Chapter 4
Clustering and Classification

Abstract Optimization is another important tool that helps in defining, designing, and in model selection in various machine learning tasks including dimensionality reduction, clustering, and classification. We discuss, in this chapter, the role of optimization in feature selection, feature extraction, clustering, and classification.

Keywords Optimization · Regularization · Feature selection · Classification · Clustering

4.1 Introduction

We have examined the roles of centrality and diversity in search and representation earlier. In the discussion, we had considered their roles in clustering and classification also. In this chapter, we will consider more details on the roles of centrality and diversity in clustering and classification.

4.2 Clustering

We have observed that a *unifying representation* of both hard and soft clustering is through matrix factorization. Specifically, if we represent the set of n l-dimensional points,

$$\mathcal{X} = \{x_1, x_2, \ldots, x_n\},$$

to be clustered as the rows of a matrix $A_{n\times l}$, then we can factorize it into the product of matrices $B_{n\times K}$ and $C_{K\times l}$ where

- $B_{n\times K}$ is the *cluster/topic assignment matrix*, B_{ik} is the membership or importance of cluster k to pattern i for $i = 1, \ldots, n$ and $k = 1, \ldots, K$.

M. N. Murty and A. Biswas, *Centrality and Diversity in Search*,
SpringerBriefs in Intelligent Systems, https://doi.org/10.1007/978-3-030-24713-3_4

- $C_{K\times l}$ is the *cluster/topic description matrix* where C_{kj} indicates the importance of feature j to cluster $k, j = 1, \ldots, l$ and $k = 1, \ldots, K$.

An observation in such a representation is that any *data matrix* $A_{n\times l}$ is a $\Re^{n\times l}$ structure. Further, any $n \times l$ matrix A has its

$$rank(A) = row - rank(A) = column - rank(A)$$

where row rank is the number of *linearly independent* rows in A and column rank is the number of linearly independent columns of A. Because clustering is grouping of rows (n patterns) and dimensionality reduction deals with columns (l features), their ranks being equal means the number of clusters and number of features are equal from the *linear independence* view.

We examine one representative each from clustering, feature selection, and feature extraction with the help of an example data set.

Example 4.1 Let $A_{4\times 3}$ be a matrix representing 4 patterns in a 3-dimensional space given by

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

For the sake of simplicity rows are replicated, rows 1 and 3 are identical and rows 2 and 4 are also same. We will examine clustering first using matrix A.

4.2.1 Clustering-Based Matrix Factorization

Clustering the 4 rows of A into $K = 2$ clusters gives us $\{A_1, A_3\}$ and $\{A_2, A_4\}$, where A_i is the ith row of A. This is obtained by selecting the first two rows, diverse rows, as the initial cluster centers and assigning the remaining two points, third and fourth rows based on nearness to the selected points. The centroids of clusters c_1 and c_2 are $(1, 0, 1)$ and $(0, 1, 0)$ respectively. This gives us

- The assignment matrix $B_{4\times 2}$ to be

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- The cluster description matrix $C_{2\times 3}$ has the 2 centroids as its rows given by

$$C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Note that

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = BC$$

Observe that because of the simplicity of the data, any clustering algorithm will lead to the same partition and if centroids of clusters or other representatives are used, again we get the same C matrix. However, some important observations are

- In general $A \approx BC$. In this example, $A = BC$ because each centroid coincides with 2 out of the 4 patterns.
- In most of the practical applications A will have elements from $\Re^+ \cup \{0\}$. The factorization is called non-negative matrix factorization (*NMF*) if elements of B and C are nonnegative reals.
- It is known that in such a *NMF* if any two out of A, B, C are given, then getting the third one is simple. In *KMA* based clustering, given A, getting the centroids and the C matrix are reasonably straightforward.
- In *NMF*, in general, we are given A and finding B and C is posed as the optimization problem

$$\min_{B,C} ||A - BC||_F \; s.t. B \geq 0, C \geq 0$$

where $||A - BC||_F$ is the squared Frobenius norm or element-wise difference between the $n \times l$ matrices A and BC.

