

Seneschal: classification and analysis in supervised mixture-modelling

Robert Munro
University of Sydney

Abstract. This paper describes an algorithm that is an extension of mixture-modelling to supervised clustering. It is demonstrated to be as accurate as current state-of-the-art machine learning algorithms across various data sets, and significantly more accurate than distance-based supervised clustering algorithms. Most significantly, it combines the classification itself with the calculation of rich information about the probabilities of class membership, the significance of attributes in relation to a classification, and the data space described by the data items and attributes.

1 Introduction

When carrying out a classification task involving machine learning, it is often desirable to have accurate information about the contribution of attributes to the classification. In some cases, a rich knowledge of the relationship between the attributes and the classification may be more important than small improvements in accuracy. For example, in natural language processing, the primary goal is often developing and testing various attributes to produce a language model, with accuracy simply used as one metric for determining the quality of that model.

Unsupervised clustering attempts to discover an optimal representation of a data set such that the data is divided into multiple clusters. The goal of such a task is typically the knowledge of that classification, given by the cluster definitions in relation to the distribution of items between clusters, and the distribution of attribute values between clusters.

Supervised learning attempts to discover an optimal representation of a data set with known class memberships. The goal of supervised learning is typically the building of a classifier for classifying unlabeled items.

Semi-supervised learning is a term applied to any combination of these, including a seeded unsupervised classification, and using unlabelled items in creating clusters. It can be defined as any supervised task that uses the attribute distributions, but not the labels, of test items. Semi-supervised learning is typically used when the ratio of training to testing items is very low, or when the attribute distributions in the training and test sets are known to be different.

Within clustering, mixture modelling is a methodology that assumes a given data set is the sum of simpler distributions. The discovery of the ‘intrinsic classification’ here is the discovery of an optimal representation as described by given statistical methods.

Mixture models [7] typically perform better than those based on apriori distance measures, such as a nearest neighbour algorithm, as they allow localised variation in the signifi-

cance of ‘distance’ according to that described by the data itself. See [1] for a more thorough discussion of the relationship between types of clustering techniques.

The algorithm described in this paper, *Seneschal*, was named after the position of a servant who acts as head steward at some function, knowing more about the appropriate ‘clustering’ of guests than the house head, but nonetheless remaining their servant.

This paper demonstrates that mixture models may be extended to supervised machine learning tasks; that with appropriate extensions and modifications such an algorithm is able to classify as accurately as many current state-of-the art algorithms, and over some data sets significantly more accurate than many existing clustering algorithms; and that it provides detailed information about the significance of attributes in relation to the classification, and may therefore be used as analysis tool.

Section 2 describes this paper’s relation to previous work. Section 3 discusses the information measures used here. Section 4 defines the algorithm used. Section 5 gives the results of testing and analysis, and section 6 concludes and discusses future directions.

1.1 Definitions

The use of terms such as *cluster* and *category* vary. Within this paper, the following terminology is defined:

classes: the target categories of a supervised task.

cluster: any set of items taken from one data set, typically associated through some measure of nearness.

subcluster: any cluster that contains items of only one class.

child cluster: a cluster whose set of items are a subset of one subcluster.

labeled/unlabeled data: data items with/without known class memberships.

training/test sets: pre-labeled/unlabeled item sets in supervised learning.

2 Related work

The work relating most to that here is [10]. There, the Bayesian unsupervised classifier *AutoClass* was extended to a supervised learner *MultiClass*. Like *AutoClass* it considers all features in defining clusters, but differs in that it only evaluates the conditional probability of the feature representing the target classes.

More recently, in [13] the unsupervised clusterer *COBWEB* was extended to the supervised *COP-KMEANS*, a constraint-based semi-supervised and supervised classifier. They report good results for semi-supervised tasks, but relatively poor results for a supervised classification.

Although not as strongly related to the methods discussed here, unsupervised mixture modelling is much more developed, as displayed with the unsupervised mixture modeller *SNOB* [14, 15].

