Feature Selection in k-Median Clustering *

Olvi L. Mangasarian† Edward W. Wild‡

Abstract

An effective method for selecting features in clustering unlabeled data is proposed based on changing the objective function of the standard k-median clustering algorithm. The change consists of perturbing the objective function by a term that drives the medians of each of the k clusters toward the (shifted) global median of zero for the entire dataset. As the perturbation parameter is increased, more and more features are driven automatically toward the global zero median and are eliminated from the problem until one last feature remains. An error curve for unlabeled data clustering as a function of the number of features used gives reduced-feature clustering error relative to the "gold standard" of the full-feature clustering. This clustering error curve parallels a classification error curve based on real data labels. This justifies the utility of the former error curve for unlabeled data as a means of choosing an appropriate number of reduced features in order to achieve a correctness comparable to that obtained by the full set of original features. For example, on the 3-class Wine dataset, clustering with 4 selected input space features is comparable to within 4% to clustering using the original 13 features of the problem.

Keywords *clustering, k-median, feature selection, non-smooth optimization, centered data, regularization.*

1 Introduction

Both the k-median and k-mean clustering algorithms for unlabeled data can be considered as unconstrained optimization problems [3, 11, 5, 2]. Both algorithms choose k cluster centers that attempt to minimize the sum of a distance measure between each point and the closest cluster center. The "distance measure" for the k-median algorithm is the 1-norm distance, whereas for the k-mean algorithm it is the *square* of the 2-norm distance [3]. Neither problem has a convex objective function, and finding a global solution to either problem may be NP-hard. However, the k-median objective is a *concave* function and a local solution to it can be quickly found in a finite number of steps [3], even though, like the k-mean objective function, it is nondifferentiable. We shall therefore utilize this fast algorithm for our feature selection approach.

The idea behind our approach is the following. We first shift all the data so that its median becomes the origin in the n-dimensional input space of the data. Then for a desired number k of clusters we determine k cluster centers that minimize the sum of 1-norm distances between each point in the given dataset to the closest cluster center, *plus* the sum of the 1-norm distances of each of the k cluster centers to the origin in n-space weighted by some positive parameter ν. This minimization is achieved by finding a stationary point of the nondifferentiable concave objective function for a fixed value of the parameter ν. Starting with $\nu = 0$, k cluster centers are found using the ordinary k-median algorithm in the full n-dimensional input space of the dataset. This gives rise to an unlabeled clustering of the data, the "gold standard" to which all subsequent clusterings in smaller dimensional subspaces are compared. As ν is increased, one or more input space features at a time become zero and one of these features is deleted from the problem. The corresponding k-median clustering is then obtained in this subspace and its "correctness" is judged by comparison with the gold standard k-median clustering of the full n-dimensional original input space. This procedure, which is extremely fast, is continued until only one input space feature is left. The desired number of reduced features is then picked as that which gives a comparable correctness to that of the full n-dimensional k-median clustering. A related method for feature selection is given in [12] for labeled classification where centroids of known classes are shrunk towards the global centroid of all classes. Our approach can also be interpreted as a *regularization* procedure that is generally used for ill-posed problems [13, 1] and in support vector machines [4], where in addition to fitting given data, the problem variables are also driven to zero parametrically in order to improve generalization correctness.

To clarify our terminology further, we refer to the three-class Wine dataset described in more detail in Section 3. This dataset which consists of 178 points

*The research described in this Data Mining Institute Report 04-01, January 2004, was supported by National Science Foundation Grant CCR-0138308 and by the Microsoft Corporation.

†Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *olvi@cs.wisc.edu*.

‡Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *wildt@cs.wisc.edu*.

in a 13-dimensional input feature space, has one of three labels associated with each data point. We do *not* use any of these labels in our feature-selecting k-median (FSKM) Algorithm 2.1. We first start clustering with the ordinary k-median algorithm in the original 13-dimensional space. We then use our theoretically derived criterion (2.11) to delete one appropriately selected feature and apply the k-median algorithm again in the reduced feature space. This process is continued until only one feature remains. After each feature deletion, we measure the *clustering error* by comparing the clustering labels in the reduced space with those of the gold standard clustering labels generated by using the original 13 features of the problem. We continue this clustering error evaluation until one feature is left. We then decide on how many features to keep based on the clustering error we wish to tolerate at a corresponding reduced number of features. We justify this clustering error criterion that does not use any of the original labels of the data by comparing it with the *classification error* that utilizes the original data labels for each clustering obtained by our approach for a reduced number of features. Figure 1 for the Wine dataset shows that the *clustering error* curve closely parallels the *classification error* curve. This justifies both the use of our label-free feature-selecting approach as well as its clustering error prediction for a reduced set of features *without* using any data labels.