4.2.2 Feature Selection

It is easy to observe that columns 1 and 3 are identical in matrix A. So, by grouping the columns and identifying diverse columns gives rise to using either columns 1 and 2 or columns 2 and 3. Suppose we use columns 1 and 2 to represent matrix B, then

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Consequently we get the same C matrix as earlier that is given by

$$C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

This simple example is ideally suited to illustrate the equivalence between features and clusters by using feature selection. Further, all the matrices involved are non-negative. So, this is an example *NMF*.

4.2.3 Principal Component Analysis (PCA)

Principal components, *PCs*, are popular linear feature extractors. Given the data represented in l-dimensional space using features $f_1, f_2, \ldots, f_l$. An extracted feature, f, is a linear combination that is obtained from the given l features. So, $f = \sum_{i=1}^{l} \alpha_i f_i$ where α_i is the weight or importance associated with the given feature f_i. In general, we can extract features using nonlinear combinations also, but that may be time consuming.

In *PCA*, the features extracted are the eigenvectors of the covariance matrix of the data. These are popularly called the principal components (*PC*s). There could be up to l PCs when A is an $n \times l$ matrix. These are ordered based on decreasing order of the respective eigenvalues. Some properties of *PCA* are

1. Because the underlying matrix is the covariance matrix, these eigenvalues are variances in the direction of the respective *PCs*. So, the first *PC* is in the maximum variance direction of the data.
2. The covariance matrix is a *symmetric matrix*. So, the eigenvectors (*PCs*) are orthogonal to each other when the corresponding eigenvalues are distinct.
3. If we take the first K out of l possible *PCs* to represent the data, it corresponds to optimizing a criterion function that captures average deviations between the given patterns in the l-dimensional space and a K-dimensional space. This minimization leads to K *PCs* as the optimal new features that are linear combinations of the given features.
4. These *PCs* provide uncorrelated directions under some conditions.

Considering the data matrix $A_{4\times 3}$, the corresponding sample covariance matrix is obtained first by getting the zero-mean normalized matrix, A^n is

$$A^n = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

and then the covariance matrix Σ given by $A^{nt}A^n$ which is

$$\Sigma = \frac{1}{4} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}.$$

The eigenvalues of Σ are 3, 0, and 0. So, the top two eigenvectors are $(1, -1, 1)^t$ and $(1, 2, 1)^t$. They are orthogonal. To make them orthonormal we normalize them to make them *unit norm* vectors to get the two *PCs* to be

$$\left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)^t, \left(\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right)^t$$

So, C_{pc} matrix is given by

$$C_{pc} = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}}. \end{bmatrix}$$

This gives us the B_{pc} matrix to be

$$B_{pc} = \begin{bmatrix} \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}}. \end{bmatrix}$$

Note that the 4 rows of B_{pc} are obtained by projecting the 4 patterns onto these two *PCs*. Projecting the first row (pattern) of A, that is $(1, 0, 1)$ gives us $(\frac{2}{\sqrt{3}}, \frac{2}{\sqrt{6}})$. The second row projection gives us $(-\frac{1}{\sqrt{3}}, \frac{2}{\sqrt{6}})$. Putting them all together, we have $A = B_{pc}C_{pc}$ given by

$$A = \begin{bmatrix} 1\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 1 \\ 0\ 1\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}}. \end{bmatrix}$$

This factorization is indicating how the 3-dimensional points are represented in the 2-dimensional *PC* space. When the rank of the matrix A is 2, which is the case here, we can represent it using 2 orthogonal basis vectors as indicated in the equality between A and $B_{pc}C_{pc}$. Also this is not an *NMF* as there are negative elements in both B_{pc} and C_{pc}.

However, the second eigenvalue of Σ is 0. So, the variance is captured by the first *PC* itself. In such a case, using the first *PC* we get approximation

$$A = \begin{bmatrix} 1\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 1 \\ 0\ 1\ 0 \end{bmatrix} \approx \begin{bmatrix} \frac{2}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}}. \end{bmatrix}$$

Here B_{pc} is the projection of the 4 rows of A onto the first PC.

