Using Supervised Clustering to Enhance Classifiers

Christoph F. Eick and Nidal Zeidat

Department of Computer Science, University of Houston,
Houston, TX 77204-3010, USA
{ceick, nzeidat}@cs.uh.edu

Abstract. This paper centers on a novel data mining technique we term *supervised clustering*. Unlike traditional clustering, supervised clustering is applied to classified examples and has the goal of identifying class-uniform clusters that have a high probability density. This paper focuses on how data mining techniques in general, and classification techniques in particular, can benefit from knowledge obtained through supervised clustering. We discuss how better nearest neighbor classifiers can be constructed with the knowledge generated by supervised clustering, and provide experimental evidence that they are more efficient and more accurate than a traditional 1-nearest-neighbor classifier. Finally, we demonstrate how supervised clustering can be used to enhance simple classifiers.

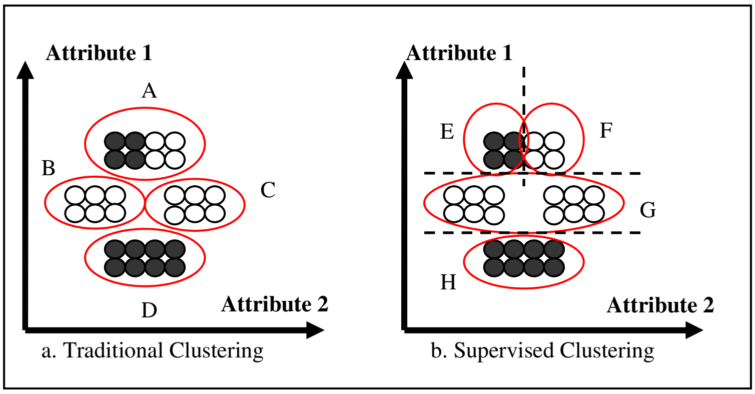
1 Introduction

This paper centers on a novel data mining technique we term *supervised clustering*. Clustering is, typically, applied to a set of unclassified examples using particular error functions, e.g. an error function that minimizes the distances inside a cluster keeping clusters tight. *Supervised clustering*, on the other hand, deviates from traditional clustering in that it is applied on classified examples with the objective of identifying clusters that have high probability density with respect to a single class. Moreover, in supervised clustering, we also like to keep the number of clusters small, and objects are assigned to clusters using a notion of closeness with respect to a given distance function.

Fig. 1, that depicts examples belonging to two classes, illustrates the differences between traditional and supervised clustering. A traditional clustering algorithm would, very likely, identify the four clusters depicted in Figure 1.a. A supervised clustering algorithm that maximizes class purity, on the other hand (see Fig. 1.b), would split cluster **A** into two clusters **E** and **F**. Another characteristic of supervised clustering is that it tries to keep the number of clusters low. Consequently, clusters **B** and **C** would be merged into a single cluster **G** without compromising class purity while reducing the number of clusters.

Our previous work [4, 13] centered on the design and implementation of algorithms for supervised clustering and on comparative studies that evaluate different supervised clustering algorithms with respect to quality of solutions found and runtime. In this paper, we will discuss how local and regional learning techniques can benefit from background knowledge that has been generated through supervised clustering.

M.-S. Hacid et al. (Eds.): ISMIS 2005, LNAI 3488, pp. 248 – 256, 2005.



a. Traditional Clustering

b. Supervised Clustering

Fig. 1. Differences between traditional clustering and supervised clustering

Section 2 discusses related work. Section 3 introduces supervised clustering in more details. Section 4 presents experimental results that show the benefits of supervised clustering. Section 5 concludes the paper.

2 Related Work

There are two approaches can be viewed as supervised clustering approaches. Sinkkonen et al. propose a very general approach called *discriminative clustering* [7] that minimizes distortion within clusters. Distortion, in their context, represents the loss of mutual information between the auxiliary data (e.g., classes) and the clusters caused by representing each cluster by a prototype. The technique seeks to produce clusters that are internally as homogeneous as possible in conditional distributions $p(c|x)$ of the auxiliary variable, c (i.e., belong to a single class), with respect to a clustering x. Similarly, Tishby et. al. introduce the *information bottleneck method* [9]. Based on that method, they propose an agglomerative clustering algorithm [8] that minimizes information loss with respect to p(c|a) with c being a class and a being an attribute.

