Report Series

Competitive Learning for Binary Valued Data

Friedrich Leisch
Andreas Weingessel
Evgenia Dimitriadou

Report No. 17
June 1998

Report Series

June 1998

SFB
'Adaptive Information Systems and Modelling in Economics and Management Science'

Vienna University of Economics
and Business Administration
Augasse 2–6, 1090 Wien, Austria

in cooperation with
University of Vienna
Vienna University of Technology

http://www.wu-wien.ac.at/am

This paper was accepted for publication in:
Proceedings of ICANN'98, International Conference on Artificial Neural Networks, Skövde, Sweden, September 2–4, 1998.
Springer Verlag.

This piece of research was supported by the Austrian Science Foundation (FWF) under grant SFB#010 ('Adaptive Information Systems and Modelling in Economics and Management Science').

Competitive Learning for Binary Valued Data

Friedrich Leisch Andreas Weingessel
Evgenia Dimitriadou

Institut für Statistik und Wahrscheinlichkeitstheorie
Technische Universität Wien
Vienna, Austria
Email: firstname.lastname@ci.tuwien.ac.at

Abstract

We propose a new approach for using online competitive learning on binary data. The usual Euclidean distance is replaced by binary distance measures, which take possible asymmetries of binary data into account and therefore provide a "different point of view" for looking at the data. The method is demonstrated on two artificial examples and applied on tourist marketing research data.

1 Introduction

Most common clustering methods such as k-means, (hard and soft) competitive learning or neural gas minimize the usual Euclidean distance, i.e., perform least squares estimation (Ripley, 1996). Euclidean distance has a natural connection with normally distributed data, for normal distributions least squares and maximum likelihood estimation coincide. However, for non-normal data other distance measures may have advantageous properties, especially when the data are asymmetric and/or discrete.

In this paper we deal with data from tourist questionnaires. Vienna is one of the worlds largest destinations for city tourism, and marketing is of strategic importance. Homogeneous target groups are very important for advertising, it is desirable to segment tourists into groups which can be addressed separately. E.g., advertising for people interested mostly in cultural events like theater or opera performances could be different than advertising for people which are mostly interested in sightseeing or shopping.

People visiting Vienna are asked to fill out a form about their vacation preferences and general hobbies. These data are subsequently used to compute profiles of "prototypical" tourists. One of our datasets is about typical vacation activities such as tennis, golf, relaxing, shopping, concerts, theater or sightseeing. All answers are boolean, where a "yes" (encoded as 1) in tennis means that the corresponding person likes to play tennis when on vacation.

Clearly this data are not normally distributed and Euclidean distance need not be a good distance measure for this kind of data. Two persons both playing

tennis have the same distance as two persons not playing tennis (in both cases the distance is 0), yet, two persons both playing tennis have more in common than two persons who both do not play tennis.

2 Clustering binary data

Special distance measures for binary data have been used in statistics and cluster analysis for a long time (Anderberg, 1973), but mostly in combination with hierarchical cluster methods (Kaufman & Rousseeuw, 1990). Hierarchical clustering is only feasible for small data sets, not for data mining in huge data sets containing several thousand cases.

Classic non-hierarchical methods such as k-means are hard to combine with binary distance measures because they need the explicit computation of cluster centers. Cluster centers are easy to compute for Euclidean or absolute distance, where they correspond to the mean or median of the cluster, respectively.

Adaptive methods such as online competitive learning have the advantage that they do not need the explicit computation of cluster centers, only the gradient of the distance measure is needed. Hence, neural network clustering methods can be used in combination with binary distance measures.

2.1 Distance measures for binary data

Numerous distance measures for binary data have been proposed in the statistical literature (Anderberg, 1973). None of this can be considered to be "the right" distance for a given real world data set or application. It is up to the user to decide which features in the data set are more important or which differences he wants to find; and then to use an appropriate distance measure to extract these features.