Within Bayesian related supervised techniques, the hybrid algorithm *NB-Tree* [5], combining decision trees with a Naive Bayes classifier, is one of the most successful recent developments, and one to which results here are compared.

3 Information measures

There are many different heuristics that may be used in mixture models, such as the EM algorithm, Bayesian measures, Minimum Description Length (MDL) and Minimum Message Length (MML). These techniques all seek an optimal model for a data set by utilising an entropy or information measure (IM).

Here, the heuristics used for creating a model of the data, performing classifications and determining significance testing are derived from existing Minimum Message Length (MML) research, but any one or a combination of the above heuristics could have been used. The motivation for choosing a measure derived from MML over other metrics was based on its superior performance as reported in [12], and its previous successful application to supervised tasks.

Reproducing the derivation of these equations is omitted here due to space constraints (see [3] and [15]), although they are largely intuitive. Here, the equations are given and discussed only in terms of how they differ from these. Because of the differences, and as other heuristics may have been used, the more generic term ‘information measure’ is used here.

3.1 Multistate attributes

These are also known as discrete, multinomial and/or categorical attributes.

Given an item i with value i_α for multistate attribute α , and given that i_α occurs in cluster C with frequency $f(i_\alpha, C)$, within the data set T , i 's information measure for n multistate attributes for C with size $s(C)$ is given by:

$$IM(i, C) = \sum_{\alpha=1}^n -\ln \frac{f(i_\alpha, C) + 1}{s(C) + \gamma} \quad (1)$$

Where γ is given by the constant:

$$\gamma = 1 - \frac{f(i_\alpha, T)}{f(i_\alpha, T) - s(T)} \quad (2)$$

This differs from that described by MML, in that it does not assume that all attribute values are equally likely to occur. Here, it is assumed that the relative probability of attribute values are the apriori relative frequencies of the entire set given by γ . In [3] and [15], it is assumed that all values are equally likely, hence γ is simply the arity of α . This was modified to exploit the supervised learning situation where there exists a correlation between a low frequency attribute value and a low frequency class.

3.2 Continuous attributes

In the current implementation, all continuous attributes are treated as Gaussian, assuming a normal distribution. Put simply, the IM of an item i in a cluster C , for a given attribute β correlates to the i 's value for β in relation to the mean and standard deviation of β in C .

Given an item i with value i_β for continuous attribute β , i 's information measure for n continuous attributes for a cluster C that for attribute β that has an average of μ_β and standard deviation of σ_β is given by:

$$IM(i, C) = \sum_{\beta=1}^n \frac{(i_{\beta} - \mu_{\beta})^2}{2\sigma_{\beta}^2} \quad (3)$$

3.3 Cost of cluster membership

Many clustering algorithms employ some additional penalty correlating to the number and/or size of clusters. In supervised learning, a fine-grained distribution is often desirable. Here, the cost of cluster membership has been replaced by the inclusion of a user-defined prevalence threshold, indicating the minimum size a cluster may be without being considered noise. Given that it is often desirable that a supervised classifier be insensitive to differences in the class frequency distributions between training and test items, this blindness to these frequencies beyond the threshold is the solution employed here.

In an early implementation, a penalty with a weighting factor was introduced, but removed from the algorithm once it was found that even small weights resulted in lower accuracy [9].

An alternative not tested here, would be to use a penalty only in the building of the classifier, but omit that part of the measure in the classification process. Yet another alternative would be to use a measure sensitive only to the frequency of a given class, rather than the whole data set.

4 Algorithm Description

The steps here are: building the classifier; classifying test items and classification analysis.

4.1 Building the classifier

After a rough initial division into subclusters, the algorithm iteratively attempts to reduce the overall IM by reassigning single items between subclusters, and combining or splitting whole subclusters. Like the EM algorithm, these actions are only taken when they result in an improved IM. The model is considered built once no improvements are possible. This proceeds as follows:

Initialise: the data space is initially split into a multidimensional ‘grid’, with divisions existing at the average value of each continuous attributes, and at plus and minus one standard deviation. For multistate items, they are simply divided according to attribute value. Items are initially assigned to a subcluster corresponding to the grid described by that item’s attribute values. This gives a very rough division of the data into subclusters, its advantage being that, once the averages and standard deviations are known, it only requires a single pass over the data.