However, there has been some work that has some similarity to our research under the heading of semi-supervised clustering. The idea of semi-supervised clustering is to enhance a clustering algorithm by using side information in the clustering process that usually consists of a "small set" of classified examples; the objective of the clustering process then is to optimize class purity (examples with different class labels should belong to different clusters) in addition to the traditional objectives of a clustering algorithm. Demiriz [3] proposes an evolutionary clustering algorithm in which solutions consist of k centroids and the objective of the search process is to obtain clusters that minimize (the sum of) cluster dispersion and cluster impurity. Basu et. al. [1] centers on modifying the K-means clustering algorithm to cope with prior knowledge. Xing [12] (and similarly [2]) takes the classified training examples and transforms those into constraints (points that are known to belong to different classes need to have a distance larger than a given threshold) and derives a modified distance function that minimizes the distance between points in the dataset that are known to be similar with respect to these constraints using classical numerical methods. The K-means clustering algorithm in conjunction with the modified distance function is then used to compute clusters.

3 Supervised Clustering

As mentioned earlier, the fitness functions used for *supervised* clustering are significantly different from the fitness functions used by *traditional* clustering algorithms. Supervised clustering evaluates a clustering based on the following two criteria:

- *Class impurity, Impurity(X).* Measured by the percentage of minority examples in the different clusters of a clustering X.
- *Number of clusters, k.* In general, we like to keep the number of clusters low; trivially, having clusters that only contain a single example is not desirable, although it maximizes class purity.

3.1 A Fitness Function for Supervised Clustering

In particular, we used the following fitness function in our experimental work (lower values for q(X) indicate 'better' clustering solution X).

$$q(X) = \text{Impurity}(X) + \beta * \text{Penalty}(k) \tag{1}$$

$$\text{where Impurity}(X) = \frac{\text{\# of Minority Examples}}{n}, \text{ and Penalty}(k) = \begin{cases} \sqrt{\dfrac{k-c}{n}} & k \geq c \\ 0 & k < c \end{cases}$$

with n being the total number of examples and c being the number of classes in a dataset. The parameter β ($0< \beta \leq 2.0$) determines the penalty that is associated with the number of clusters, k, in a clustering: higher values for β imply larger penalties for a higher number of clusters. The objective of Penalty(k) is to dampen the algorithm's natural tendency to increase the number of clusters. However, this dampening ought not to be linear because the effect of increasing the number of clusters from k to k+1 has much stronger effect on the clustering outcome when k is low than when k is high. Consequently, we selected a non-linear function for Penalty(k) that has higher slope when k is low. Finding the best, or even a good, clustering X with respect to the fitness function q is a challenging task for a supervised clustering algorithm [13].

3.2 Representative-Based Supervised Clustering Algorithms

There are many possible algorithms for supervised clustering. Our work centers on the development of *representative-based* supervised clustering algorithms. Representative-based clustering aims at finding a set of representative objects in a dataset O, and creates clusters by assigning objects to the cluster of their closest representative. Representative-based supervised clustering algorithms seek to accomplish the following goal: *Find a subset O_R of O such that the clustering X obtained by using the objects in O_R as representatives minimizes q(X).*

As part of our research, several representative-based clustering algorithms have been proposed [4,13], 3 of which are briefly described below:

1. SPAM is a variation of the popular traditional clustering algorithm PAM [5] that uses q(X) as its fitness function.
2. SRIDHCR starts with a randomly chosen initial set of representatives. The algorithm, then, greedily tries to improve this solution by inserting or deleting single representatives. Moreover, the algorithm is restarted *r* times.
3. SCEC uses evolutionary computing and evolves a population of solutions over a fixed number of generations based on the principle of the survival of the fittest. The genetic composition of solutions is changed by applying mutation and crossover operators.

3.3 Relationship to Local and Regional Learning

One way to characterize inductive learning techniques is to analyze if and to which extent they support the notion of locality. A *k*-nearest neighbor classifier is an example of a local learning technique (assuming it uses a low k-value); only objects that are very close to the object to be classified are used to predict the class label of that object. Other techniques subdivide the search space into different regions and use regional knowledge to fit the best model to each region. A good example for such a regional technique is a regression tree. In contrast to local techniques, a much larger number of examples are used to predict the class of an unseen example. Finally, global techniques try to fit a single model to the complete dataset. A good example for a global technique is classical regression analysis that tries to find a single curve that minimizes a prediction error with respect to all the examples that belong to the dataset.

As we will discuss in Section 4 of this paper, the parameter β plays key role in determining whether the patterns identified by supervised clustering are local (i.e., high value for β) or are regional (i.e., lower values for β).