We briefly outline the contents of the paper now. In Section 2 we derive the theory behind our feature selecting k-median (FSKM) algorithm and state our algorithm for increasing values of ν. Section 3 gives computational and graphical results that show the effectiveness and utility of FSKM. Section 4 concludes the paper.

A word about our notation and background material follows. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$ and the p-norm of x, $(\sum_{i=1}^{n}|x_i|^p)^{\frac{1}{p}}$, will be denoted by $\|x\|_p$. For a matrix $A \in R^{m\times n}, A_i$ is the ith row of A which is a row vector in the n-dimensional real space R^n. A column vector of ones of arbitrary dimension will be denoted by e. The symbol $:=$ denotes definition. For a convex function $f : R^n \longrightarrow R^1$ that is nondifferentiable, such as $\|x\|_1$, a subgradient $\partial(x) \in R^n$ exists [10, 9] with the property that:

$$(1.1) \quad f(y) - f(x) \geq \partial f(x)'(y - x),\ \forall x, y \in R^n.$$

Thus for $\|x\|_1$, $x \in R^n$ and $i = 1, \ldots, n$:

$$(1.2) \qquad (\partial\|x\|_1)_i = \begin{cases} -1 & \text{if } x_i < 0 \\ \in [-1, 1] & \text{if } x_i = 0 \\ +1 & \text{if } x_i > 0 \end{cases}$$

The subgradient plays the role of a gradient for differentiable convex functions, except that it is not unique. Thus a necessary and sufficient condition for x to be a minimizer of $f(x)$ is that

$$(1.3) \qquad \partial f(x) = 0.$$

For a countable set S, $card(S)$ denotes the cardinality of S, that is the number of elements in S.

2 Feature Selecting k-Median (FSKM) Theory and Algorithm

The k-median clustering algorithm [3] consists of two basic steps. Given k initial or intermediate cluster centers, the *first* step consists of assigning each point to the closest cluster center using the 1-norm distance. The *second* step consists of generating k new cluster centers, each being the median of each cluster. It is the second step that we shall modify, in order to remove possibly irrelevant input space features from the problem, as follows. Since the median of a cluster is the point (or set of points) that minimizes the sum of the 1-norm distances to all the points in the cluster, we shall perturb this minimization problem by adding to its objective function a weighted term with weight ν consisting of the 1-norm distance to global median of zero for the entire dataset. As the weight ν gets sufficiently large, all the features will become zero and are eliminated from the problem. Conversely, if $\nu = 0$, then we have the ordinary k-median algorithm. We derive now the optimality condition for minimizing the nondifferentiable objective function for the perturbed objective function for this step of the modified k-median algorithm.

Let the given dataset, consisting of m points in R^n, be represented by the matrix $A \in R^{m\times n}$. We shall assume without loss of generality that a median of the m rows of A is $0 \in R^n$. Assume further, that k clusters have been generated by the k-median algorithm and are represented by the k submatrices of A:

$$(2.4) \quad A^\ell \in R^{m(\ell)\times n}, A^\ell_i = A_{i\in J(\ell)},\ \ell = 1, \ldots, k,$$

where $J(\ell) \subset \{1, \ldots, m\}$, $\ell = 1, \ldots, k$, is a partition of $\{1, \ldots, m\}$. The k perturbed optimization problems that need to be solved at this second step of the modified k-median algorithm consist of the following k unconstrained minimization problems. Find k cluster centers $c^\ell \in R^n$, $\ell = 1, \ldots, k$, with one or more components being zero, depending on the size of ν. Each $c^\ell \in R^n$, $\ell = 1, \ldots, k$ is a solution of:

$$\min_{c \in R^n} \sum_{i \in J(\ell)} \|A_i - c\|_1 + \nu\|c\|_1,\ \ell = 1, \ldots, k. \tag{2.5}$$

