldots, \Phi_{Km})$, with the denotations $\alpha_k = P(Y = y_k)$, and $\Phi_{ki} = (\phi_{ki1}, \ldots, \phi_{kin_i})$, where $\phi_{kil} = P(X_i = x_{il}|Y = y_k)$.

3 Batch classifications with the EM algorithm

In this paper we consider prediction problems where the goal is to correctly classify a set of L *test vectors* $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_L$ by using the given training data D. In the following, let X_m denote the class variable, the values of which are to be estimated, in which case all the test vectors $\boldsymbol{q}_j$ are of the form

$$\boldsymbol{q}_j = (X_1 = x_1, \ldots, X_{m-1} = x_{m-1}).$$

The standard Bayesian procedure for solving this problem is to first to construct the *maximum a posterior probability (MAP)* model $\hat{\Theta}_D$ with respect to the given data D, and then to classify the test cases independently by using the model constructed. In the Bayesian framework for a single test vector $\boldsymbol{q}_j$ this is done by determining the following *predictive distribution* for all the possible values x_m of the class variable X_m:

$$P(x_m \mid \boldsymbol{q}_j, \hat{\Theta}_D) \propto P(\boldsymbol{q}_j, x_m \mid \hat{\Theta}_D) = \sum_{k=1}^{K} P(\boldsymbol{q}_j, x_m, z_j = k \mid \hat{\Theta}_D), \qquad (1)$$

where the value of variable z_j denotes the value of the clustering variable Y corresponding to the test case $\boldsymbol{q}_j$.

Unfortunately, determining the MAP parameter values $\hat{\Theta}_D$ exactly is not possible in practice because of the missing data imposed by the latent variable Y. However, the *Expectation Maximization (EM)* algorithm [1] is an iterative algorithm that can be used for finding an approximation of the MAP model. The EM algorithm can also be understood as an unsupervised clustering algorithm, where the estimated values of the latent variable determine the (probabilistic) clusters. In the E-step of the algorithm the conditional expected values of the sufficient statistics of the complete data (D, Z) are needed, in our case the expected values of the parameters h_k and f_{kil}, where $h_k = \sum_{j=1}^{N} z_{jk}$ is the number of instantiations in cluster k, and $f_{kil} = \sum_{j=1}^{N} z_{jk} v_{jil}$ is the number of instantiations in cluster k with variable X_i having value x_{il}. Here the indicator variable z_{jk} has value 1 if $\boldsymbol{d}_j$ is sampled from $P(\cdot|Y = y_k)$, and the indicator variable v_{jil} has value 1 if $d_{ji} = x_{il}$.

The expectations of the sufficient statistics at the time step t of the EM algorithm are computed by setting $\bar{h}_k = E[h_k \mid D, \Theta^{(t)}] = \sum_{j=1}^{N} w_{jk}$, and $\bar{f}_{kil} = E[f_{kil} \mid D, \Theta^{(t)}] = \sum_{j=1}^{N} w_{jk} v_{jil}$, where

$$w_{jk} = E[Z_{jk} \mid D, \Theta^{(t)}] = \frac{\alpha_k^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{kil}^{(t)}\right)^{v_{jil}}}{\sum_{k'=1}^{K} \left(\alpha_{k'}^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{k'il}^{(t)}\right)^{v_{jil}}\right)}.$$

In the batch classification case, the predictive distribution can be written as

$$P(c_1, \ldots, c_L, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_L \mid \hat{\Theta}_{D,Q}) = \prod_{j=1}^{L} P(\boldsymbol{q}_j, c_j \mid \hat{\Theta}_{D,Q}), \qquad (2)$$

where $\boldsymbol{c} = (c_1, \ldots, c_L)$ denotes a vector consisting of classifications of all the test vectors $Q = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_L)$. Consequently, the test vectors can be classified independently also in the batch classification case, *but the MAP model $\hat{\Theta}$ must in this case be estimated by using the joint database (D, Q), not the original data D alone.* In addition to this, the missing data consists not only of the cluster indicators z_{jk}, but also of the classifications of the test vectors, $c_1, \ldots, c_L$. By conditioning the class indicator variables $C_1, \ldots, C_L$ with the cluster indicator

variables, we get $\bar{c}_{jlk} = P(C_j = l \mid Z_{jk} = 1, \boldsymbol{q}_j, \Theta^{(t)}) = \phi^{(t)}_{kml}$, where as before, m denotes the index of the class variable. The last equality follows from the fact that the attributes are assumed to be independent, given the value of the clustering variable Y.

Because of the absence of the values of the class variable X_m, the expectations of the cluster indicators corresponding to the test cases must be calculated differently than with the standard EM. The modified formulas for computing these expectations are given by

$$w_{jk} = E[Z_{jk} \mid D, \Theta^t] = \frac{\alpha^{(t)}_k \prod_{i=1}^{m-1} \prod_{l=1}^{n_i} \left(\phi^{(t)}_{kil}\right)^{v_{jil}}}{\sum_{k'=1}^{K} \left(\alpha^{(t)}_{k'} \prod_{i=1}^{m-1} \prod_{l=1}^{n_i} \left(\phi^{(t)}_{k'il}\right)^{v_{jil}}\right)}. \tag{3}$$

In addition, the expectations of the parameters f_{kml} must now be calculated by using $\bar{f}_{kml} = \sum_{j=1}^{L} w_{jk} \bar{c}_{jlk}$. Detailed derivation of these formulas is similar to the derivations used in [5], but technically somewhat involved and omitted here.