Reassign: for each item, calculate if it possesses a lower IM for a subcluster other than the one it is currently a member of. If so, reassign the item to the subcluster for which it has the lowest IM.

Combine: for each pair of subclusters, combine them if it results in a lower overall IM for the items in both clusters.

Split: when an item is assigned to a subcluster (or reassigned to the same subcluster), it is also assigned to one of two child clusters of that subcluster. The choice of which to assign it to is also based on the IM of assigning the item to each child cluster. If the overall IM of the two child clusters is less than that of the subcluster, then the subcluster is split into two subclusters corresponding to the two child clusters. While the method here does not guarantee an optimal distribution between child clusters, it can be seen as a good approach to the extent that using an IM is better than a random distribution, and it will also scale better than iteratively attempting random redistributions.

In all the above steps, a constraint maintained throughout is that no training items are allowed in a cluster containing items of another class. Many semi-supervised models build unsupervised models over the entire data set, and then assign a class to each cluster according to the most frequent class of the items in that cluster. This has not been implemented here, as once these clusters have been assigned a class it must be assumed that all items in that cluster known to not belong to that class are either noise or misclassifications. If this is the case, than those items should not have contributed to defining or creating that cluster. In effect, it is training to noise, and therefore prone to overfitting. For a further discussion of the problems inherent in a semi-supervised approach to mixture modelling see [4].

4.2 *Classifying the test items*

Although this algorithm will allow semi-supervised and seeded classifications, all testing described in this paper is for a supervised classification.

Here, after the classifier has been built, each test item is assigned the class of the subcluster for which it has the lowest IM. ‘Ties’ are treated as an incorrect classification.

As items are given a set of probabilistic distributions across the various clusters, the output of a classification can be used informatively as part of an ensemble learner or manipulated to influence the precision and recall values of a given class, but a full discussion of the advantages of a probabilistic classification over a discrete one is outside the scope of this paper.

4.3 *Classification analysis*

Part of the robustness of this algorithm derives from the fact it uses the same metrics for clustering, classification and analysis. The most obvious analysis to perform is to explore the item membership and frequency of the subclusters of each class, but investigating the relationships between subclusters and an attribute analysis may also be desirable.

The closeness of a cluster C_1 to a second cluster C_2 is given by the average IM of assigning the items of C_1 to C_2 .

This value will not necessarily be the same in reverse. If C_1 is a relatively tight cluster, perhaps even embedded within the range of C_2 for all attributes, then the average cost of assigning the items from C_2 to C_1 will be larger, as the average number of standard deviations from an item in C_2 to C_1 will itself be greater. If C_1 and C_2 are subclusters of different classes, then this measure will be the amount by which the clusters define the class membership of their items, with respect to the portion of the data space described by these clusters.

Rather than investigating the entire IM of a cluster, comparing the IM of one attribute to others can show which of the attributes is most significantly distributed within that cluster.