Since each of these problems is *separable* in the components c_j, $j = 1, \ldots, n$ of c, we can consider the following *1-dimensional* minimization problem for each component c_j, which we denote for simplicity by $c \in R^1$, and for $a_i := A_{ij}$, $i \in J(\ell)$ for a fixed $j \in \{1, \ldots, n\}$ as follows:

$$\min_{c \in R^1} \sum_{i \in J(\ell)} |c - a_i| + \nu|c|,\ \ell = 1, \ldots, k. \tag{2.6}$$

Here $A_{J(\ell)}$ denotes the subset of the rows of A that are in cluster ℓ. Setting the subgradient (see Equations (1.1)-(1.3)) of the objective function of (2.6) equal to zero gives the following necessary and sufficient optimality condition for a fixed $j \in \{1, \ldots, n\}$ and for a fixed cluster $\ell \in \{1, \ldots, k\}$:

$$\begin{aligned} & card\{i|c > a_{i \in J(\ell)}\} - card\{i|c < a_{i \in J(\ell)}\} \\ & +[-1,1] \cdot card\{i|c = a_{i \in J(\ell)}\} \\ & +\nu \cdot \left\{ \begin{array}{ll} -1 & \text{if } c < 0 \\ {[-1,1]} & \text{if } c = 0 \\ +1 & \text{if } c > 0 \end{array} \right\} = 0. \end{aligned} \tag{2.7}$$

Henceforth, $[-1, 1]$ denotes some point in the closed interval $\{x| -1 \le x \le 1\}$. Thus, for a cluster center c to be zero, for a fixed $j \in \{1, \ldots, n\}$, $a_i := A_{ij}$, and for a fixed cluster $\ell \in \{1, \ldots, k\}$, we need to have:

$$\begin{aligned} & card\{i|0 > a_{i \in J(\ell)}\} - card\{i|0 < a_{i \in J(\ell)}\} \\ & +[-1,1]card\{i|0 = a_{i \in J(\ell)}\} + [-1,1] \cdot \nu = 0. \end{aligned} \tag{2.8}$$

Simplifying this expression by replacing the first $[-1, 1]$ by the zero subgradient and solving for ν, we have that:

$$\nu = \frac{card\{i|0 < a_{i \in J(\ell)}\} - card\{i|0 > a_{i \in J(\ell)}\}}{[-1,1]}, \tag{2.9}$$

which is satisfied if we set:

$$\nu \ge |card\{i|0 < a_{i \in J(\ell)}\} - card\{i|0 > a_{i \in J(\ell)}\}|. \tag{2.10}$$

Hence we can state the following result based on the above analysis.

PROPOSITION 2.1. **Cluster Center with Selected Features** *A solution c to the perturbed cluster center optimization problem (2.5) has zero components $c_j = 0$ for each $j \in \{1, \ldots, n\}$, such that:*

$$\nu \ge |card\{i|0 < A_{i \in J(\ell),j}\} - card\{i|0 > A_{i \in J(\ell),j}\}|. \tag{2.11}$$

It follows that if we set ν large enough one or more input space features are killed. Hence we can gradually increase ν from zero and systematically kill at least one feature at a time. This property suggests the following algorithm.

ALGORITHM 2.1. FSKM: Feature Selecting k-Median Algorithm

1. Shift the dataset $A \in R^{m \times n}$ such that $0 \in R^n$ is its median.

(i) Use the k-median the algorithm to cluster into k clusters.

(ii) For each input space component $j \in \{1, \ldots, n\}$ and for each cluster $A_{J(\ell)}$, $\ell \in \{1, \ldots, k\}$ compute:

$$\nu_j^\ell = |card\{i|0 < A_{i \in J(\ell),j}\} - card\{i|0 > A_{i \in J(\ell),j}\}|. \tag{2.12}$$

(iii) Delete feature(s) $\bar{j}$ by deleting column(s) $A_{\cdot\bar{j}}$ for which:

$$\nu_{\bar{j}} = \min_{1 \le j \le n} \max_{1 \le \ell \le k} \nu_j^\ell. \tag{2.13}$$

(iv) Stop if A has no columns remaining, else let $A = \bar{A} \in R^{m \times \bar{n}}$, $n = \bar{n}$, where $\bar{A}$ is the matrix with reduced columns.

(v) Go to to (i).