4 Benefits of Supervised Clustering

4.1 Supervised Clustering for Creating Summaries and Background Knowledge

Figure 2 shows how cluster purity and the number of clusters *k* for the best solution found, changes as the value of parameter β increases for the Vehicle and the Diabetes datasets (the results were obtained by running algorithm SRIDHCR). As can be seen in Figure 2, as β increases, more penalty is associated with using the same number of clusters and the algorithm tries to use a lower number of clusters resulting in decreasing cluster purity. It is interesting to note that the Vehicle dataset seems to contain smaller regions with above average purities. Consequently, even if β increases beyond 0.5 the value of *k* remains quite high for that dataset. The Diabetes dataset, on the other hand, does not seem to contain such localized patterns; as soon as β increases beyond 0.5, *k* immediately reaches its minimum value of 2 (there are only two classes in the Diabetes dataset).

In general, we claim that supervised clustering is useful for creating background knowledge with respect to a given dataset. Examples include:

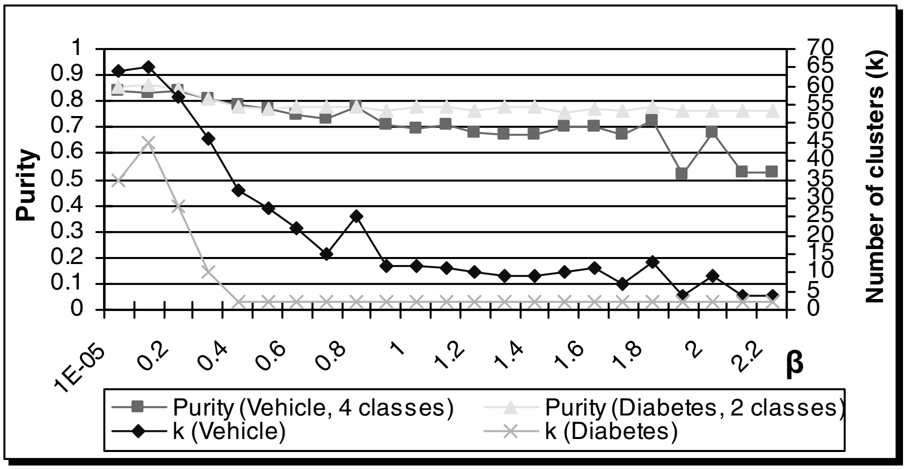


Fig. 2. How Purity(*k*) and *k* change as the value of increases

1. It shows how instances of a particular class distribute in the attribute space; this information is of value for "discovering" subclasses of particular classes.
2. Maps for domain experts can be created that depict class densities in clusters and that identify which clusters share decision boundaries with each other.
3. Statistical summaries can be created for each cluster.
4. Meta attributes, such as various radiuses, distances between representatives, etc. can be generated, and their usefulness for enhancing classifiers can be explored.

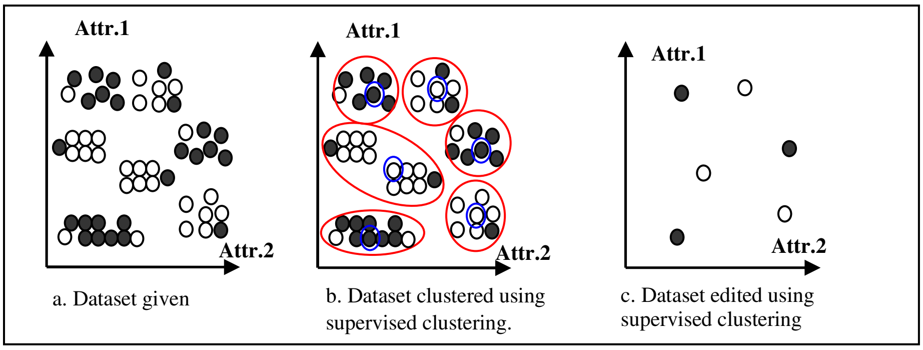
4.2 Using Cluster Prototypes for Dataset Editing to Enhance NN-Classifiers

The objective of *dataset editing* [11] is to remove examples from a training set in order to enhance the accuracy of a classifier. In this paper, we propose using supervised clustering for *editing* a dataset O to produce a reduced subset O_r consisting of cluster representatives that have been selected by a supervised clustering algorithm. A 1-Nearest-Neighbor (1-NN) classifier, that we call nearest-representative (NR) classifier, is then used for classifying new examples using subset O_r instead of the original dataset O. We call this approach *supervised clustering editing* (SCE for short).