In the M-step, the parameter values are updated in such a way that the obtained expected posterior is maximized (for the update formulas, see e.g. [5]).

4 Empirical results

To validate the batch classification approach described in the previous section, we performed a series of experiments with a set of public domain classification datasets from the UCI repository[1]. For simplicity, in our experiments we used the uniform prior (Dirichlet with all the hyperparameters set to 1) for both the independent (IC) and batch classification (BC). In the independent classification case each classification query was classified one at a time by using the MAP prediction defined by formula (1), where the approximation found by the EM algorithm was taken as the MAP model $\hat{\Theta}$. In the batch classification case, the predictive distribution (2) was used instead, the difference being that the MAP model was estimated from the joint database (D, Q) by using the EM algorithm as described in Section 3. Description of the datasets used, and the crossvalidated classification results obtained can be found in Table 1. The results are averages over 100 independent crossvalidation runs, and the number of folds used was the same as in [6].

The results show that the batch classification approach does not demonstrate significant improvement over independent predictions. The reasons for this are twofold. Firstly, as discussed before, it seems likely that the increase in the amount of missing data makes the search for good local maxima in the enlarged search space much more difficult, so the theoretical advantage of using the query information is in this case nullified by the increase in the complexity of the search problem. Secondly, if the training data is already sufficient to model the joint distribution well, the auxiliary information in the query batch Q (sampled from the same distribution) cannot improve the predictions significantly.

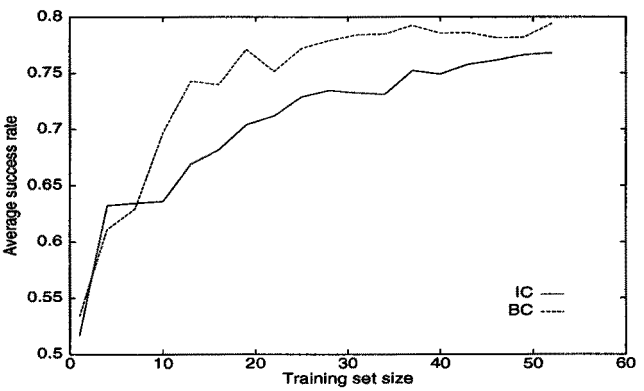
[1] http://www.ics.uci.edu/~mlearn/

Table 1. The datasets used in the experiments and the corresponding crossvalidated classification accuracies obtained.

Dataset	Size	Attrs	Classes	IC	BC
Lymphography	148	19	4	73.0	73.3
Hepatitis	150	20	2	79.5	79.6
Heart disease	270	14	2	81.9	81.9
Primary tumor	339	18	21	41.5	41.8
Australian	690	15	2	81.3	80.6
Diabetes	768	9	2	68.6	68.6

In order to test the latter hypothesis we performed a second set of experiments to see how the methods perform when the training sets D are not sufficient for building very good models, and small with respect to the size of the query batch Q. In these experiments we sampled small training sets randomly from the datasets, and used the rest of the data as the test set. For each case, classification was done by using both IC and BC methods. The average classification success rate was then plotted as a function of the size of the training set. In Fig. 1 a typical behavior can be seen. Each data point corresponds to an average of 100 independent tests.

Fig. 1. Average classification success rate as a function of the size of the training set in the Heart Disease data set case.



In these tests the results show a clear difference in the performance. The batch approach is more efficient in extracting regularities present in the data, and outperforms the standard IC approach in cases with very small amounts of data. When the size of the training set is increased, we can see IC "catching up" as the amount of training data becomes more sufficient for constructing a good model for the joint distribution. It seems probable that this saturation effect is the cause for the indifference in the results in the first set of experiments, since in crossvalidation the amount of training data is usually quite high, e.g., in 10-fold

crossvalidation 90% of all the data available. This observation seems even more plausible as it is known that for many of the UCI data sets a rather small sample of the actual training data is enough for building a good predictive model (see the discussion in [4]).

5 Conclusion

We have studied the batch classification problem where multiple predictions can be made simultaneously, instead of performing the classifications independently one at a time. We demonstrated how the standard EM algorithm for finite mixtures can be modified for estimating both the missing latent variable data, and the classification data at the same time. In this unifying approach EM can be used both for model construction from training data and for making predictions. The empirical results with public domain classification datasets indicate that the batch approach may outperform the standard independent classification approach in cases with small sample sizes, where the extra information in the query batch can improve the model constructed.

Acknowledgements This research has been supported by the Technology Development Center (TEKES), and by the Academy of Finland. The primary tumor and the lymphography domains were obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklič for providing the data.

References

1. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
2. W. Emde. Inductive learning of characteristic concept descriptions from small sets of classified examples. In F. Bergadano and L. De Raedt, editors, *Proceedings of the 7th European Conference on Machine Learning (ECML94)*, pages 103–121, 1994.
3. B.S. Everitt and D.J. Hand. *Finite Mixture Distributions.* Chapman and Hall, London, 1981.
4. P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. Comparing predictive inference methods for discrete domains. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 311–318, Ft. Lauderdale, Florida, January 1997. Also: NeuroCOLT Technical Report NC-TR-97-004.
5. P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group 8556: Neural and Computational Learning (NeuroCOLT), 1996.
6. H. Tirri, P. Kontkanen, and P. Myllymäki. Probabilistic instance-based learning. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 507–515. Morgan Kaufmann Publishers, 1996.
7. D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, New York, 1985.