We note that Step (iii) in the FSKM Algorithm above determines precisely which input space feature(s) will be deleted next, based on successively increasing values of the perturbation parameter ν. Thus, formula (2.13) of Step (iii) sets apart our algorithm from a lengthy greedy n-choose-1 approach that systematically deletes one feature at a time. Such a greedy approach chooses to delete the feature which minimizes the clustering error for the remaining features. This procedure is repeated n times until one feature is left. Hence, instead of n applications of the k-median algorithm needed by FSKM, a greedy approach would need $\frac{n(n+1)}{2}$ applications of the k-median algorithm.

We turn now to our computational results to show the effectiveness of the FSKM Algorithm.

3 Computational Results

To illustrate the performance of our algorithm, we tested it on five publicly available datasets, four from the UCI Machine Learning Repository [7] and one available at [8]. We ran Algorithm 2.1 30 times on each dataset, and we report average results. If Algorithm 2.1 produced multiple candidate features for elimination in Step (iii), then only one randomly chosen feature

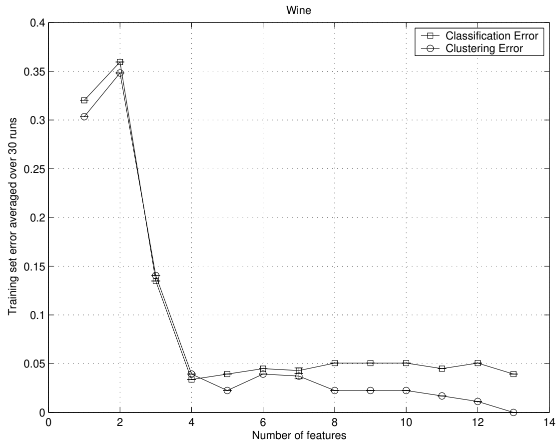

Figure 1: **Error curves for the 3-class Wine dataset with 178 points in 13-dimensional space are plotted as a function of the number features selected by FSKM. The average range of ν computed by (2.13) was from 42 to 55. Note that the low variance between runs on this dataset makes the error bars essentially invisible.**

from this set was eliminated. The k-median algorithm was initialized with centers chosen by the following procedure which is similar to that of [3]. For each feature, $4k$ bins of equal size were created. The data was sorted into these bins, and the k initial centers were chosen by taking the midpoint of the k most populous bins for each feature. Consider using this procedure for $k = 2$. One initial center will have each coordinate be the midpoint of the most populous bin for the corresponding feature, while the other initial center will have each coordinate be the midpoint of the second most populous bin for the corresponding feature. The decision to use $4k$ bins was made arbitrarily and not adjusted while developing the algorithm or performing the experiments.

Figure 1 gives results for the Wine dataset [7]. The two curves shown are the classification error and the clustering error. The *classification error* curve, marked by squares, is computed by labeling members of each cluster with the majority label of the cluster, where the labels are the actual class labels from the dataset. These class labels are used only in generating the classification error curve and *not* in obtaining the clusters. The error is the number of incorrectly classified examples divided by the number of examples in the dataset. The entire dataset is used both for the clustering and evaluation of the error. No data is left out. The *clustering error* curve, marked by circles, is computed by accepting the clusters produced by k-median on the full-featured dataset as the gold standard labeling, and then using the following procedure for computing the error without using any class labels, as would be the case for unlabeled data clustering. For each reduced dataset, members of each cluster are marked with the majority gold standard label of that cluster. The gold standard labels are used only in generating the clustering error curve and *not* in obtaining the clusters. Note that the clustering error on the full-featured dataset is always zero by definition. Error bars show one sample standard deviation above and below each point. Total time to generate the error curves which entails running the k-median algorithm 390 times and plotting the the error curves, all within MATLAB [6], took 205.1 seconds on a 650MHz, 256MB RAM desktop machine running Red Hat Linux, Version 9.0.

The curves in Figure 1 show that the clustering error curve increases slightly as the input space dimensionality is reduced from 13 features to 4 features, and then increases very sharply as the data dimensionality is further reduced from 4 features down to 2 features. The classification error curve decreases slightly as the data dimensionality is reduced from 13 to 4 features, and then increases similarly to the clustering error curve as the number of features is reduced from 4 to 2. As the number of features is reduced from 2 to 1, both curves decrease. The number of features can be reduced to 4 from 13 while keeping the clustering error less than 4% and decreasing the classification error by 0.56 percentage points.