Consider two n-dimensional binary vectors $x = (x_1, \ldots, x_n)'$ and $y = (y_1, \ldots, y_n)'$. We define the 2×2 contingency table

		x		
		1	0	
y	1	α	β	$\alpha + \beta$
	0	γ	δ	$\gamma + \delta$
		$\alpha + \gamma$	$\beta + \delta$	n

where $\alpha = \#\{i : x_i = y_i = 1\}$ denotes the number of components where both x and y are one, $\beta = \#\{i : x_i = 0, y_i = 1\}$, $\gamma = \#\{i : x_i = 1, y_i = 0\}$, and $\delta = \#\{i : x_i = 0, y_i = 0\}$. We restrict ourselves to distance measures of type $D(x, y) = D(\alpha, \beta, \gamma, \delta)$. E.g., the well known Hamming distance (number of different bits of x and y) can be written as $D(x, y) = \beta + \gamma$.

As mentioned in the introduction, we prefer asymmetric distance measures giving more weight to common ones than common zeros, because two common ones represent a common preference of two persons, whereas two zeros simply state that both persons do not like the respective activity. In the following we will concentrate on the following two (closely related) distances:

$$D_1(x, y) = \frac{\beta + \gamma}{\alpha + \beta + \gamma}, \qquad D_2(x, y) = \frac{\beta + \gamma}{2\alpha + \beta + \gamma}$$

D_1 is the famous Jaccard coefficient (see, e.g., Kaufman & Rousseeuw, 1990). Since both do not depend on δ, questions where both subjects answered "no" are ignored. Distance D_1 is the percentage of disagreements in all answers where at least one subject answered "yes". Distance D_2 is similar, but puts more weight on answers where both subjects answered "yes".

2.2 Binary competitive learning

Hard competitive learning is a well-known online stochastic gradient descent algorithm for minimization of the average distance of a given set of data to its closest center. See, e.g., Fritzke (1997) for a survey on competitive learning methods. Let $X_N = \{x^1, \ldots, x^N\}$ denote the data set available for training and let $C_K = \{c^1, \ldots, c^K\}$ be a set of K centers. Further let $c(x) \in C_K$ denote the center closest to x with respect to some distance measure D. Then competitive learning tries to minimize $\sum_{n=1}^{N} D(x^n, c(x^n))$.

The simplest online hard competitive learning algorithm works as follows:

1. *Initialize C_K at random (either by picking K points from X_N or from a random number generator). Set $t = 0$.*
2. *Pick a random point x^i from X_N.*
3. *Let $c_t^j = c(x^i)$ be the center closest to x^i at step t. Update the centers according to $c_{t+1}^j = c_t^j - \eta_t \nabla D(x^i, c_t^j)$ and $c_{t+1}^l = c_t^l$ for $l \neq j$ where ∇D is the gradient of D and η_t is a decreasing learning rate.*
4. *Stop if some convergence criterion (number of iterations, error) is fulfilled, else $t = t + 1$ and goto step 2.*

Two tasks have to be solved for using a binary distance D with this algorithm: First, the gradient of D with respect to the second argument c^j has to be computed, which is straightforward. Second, a real-valued decreasing learning rate will result in non-binary (i.e., real valued) centers; hence, $D(x, c)$ must also be defined for non-binary centers c.

We overcome the second problem by interpreting a real-valued center $c = (c_i, \ldots, c_n)'$ with elements $0 \leq c_i \leq 1$ as a vector of probabilities, where c_i is the probability, that the corresponding component is one. Note, that this approach is closely related to conventional Euclidean clustering, where the cluster centers are equal to the mean of the clusters and therefore equal to the probabilities of having a 1 in the corresponding component, if a variable is binary.

If a component c_i^j of center c^j after the update step 3 is larger than 1, we replace it by 1. Similarly, we replace it by 0 if c_i^j is negative. In our experiments this lead to almost binary centers upon convergence, i.e., the c_i^j of the final centers (almost) equal 0 or 1. Further investigation of this strategy is necessary.