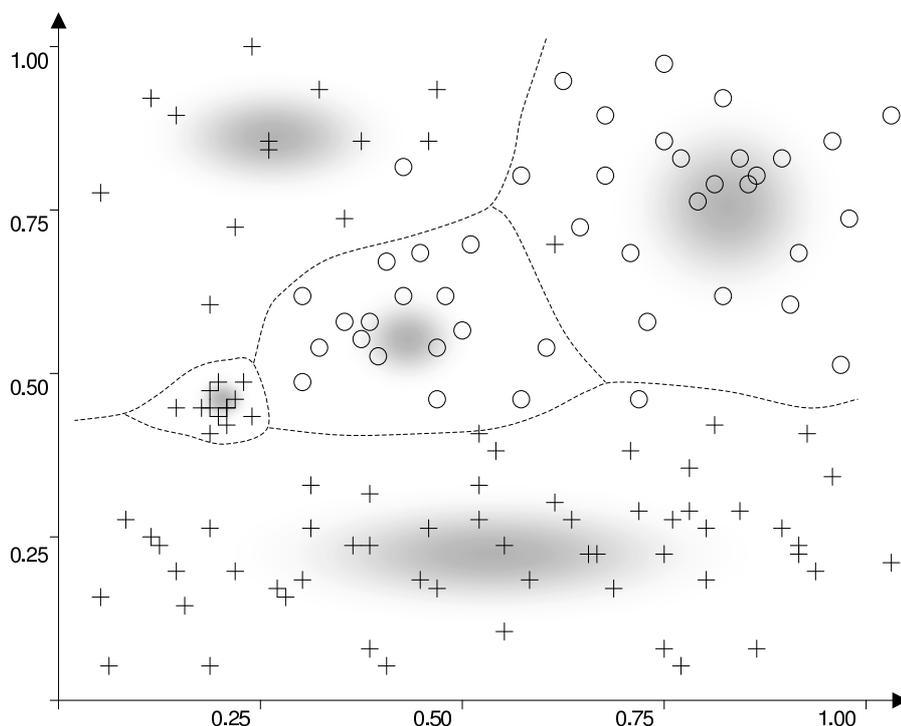


Figure 1: An example of a mixture model of two classes with five subclusters in two dimensions

An alternative that is probably more appropriate for supervised learning is to compare the relationships between subclusters based on one only attribute, or on a subset of them.

Significance at the subcluster level of the data is much richer than that at level of the full set. For example, given a two-class ‘chess-board’ of items, there is no difference in the distribution of the two classes across the full set, that is, both the black squares and white squares are uniformly distributed across both axes. Between adjacent squares, however, the localised significance is absolute. A less severe example of this can be seen in Figure 1. For $y < 0.4$, the distribution along the x axis is not significant in terms of the classification, but is of greater and varying degrees of significance for all $y > 0.4$. With this information, it is easy to see how such a model may be used as tool for feature selection. The global significance of a feature may be calculated as the weighted aggregate of its subcluster significances, or simply the maximum of its local significances.

Figure 1 also provides a good example of how the effective boundary of a cluster in a mixture model will differ from that defined by a clustering algorithm employing an apriori distance measure. At $(0.25, 0.5)$ an effective boundary exists between a relatively dense cluster and three relatively sparse clusters. This boundary ‘hugs’ the boundary of the dense cluster as, despite the large number of items, the standard deviations are relatively lower.

Table 1: Accuracy comparison

Data set	Max SVM	Naive Bayes	NB-Tree	Seneschal
Adult	0.85	0.84	0.86	0.83
LED24	0.67	0.73	0.73	0.74
Mushrooms	1.00	1.00	1.00	1.00
Tic-tac-toe	0.95	0.71	0.70	1.00
Letter	(0.89)	0.73	0.88	0.84

5 Implementation Testing

Testing seeks to compare the accuracy of this algorithm to other supervised machine learning algorithms, its use in parameter selection, and its use as an analytical tool.

While optimising accuracy is not the main goal of this paper, it is necessary that the algorithm perform an accurate classification for the information contained in that classification to be an accurate representation of the data.

Here, a ten-fold cross-validation was performed on the Adult, Mushrooms, Tic-tac-toe, and Letter data sets. The results discussed and reported in Tables 1 and 2 are from just one of these (arbitrarily the first), as an analysis of attribute significance across multiple classifiers is itself a complicated task not undertaken here. The other nine results showed no change in accuracy for the Tic-tac-toe or Mushrooms sets, a marginal increase in the average result for Adult, and a marginal decrease in the average result for Letter. The LED24 set, a synthetically generated set, was generated using separate seeds for the training and test sets so that the two distributions were identically distributed but not simply identical.