One key observation to make about Figure 1 and subsequent figures is the following. Since the real world application of FSKM is to unlabeled data, we can only generate a *clustering* error curve similar to that of Figure 1. This curve will help us decide on the magnitude of error we wish to tolerate, which determines how many and which features to keep. The validity of such a procedure is based on the parallelism between the clustering error curve based on unlabeled data, and the classification error curve based on the labels of the datasets in the current experiments.

The results of our algorithm on the Votes dataset [7] are in Figure 2. The procedure for generating the curves is exactly the same as described above for the Wine dataset. Note that both the classification and clustering error increase slightly as the number of features is reduced from 16 to 12. Then the classification error increases briefly and then tends to decrease while the clustering error tends to increase slightly as the number of features is reduced from 12 to 3. Finally, both the classification and the clustering error increase more sharply as the number of features is reduced from 3 to 1. After reducing the number of features down to 3, the

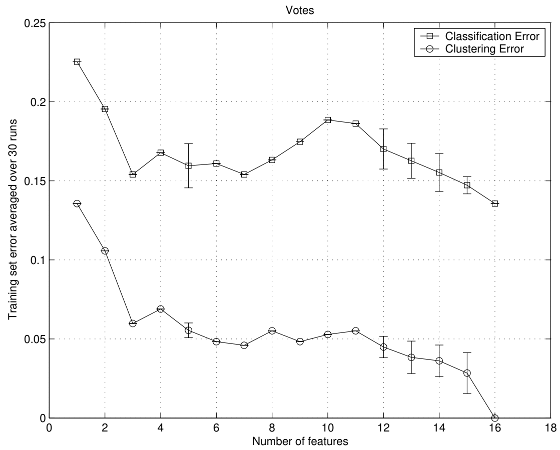

Figure 2: **Error curves and variance bars for the 2-class Votes dataset with 435 points in 16-dimensional space are plotted as a function of the number features selected by FSKM. The average range of ν computed by (2.13) was from 0 to 192.**

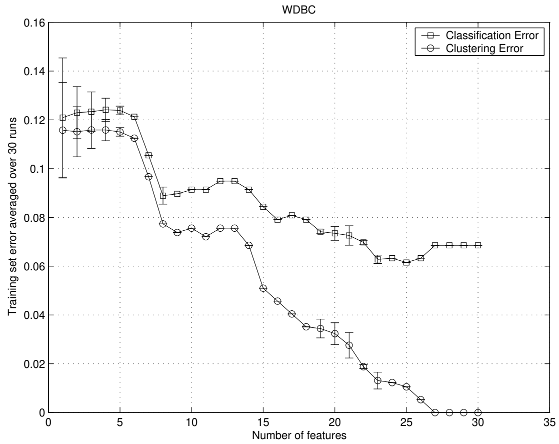

Figure 3: **Error curves and variance bars for the 2-class WDBC dataset with 569 points in 30-dimensional space are plotted as a function of the number features selected by FSKM. The average range of ν computed by (2.13) was from 188 to 284.**

clustering error is less than 10%, and the classification error has only increased by 1.84 percentage points.

Results for the WDBC dataset [7] are in Figure 3. For this dataset, the classification error does not increase as much as the clustering error as the number of features is reduced from 30 to 14. At that point, the two curves mirror one another closely as the number of features is reduced further. Note that reducing the number of features from 30 to 27 causes no change in clustering or classification error. Reducing the number of features to 7 keeps the clustering error less than 10%, while increasing the classification error by 3.69 percentage points.

Figure 4 shows the results for the Star/Galaxy-Bright dataset [8]. For this dataset, the classification and clustering error curves behave differently. However, note that the clustering error curve tends to increase only slightly as the number of features decreases. This behavior is what we want. Overall, the classification error curve decreases noticeably until 6 features remain and then begins to increase, indicating that some of the features may be obstructing the classification task. The problem can be reduced to 4 features and still keep the clustering error under 10% while decreasing the classification error by 1.42 percentage points from the initial error using 14 features.