Let x be a (binary) data vector, and let c be a real-valued center with components in $[0, 1]$, then the *expected values* of the contingency table entries are given by $\alpha = x'c$, $\beta = (1 - x)'c$, $\gamma = x'(1 - c)$, $\delta = (1 - x)'(1 - c)$. After some simple algebra we get the gradients of D_1 and D_2

$$\frac{\partial D_1}{\partial c_j} = \frac{(1 - x_j)x'c - x_j(\alpha + \beta + \gamma)}{(\alpha + \beta + \gamma)^2} = \begin{cases} \frac{\alpha}{(\alpha+\beta+\gamma)^2}, & x_j = 0 \\ \frac{-1}{(\alpha+\beta+\gamma)^2}, & x_j = 1 \end{cases}$$

	z_1 x_1, x_2, x_3	z_2 x_4, x_5, x_6	z_3 x_7, x_8, x_9	z_4 x_{10}, x_{11}, x_{12}	m
Type 1	high	high	low	low	1000
Type 2	low	low	high	high	1000
Type 3	low	high	high	low	1000
Type 4	high	low	low	high	1000
Type 5	low	high	low	high	1000
Type 6	high	low	high	low	1000

Table 1: Scenario 1: Symmetric distribution of 0s and 1s.

Similarly,

$$\frac{\partial D_2}{\partial c_j} = \begin{cases} \frac{\alpha}{(2\alpha+\beta+\gamma)^2}, & x_j = 0 \\ \frac{-(\alpha+\beta+\gamma)}{(2\alpha+\beta+\gamma)^2}, & x_j = 1 \end{cases}$$

3 Experiments

3.1 Artificial Data

The questionnaires used by our research partners in tourism marketing use groups of questions concerning related questions. E.g., there are questions whether a person likes sports such as tennis, cycling, swimming, riding or water sports. Another group of questions is about cultural activities such as concerts, theater or museums. Obviously, answers inside such groups are correlated, i.e., a person generally interested in culture is more likely to visit both the theater and concerts than a person not so interested in culture.

This leads to the concept of latent variables, which cannot be measured directly, but through several observable variables. In the example given above, the latent variable would correspond to the feature "generally interested in culture"; the observable variables are concerts, theater and museum.

We compared clustering with Euclidean distance and the binary distances D_1 and D_2 on two artificial examples resembling this structure: We have 4 latent variables $z_1, \ldots, z_4$, which are measured through a varying number of observable variables x_i.

Table 1 shows a scenario where each latent variable z_i is represented by three observable variables x_j. Six types of data are generated, each consisting of 1000 data points. Type 1 has a high probability (80%) that the variables corresponding to z_1 and z_2 are 1, and a low probability (20%) that the remaining variables are 1. Type 2 is exactly reverse, etc. (Dolnicar et al., 1998). Both competitive learning with Euclidean distance and with the two binary distances D_1 and D_2 found the six clusters without problems. We also tried k-means clustering, giving similar results.

Differences between binary and Euclidean distances should emerge if we break the symmetry in the distribution of 0s and 1s. Table 2 shows a asymmetric scenario with 5 clusters in 3 groups (I, II, III); the clusters have also different sizes.

In this examples, the Euclidean-based algorithms did not give stable results, i.e., different restarts of the algorithm typically gave different cluster centers.

		z_1 x_1, x_2	z_2 x_3, x_4, x_5	z_3 x_6, x_7	z_4 x_8, x_9, x_{10}	m
I	Type 1	high	high	low	low	200
	Type 2	high	low	low	low	800
II	Type 3	low	low	high	high	200
	Type 4	low	low	high	low	800
III	Type 5	low	low	low	low	2000