Although the other results in Table 1 were most likely obtained from different splits of the data to those used here, it is reasonable to assume that they would not differ too significantly. The alternative, implementing all these algorithms for the sake of comparison, is outside the scope of this research, but it's still more desirable to present comparisons with some of the highest reported results at the expense of precision in comparison. The result in brackets indicates that the set was reduced to a two class problem. The data sets, taken from the UCI Machine Learning Repository [8], were primarily selected because many results existed for the other algorithms against which *Seneschal* may be compared. Within these possible selections, the sets here were chosen because they are among the larger UCI sets, and for the following reasons:

Adult: a variety of distributions of multistate and continuous attributes.

LED24: 7 'meaningful' attributes containing 10% noise, 17 attributes are pure noise.

Mushrooms: contains missing values.

Tic-tac-toe: a data set that many supervised clustering algorithms perform very poorly on.

Letter: attribute type selection (see section 5.1)

The results for NB-Tree and Naive Bayes are those reported in [5].

Support Vector Machines represent a large part of recent developments in machine learning algorithms, but state-of-the art SVM's often perform much better on two class problems and typically don't explicitly support multistate attributes. The SVM results in Table 1 are, to

Table 2: Seneschal classification details

Data set	Test acc.	Training acc.	IGR	num classes	num subclusters
Adult	0.83	0.83	0.5539	2	8
LED24	0.74	0.74	0.0009	10	10
Mushrooms	1.00	1.00	0.0007	2	81
Tic-tac-toe	1.00	1.00	0.0031	2	23
Letter	0.84	0.87	0.0013	26	246

the authors best knowledge, the highest reported results from SVM’s on these data sets, but the search was not exhaustive.

The SVM results for the Adult, Tic-tac-toe and Letter sets are the maximum reported values from a comparison of a number of SVM’s in [6]. There, multistate attributes were translated into multiple ‘binary’ continuous attributes. For the Letter data set they reduced the set to the two classes representing the letters ‘A’ and ‘B’. When classifying only these two classes *Seneschal* obtained 100% accuracy.

The LED24 SVM results are from [2] (they report an accuracy of 73% once feature selection has removed the 17 pure noise attributes). The Mushrooms SVM results are reported in [11].

For the Tic-tac-toe set, [10] reports 98% accuracy with *MultiClass*. With *COP-KMEANS*, [13] reports 83% accuracy on the Mushrooms data set, but only 55% on the Tic-tac-toe test set. See section 5.2 for a discussion of this.

5.1 Attribute type selection

The attributes in the Letter data set are integers, all with 16 different values representing various measurements/ratios of a given letter. Here, two tests were completed: one treating all attributes as continuous, the other as multistate.

The decision about which to use was based on the information gain ratio (IGR), given by the aggregate IM of all subclusters divided by the IM of the whole set treated as one cluster. Treated as multistate attributes, this was 0.0013, as given in Table 2, but for continuous it was 0.9319. On the basis of this the former was chosen as the more appropriate representation. The accuracy obtained in treating these attributes as continuous was 0.63, which confirms that using a multistate representation was more appropriate for this data set.

The results for the Letter set are still the least impressive of those reported here, not so much for accuracy but because they are the only ones that overfit the data, as displayed by the differences in test set and training set accuracies in Table 2. It is possible that the model reached a local minimum, although, as it overfit, only the large number of subclusters evidences this within the training set.

Perhaps treating some attributes as continuous and some as multistate would produce better results, but it’s as likely that the nature of the data set was such that the information measures described here are unable to exploit the attribute distributions for a more accurate classification, as the highly normalised attribute values may have removed some of the ‘clusteredness’ that *Seneschal* takes advantage of in building the classifier, and resulted in a distribution that is probably best described as ordinal, rather than Gaussian or multistate.

5.2 Data analysis

The Tic-tac-toe data set provides a good model for analysis as the relative significance of the attributes, each representing a square on the board, should be well-known to anyone who's played the game.

As expected, the 'middle square' attribute was reported as most significant, given by the lowest aggregate IM over all subclusters. Interestingly, within single subclusters, it was reported as either very significant or not significant. Intuitively, it is likely that this is due to the fact that the middle square is irrelevant if a winning row exists along one edge.