Figure 3 gives an example that illustrates how supervised clustering is used for dataset editing. Figure 3.a shows a dataset that has not been clustered yet. Figure 3.b shows the same dataset partitioned into 6 clusters using a supervised clustering algorithm. Cluster representatives are marked with circles around them. Figure 3.c shows the resulting subset O_r after the application of supervised clustering editing.

In general, an editing technique reduces the size of a dataset, n, to a smaller size k; we define the dataset compression rate of an editing technique as follows:

$$Compression\ Rate = 1 - \frac{k}{n} \tag{3}$$



a. Dataset given

b. Dataset clustered using supervised clustering.

c. Dataset edited using supervised clustering

Fig. 3. Editing a dataset using supervised clustering

We applied our editing technique on a benchmark of 8 datasets obtained from [6]. Since β directly affects the size of reduced set O_r (larger β values produce smaller O_r sets while smaller β values tend to produce larger O_r sets) and in order to explore different compression rates, two different values for parameter β were used: 1.0 and 0.4. Prediction accuracies were measured using 10-fold cross-validation. Representatives for the nearest representative (NR) classifier were computed using the SRIDHCR algorithm. In our experiments, SRIDHCR was restarted 50 times, and the best solution (i.e., set of representatives) found in the 50 runs was used as the edited subset for the NR classifier. We also computed prediction accuracy for a traditional 1-NN classifier that uses all training examples when classifying a new example. Table 4 reports the accuracies obtained using the edited subset and the original dataset as well as the average dataset compression rates for supervised clustering editing. Due to the fact that the supervised clustering algorithm has to be run 10 times, once for each fold, Table 4 also reports the average, minimum, and maximum number of representatives found in the 10 runs.

Inspecting the results in Table 4, we can see that the SCE approach accomplished significant improvement in accuracy for the Heart-Stat Log, Diabetes, Waveform, and Iris-Plants datasets, outperforming the traditional 1-NN-classifier. Further inspecting the second and third columns of Table 4, we notice that with the exception of the Glass and the Segmentation datasets, SCE accomplishes compression rates of more than 94% without a significant loss in prediction accuracy for the other 6 datasets.

The reader might ask why it is necessary to develop supervised clustering algorithms for the purpose of editing. Couldn't the same objective be accomplished by clustering examples of each class separately using a traditional clustering algorithm, such as PAM [5]? Figure 4, that shows examples of a dataset consisting of 2 classes 'X' and 'O', illustrates why this is not a good idea. If this dataset is edited using supervised clustering, the red (underlined) **<u>O</u>** example and the purple (underlined) **<u>X</u>** example would be picked as representatives. On the other hand, if examples of each class are clustered separately, the blue (italic) ***O*** example and the purple (underlined) **<u>X</u>** example would be picked as the representatives. Note that the blue (italic) ***O*** representative is **not** a good choice for dataset editing, because it "attracts" examples belonging to the class 'X' which leads to misclassifications.

Table 4. Dataset compression rates for SCE and prediction accuracy for the NR and 1-NN

B	Avg. *k* [Min-Max] for SCE	SCE Compression Rate (%)	NR Prediction Accuracy	1-NN Prediction Accuracy
Glass **(214)**				
0.4	25 [19-29]	88.4	0.589	0.692
1.0	6 [6 – 6]	97.2	0.575	0.692
Heart-Stat Log **(270)**				
0.4	2 [2 – 2]	99.3	0.833	0.767
1.0	2 [2 – 2]	99.3	0.838	0.767
Diabetes **(768)**				
0.4	9 [2-18]	98.8	0.736	0.690
1.0	2 [2 – 2]	99.7	0.745	0.690
Vehicle **(846)**				
0.4	38 [26-61]	95.5	0.667	0.700
1.0	14 [9-22]	98.3	0.665	0.700
Heart-H **(294)**				
0.4	2	99.3	0.793	0.783
1.0	2	99.3	0.809	0.783
Waveform **(5000)**				
0.4	28 [20-39]	99.4	0.841	0.768
1.0	4 [3-6]	99.9	0.837	0.768
Iris-Plants **(150)**				
0.4	3 [3 – 3]	98.0	0.973	0.947
1.0	3 [3 – 3]	98.0	0.953	0.947
Segmentation (2100)				
0.4	30 [24-37]	98.6	0.919	0.956
1.0	14	99.3	0.889	0.956