Results for the Cleveland Heart dataset [7] are in Figure 5. Note that although the increase in clustering error when reducing from 13 features to 9 features is

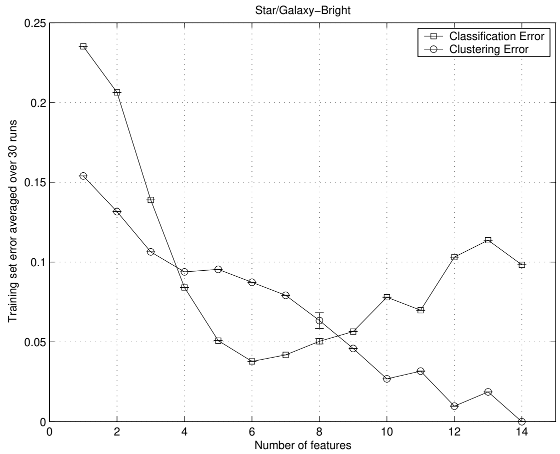

Figure 4: **Error curves and variance bars for the 2-class Star/Galaxy-Bright dataset with 2462 points in 14-dimensional space are plotted as a function of the number features selected by FSKM. The average range of ν computed by (2.13) was from 658 to 1185.**

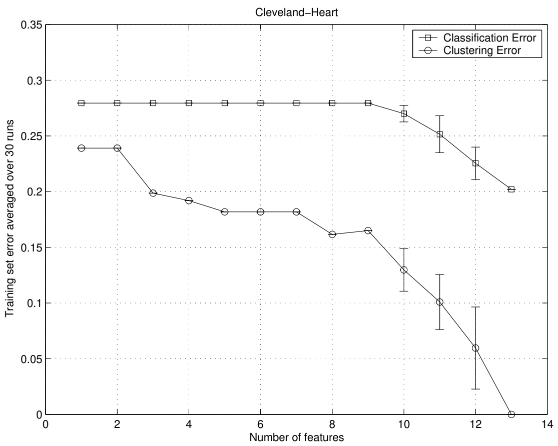

Figure 5: **Error curves and variance bars for the 2-class Cleveland Heart dataset with 297 points in 13-dimensional space are plotted as a function of the number features selected by FSKM. The average range of ν computed by (2.13) was from 0 to 113.**

very large, subsequent increases are not so severe. In addition, the classification error curve behaves similarly to the clustering error curve in the sense that both curves have the greatest increase going from 13 features to 9 features. Using FSKM to remove 5 features causes the clustering error to be less than 17%, and increases classification error by 7.74 percentage points.

4 Conclusion

FSKM is a fast and efficient method for selecting features of unlabeled datasets that give clusters that are similar to clusters obtained in the full dimensional space of the original data. In addition, features selected by FSKM may be useful for labeled feature selection. For example, the 6 features selected by FSKM for the Star/Galaxy-Bright dataset gave an error of 3.78% compared with 9.83% error with the full 14 features. Using the features chosen by FSKM could eliminate the costly search for the best 6 out of 14 features. Exhaustively searching for those 6 features would require $\binom{14}{6} = 3003$ k-median runs as opposed to our 9 k-median runs.

It is hoped that future research into the theory used here to justify the feature selection procedure of FSKM will have further application to other algorithms of machine learning and data mining.

References

[1] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7:108–116, 1995.

[2] P. S. Bradley, Usama M. Fayyad, and O. L. Mangasarian. Data mining: Overview and optimization opportunities. *INFORMS Journal on Computing*, 11:217–238, 1999. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-01.ps.

[3] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 368–374, Cambridge, MA, 1997. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-03.ps.

[4] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 171–203, Cambridge, MA, 2000. MIT Press.

[5] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1988.

[6] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2001. http://www.mathworks.com.

[7] P. M. Murphy and D. W. Aha. UCI machine learning repository, 1992. www.ics.uci.edu/∼mlearn/MLRepository.html.

[8] S. Odewahn, E. Stockwell, R. Pennington, R. Humphreys, and W. Zumach. Automated star/galaxy discrimination with neural networks. *Astronomical Journal*, 103(1):318–331, 1992.

[9] B. T. Polyak. *Introduction to Optimization*. Optimization Software, Inc., Publications Division, New York, 1987.

[10] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[11] S. Z. Selim and M. A. Ismail. K-Means-Type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:81–87, 1984.

[12] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[13] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill–Posed Problems*. John Wiley & Sons, New York, 1977.