Table 2: Scenario 2: Asymmetric distribution of 0s and 1s.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Size
D_1	c^1	1	1	0	0	0	0	0	0	0	0	1509
	c^2	0	0	0	0	0	1	1	0	0	0	2491
E	c^1	0.09	0.09	0.12	0.10	0.11	0.92	0.91	0.38	0.36	0.34	1027
	c^2	0.37	0.36	0.18	0.18	0.19	0.10	0.08	0.09	0.09	0.10	2973
D_1	c^1	1	1	0	0	0	0	0	0	0	0	1480
	c^2	1	1	1	1	1	0	0	0	0	0	804
	c^3	0	0	0	0	0	1	1	1	1	1	795
	c^4	0	0	0	0	0	1	1	0	0	0	921
E	c^1	0.08	0.08	0.08	0.08	0.09	0.00	0.11	0.11	0.08	0.11	1778
	c^2	0.09	0.09	0.13	0.10	0.10	1.00	0.69	0.17	0.00	0.13	790
	c^3	0.91	0.89	0.35	0.36	0.38	0.10	0.09	0.11	0.11	0.12	1024
	c^4	0.10	0.10	0.10	0.10	0.10	0.86	0.86	0.58	0.93	0.58	408
D_1	c^1	0	0	0	0	0	1	1	1	1	1	1515
	c^2	0	0	0	0	1	0	0	0	0	0	196
	c^3	1	1	1	1	1	0	0	0	0	0	645
	c^4	1	1	0	0	0	0	0	0	0	0	811
	c^5	0	0	0	0	0	1	1	0	0	0	833
E	c^1	0.09	0.09	0.12	0.10	0.10	1.00	0.78	0.27	0.28	0.27	1082
	c^2	0.10	0.10	0.10	0.08	0.07	0.13	0.06	0.67	0.53	0.18	375
	c^3	0.91	0.90	0.35	0.36	0.39	0.12	0.08	0.09	0.10	0.12	1006
	c^4	0.08	0.08	0.08	0.09	0.09	0.00	0.00	0.00	0.00	0.11	1293
	c^5	0.11	0.08	0.10	0.07	0.14	0.00	1.00	0.13	0.12	0.10	244

Table 3: Results for Scenario 2. D_1 denotes binary distance, E Euclidean distance.

Using binary distances gave stable results. Table 3 shows typical results with two, four and five cluster centers: Euclidean distance always found one or more large clusters corresponding to type 5 (many 0s), and could not recover types 1–4 clearly. Using 5 cluster centers did not improve the situation. Binary distance D_1 always recovers types 1–4, but ignores type 5 due to its definition if two or four centers are used. In case of two cluster centers, groups I and II are recovered. Distance D_2 gave similar results.

The clusters corresponding to the first cluster center are larger than the other ones, because draws (a data point is equidistant to 2 centers) have been resolved by assigning the point to the first cluster of equal distance. This way only one cluster with a center close to the zero vector gets "contaminated" by points from type 5. Resolving draws by tossing a coin would yield cluster sizes that are not so different.

Tables 4–9 show crosstables of original types versus cluster membership. Each row shows how many points of a certain type have been assigned to which

		Cluster 1	Cluster 2
I	Type 1	198	2
	Type 2	779	21
II	Type 3	2	198
	Type 4	7	793
III	Type 5	523	1477

Table 4: Scenario 2, binary distance, 2 cluster centers.

		Cluster 1	Cluster 2
I	Type 1	1	199
	Type 2	8	792
II	Type 3	197	3
	Type 4	715	85
III	Type 5	106	1894

Table 5: Scenario 2, Euclidean distance, 2 cluster centers.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4
I	Type 1	5	195	0	0
	Type 2	549	234	9	8
II	Type 3	0	1	197	2
	Type 4	6	14	138	642
III	Type 5	920	360	451	269

Table 6: Scenario 2, binary distance, 4 cluster centers.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4
I	Type 1	3	0	197	0
	Type 2	69	14	714	3
II	Type 3	2	4	1	193
	Type 4	47	577	7	169
III	Type 5	1657	195	105	43

Table 7: Scenario 2, Euclidean distance, 4 cluster centers.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
I	Type 1	0	0	195	5	0
	Type 2	16	11	176	588	9
II	Type 3	197	0	1	0	2
	Type 4	208	6	10	2	574
III	Type 5	1094	179	263	216	248

Table 8: Scenario 2, binary distance, 5 cluster centers.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
I	Type 1	0	0	197	2	1
	Type 2	10	12	706	58	14
II	Type 3	158	38	1	0	3
	Type 4	710	22	8	4	56
III	Type 5	204	303	94	1229	170

Table 9: Scenario 2, Euclidean distance, 5 cluster centers.

cluster, respectively. For 2 cluster centers (Tables 4, 5) the partition of binary and Euclidean clustering are very similar, although the cluster centers are different.