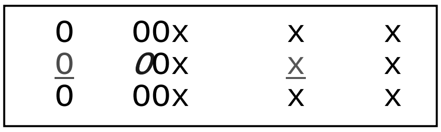


Fig. 4. Supervised clustering editing vs. clustering each class separately

4.3 Using Supervised Clustering to Enhance Simple Classifiers

Another capability of supervised clustering is that it can be used to enhance classifiers by using regional knowledge. Referring to Figure 1 again, we could transform the problem of classifying examples belonging to the two classes “black circles” and “white circles” into the “simpler” problem of classifying the examples that belong to clusters E, F, G, and H. The reduced complexity can be attributed to the fact that those 4 “clusters” are linearly separable (note the dotted lines in Fig. 1.b) whereas the original 2 classes are not. Vilalta et. al. [10] proposed a methodology that first clusters the examples of each class separately using a traditional clustering algorithm, and then learns a classifier by treating the so obtained clusters as separate classes.

We propose to use supervised clustering for class decomposition instead of clustering each class separately using traditional clustering algorithm because supervised clustering has the tendency to merge clusters of the same class if found close to each others, such as cluster G in Figure 1.b. To test this idea, we conducted an experiment where we compared the prediction accuracy of a traditional Naïve Bayes classifier with a Naïve Bayes classifier that treats each cluster as a separate class. We used 4 UCI datasets as a benchmark. We used SRIDHCR supervised clustering algorithm (with β set to 0.25) to obtain the clusters. The results reported in Table 5 indicate that using class decomposition improved the prediction accuracy for 3 of the 4 datasets tested. Furthermore, analyzing the results further, we see that the accuracy improvement for the Vehicle dataset (23.23%) is far higher than that for the Diabetes dataset (0.52%). This result is consistent with the analysis presented in section 4.1.

Table 5. Prediction accuracy of "Naive Bayes" and "Naive Bayes with class decomposition"

Dataset	**Naïve Bayes (NB)**	**NB with Class Decomposition**	**Improvement**
Diabetes	76.56	77.08	0.52%
Heart-H	79.73	70.27	–9.46%
Segment	68.00	75.045	7.05%
Vehicle	45.02	68.25	23.23%

5 Conclusion

In this paper a novel data mining technique we named *supervised clustering* was introduced that, unlike traditional clustering, assumes that the clustering algorithm is applied to classified examples and has the objective of identifying clusters that have a high probability density with respect to a single class. We discussed how local and regional knowledge that has been generated by supervised clustering can be used to enhance classifiers. We also demonstrated how regional knowledge generated by supervised clustering can be used for enhancing simple classifiers. We also believe that running a supervised clustering algorithm gives valuable information about how the examples of the dataset distribute over the attribute space.

In addition to developing efficient and scalable supervised clustering algorithms, our future work centers on using supervised clustering for dataset compression, for learning subclasses, and for distance function learning.

References

1. Basu, S., Bilenko,M., Mooney, R.: Semi-supervised Clustering by Seeding. In Proceedings of the Nineteenth International Conference on Machine Learning (ICML'02). Sydney, Australia, July 2002. 19-26,
2. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning Distance Functions Using Equivalence Relations. In Proc. ICML'03, Washington DC, August 2003.
3. Demiriz, A., Benett, K.-P., Embrechts, M.J.: Semi-supervised Clustering using Genetic Algorithms. In Proc. ANNIE'99.
4. Eick, C., Zeidat, N., Zhao, Z..: Supervised Clustering – Algorithms and Benefits. Proc. ICTAI'04, Boca Raton, FL, November 2004.

5. Kaufman, L., Rousseeuw P. J.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
6. University of California at Irving, Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html
7. Sinkkonen, J., Kaski, S., and Nikkila, J.: Discriminative Clustering: Optimal Contingency Tables by Learning Metrics. In Proc. ECML'02. Helsinki, Finland, August 2002.
8. Slonim, N. and Tishby, N.: Agglomerative Information Bottleneck. Neural Information Processing Systems (NIPS'99).
9. Tishby, N., Periera, F.C., and Bialek, W.: The Information Bottleneck Method. In proceedings of the 37th Allerton Conference on Communication and Computation, 1999.
10. Vilalta, R., Achari, M., and Eick C.: Class Decomposition Via Clustering: A New Framework For Low-Variance Classifiers. In Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, FL, November 2003.
11. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics, 2:408-420, 1972.
12. Xing, E.P., Ng, A., Jordan, M., Russell, S.: Distance Metric Learning with Applications to Clustering with Side Information. Advances in Neural Information Processing 15, MIT Press, 2003.
13. Zeidat, N., Eick, C.: Using k-medoid Style Algorithms for Supervised Summary Generation. Proc. MLMTA'04, Las Vegas, June 2004.