In [13] it was speculated that the poor performance of *COP-KMEANS* on this data set was due to the need to capture correlation between attributes, but as *Seneschal* also fails to explicitly capture attribute correlation, it would seem to be enough to discover the local significance, and to perform subclustering and classifications based on it, to obtain high accuracy for such data.

6 Conclusions

It has been demonstrated that extending mixture-modelling to supervised learning is possible such that it performs as accurately as many state-of-the-art algorithms over various data sets, and significantly more accurate than distance-based supervised clustering algorithms, while retaining the desirable properties of clustering algorithms and mixture models, such as robustness across missing values, noise tolerance and the provision of rich information about the items and attributes in relation to a classification.

6.1 Ongoing / Future work

Work on this algorithm and its application is ongoing:

1. Optimising the efficiency of the algorithm is ongoing, which was why a cost analysis was not included here.
2. The motivation for developing this algorithm came from the need for an algorithm to provide attribute information about probabilistic classifications in computational linguistics, and so future work will be directed towards this area, but testing over other domains, especially spatial data, would be interesting.
3. Although the search for an optimal mixture model is NP-hard, and decreasing IM is not necessarily monotonic with respect to increasing accuracy, any technique that consistently produced a more optimal model is desirable.
4. Here, all continuous attributes were treated as Gaussian. Within MML there also exist definitions for circular and Poisson attributes [15], the addition of which here would be inherently beneficial. While the extension to such a model would be simple, it might also be interesting to see how the attribute-type selection methods described in section 5.1 could be automated to provide an optimised selection within continuous attributes.
5. The information reported would be richer if it explicitly described correlations between attributes. This is probably the biggest shortcoming of the algorithm as an analytical tool.

References

- [1] P. Berkhin. Survey of Clustering Data Mining Techniques, Accrue Software, [http://www.accrue.com/products/rp_cluster_review.pdf] (2002)
- [2] P. Bhattacharyya, V. Sindhwani and S. Rakshit. Information Theoretic Feature Crediting in Multiclass Support Vector Machines, *Proceedings of the first SIAM International Conference on Data Mining*, (2001)
- [3] D.M. Boulton. The Information Criterion for Intrinsic Classification. *PhD thesis*, Monash University (1975)
- [4] F.G. Cozman, I. Cohen and M.C. Cirelo, Semi-Supervised Learning of Mixture Models and Bayesian Networks, *Proceedings of the Twentieth International Conference of Machine Learning*, (2003)
- [5] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (1996)
- [6] O.L. Mangasarian and D.R. Musicant Lagrangian Support Vector Machines, *Journal of Machine Learning Research*, vol 1, (2001), 161–177
- [7] G.J. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley (2000)
- [8] C.J. Merz and P.M. Murphy. UCI Repository of machine learning databases [<http://www.ics.uci.edu/mllearn/MLRepository.html>], Irvine, CA: University of California, (1996)
- [9] R. Munro. Classification and analysis in supervised mixture-modelling. *School of Information Technologies Technical Report no. 539*, University of Sydney (2003)
- [10] H.S. Roy. Sharp, Reliable, Predictions Using Supervised Mixture Models. *PhD thesis*, Stanford University (1995)
- [11] S. Rüping. Incremental Learning with Support Vector Machines, [http://www-ai.cs.uni-dortmund.de/DOKUMENTE/rueping_2002c.pdf] (2002)
- [12] M.A. Upal and E. Neufeld. Comparison of unsupervised classifiers. *Proceedings of the first International Conference on Information, Statistics and Induction in Science*, (1996), 342–353.
- [13] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl. Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, (2001) 577–584.
- [14] C.S. Wallace and D.M. Boulton. An Information Measure for Classification. *Computer Journal*, vol 11.2, (1968), 185–194.
- [15] C.S. Wallace and D.L. Dowe. MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions. *Statistics and Computing*, vol 10.1, (2000), 73–83.