Using four cluster centers (Table 6), binary clustering recovers types 1–4, the crosstable has a clear block structure. Note that types 1 and 2 overlap due to the construction of the scenario, hence they get also mixed by the cluster algorithm. The same is valid for types 3 and 4, respectively. Type 5 is ignored and distributed over the 4 clusters.

Euclidean clustering (Table 6) uses one center for type 5, hence at least one of types 1–4 cannot be recovered. In the example given, types 1 and 2 are put together in cluster 3. Using 5 centers could not improve the performance of Euclidean clustering on Scenario 2.

3.2 Tourism Data

As described above we also clustered binary data from tourist questionnaires with 12 variables and 15066 cases. Results can be seen in Table 10. The binary clustering shows a clear structure. There is one big cluster (# 3) featuring "classical" tourist characteristics such as swimming, relax, shopping or sightseeing. Additionally there are 3 smaller clusters of almost the same size. # 4 can be seen as "typical tourist" similar to # 3, but with additional Viennese specialities like "Heurigen". # 2 is a more sportive kind of tourist (additionally cycling and water sports). Finally # 1 is a type of tourists "doing nothing but relax".

The results of Euclidean clustering are much more fuzzy. All clusters have almost the same size (3200-4800) and their profiles are not as distinctive as the results from binary clustering. Some clusters are similar to the results from binary clustering, but one needs additional postprocessing like thresholding to be able to read the results. However, by thresholding of the centers one alters the partition of the data set, hence the partition must not correspond to the used distance anymore.

4 Summary

A—due to our knowledge—new approach for competitive learning with asymmetric binary distance measures has been proposed. This way, we provide a different "point of view" for looking at the data. For real world data, there is no way of determining which clustering algorithm is "best", because the data generating process is unknown. Typical clustering algorithms try to minimize a

	tennis		riding		swim		relax		sightsng		theater		
	cycling		golf		w.sport		shop		museum		heuriger		size
D_1	0	0	0	0	0	0	1	0	0	0	0	0	2224
	0	1	0	0	1	1	1	1	1	0	0	0	2804
	0	0	0	0	1	0	1	1	1	1	0	0	7547
	0	0	0	0	0	0	1	1	1	1	1	1	2491
E	.07	.10	.02	.03	.32	.08	.00	.20	.55	.39	.10	.17	3268
	.06	.07	.02	.03	.40	.14	.77	.78	.95	1.0	.23	.28	4803
	.06	.03	.02	.02	.48	.11	1.0	.45	.46	.04	.04	.09	4090
	.23	.98	.09	.07	.95	.40	.81	.62	.72	.48	.13	.30	2905

Table 10: Clustering results for tourism data.

loss function depending on differences between cluster centers and data points. Using several different distance measures (and knowing about their special characteristics) can give valuable further insight into a data set.

Of course, one is not limited to the two binary distance measures proposed in this paper. A lot of different distance measures can be found in the literature, most of which can easily be adopted to our framework. Also, there are more popular competitive learning algorithms (e.g., soft competitive learning, neural gas or Kohonen maps), which could be generalized for binary distance measures in the same way. All these questions are currently under investigation.

References

Anderberg, M. R. (1973). *Cluster analysis for applications.* New York, USA: Academic Press Inc.

Dolnicar, S., Leisch, F., Weingessel, A., Buchta, C., & Dimitriadou, E. (1998). *A Comparison of Several Cluster Algorithms on Artificial Binary Data Scenarios from Tourism Marketing.* Working Paper Series 7, SFB "Adaptive Information Systems and Modeling in Economics and Management Science", http://www.wu-wien.ac.at/am/workpap.html.

Fritzke, B. (1997). Some competitive learning methods. http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/.

Kaufman, L. & Rousseeuw, P. J. (1990). *Finding Groups in Data.* New York, USA: John Wiley & Sons, Inc.

Ripley, B. D. (1996). *Pattern recognition and Neural networks.* Cambridge, UK: Cambridge University Press.