This approximation amounts to $||A - B_{pc}C_{pc}||_F = \frac{16}{3}$, where each pattern is approximated with an error of $\frac{4}{3}$. However, the 1-dimensional representation is able to discriminate between the patterns 1 and 3 from the patterns 2 and 4. There could be other approximations with a lesser value of 4 as the squared Frobenius norm for the following.

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \approx \begin{bmatrix} \sqrt{3} \\ 0 \\ \sqrt{3} \\ 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Even though some discrimination between elements of the two clusters is exhibited in the *PCs* space, in general the first K *PCs* may not be able to retain the discrimination present in the l-dimensional space. The reason is that the underlying optimization is planned to reduce the expected squared deviation between the patterns in the l-dimensional space and their representations in the K-dimensional space specified by minimization of

$$E[(x^l - x^K)^t(x^l - x^K)],$$

where x^l and x^K are original pattern and its approximation, that is represented in the $K(< l)$ dimensional space respectively and E is the expectation operation. The following high-level summary of the properties will link the above criterion function and the *PCs*.

- Note that x^l is a vector in a l-dimensional space. So, it can be uniquely represented using l orthonormal basis vectors $v_1, \ldots, v_l$. Specifically,

$$x^l = \sum_{i=1}^{l} d_i v_i$$

  where d_is are some real numbers, for $i = 1, \ldots, l$.
- Now x^K may be viewed as coming out of K-dimensional subspace and

$$x^K = \sum_{i=1}^{K} d_i v_i$$

- The error, by exploiting the orthonormality property of $v_1, v_2, \ldots, v_l$ will reduce to

$$error = E[(x^l - x^K)^t(x^l - x^K)] = \sum_{i=K+1}^{l} v_i^t \Sigma v_i = \sum_{i=K+1}^{l} v_i^t v_i \lambda_i = \sum_{i=K+1}^{l} \lambda_i$$

  where v_i and λ_i are an eigenvector and the respective eigenvalue of Σ.

- This error is minimized when $\lambda_{K+1}, \lambda_{K+2}, \ldots, \lambda_l$ are smaller. This indicates that $\lambda_1, \lambda_2, \ldots, \lambda_K$ need to be the larger eigenvalues. Correspondingly, $v_1, v_2, \ldots, v_K$ are the eigenvectors that characterize x^K.
- So, first *K PCs* are the eigenvectors of Σ which can uniquely characterize projection of each pattern in the K space.

So, error considered is intuitively appealing as it minimizes the average error between patterns in the l space and the respective projections in the K *PC*s space. This optimization is reproduction friendly and the basis vectors in the K space capture the variance in the data to the best possible extent. However, there is no guarantee that the *K PCs* retain the discrimination present in the patterns.

4.2.4 Singular Value Decomposition (SVD)

A more general factorization of $A_{n\times l}$ may be viewed $A_{n\times l} = B_{n\times n}D_{n\times l}C_{l\times l}$, where D is a diagonal matrix with $n - l$ zero rows if $n > l$ or with $l - n$ zero columns if $n < l$. In the earlier cases, where $A = BC$, D may be viewed as having in its diagonal portion the identity matrix I.

SVD may be viewed as

- orthonormal eigenvectors of the symmetric matrix AA^t as the columns of B.
- orthonormal eigenvetors of the symmetric matrix A^tA as the rows of C.
- Square roots of the eigenvalues of AA^t or A^tA, based on whether $n < l$ or $l < n$ respectively, as the diagonal entries of D with remaining elements to be 0. These diagonal entries are called the singular values of A.
- Importantly, SVD always gives $B,\ D,\ and\ C$ such that $A = BDC$, an exact deterministic factorization of any A matrix.

Consider the matrix A given in the example, we have

$$A^tA = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 2 \end{bmatrix}.$$

The eigenvalues of A^tA are $4,\ 2,\ and\ 0$ the respective eigenvectors are $(1, 0, 1)^t$, $(0, 1, 0)^t$, $(1, 0, -1)^t$. They are orthogonal and by normalizing them to be unit norm vectors, we get the C matrix as

$$C = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Similarly, the eigenvalues of AA^t are $4,\ 2,\ 0,\ 0$ and respective orthonormal eigenvectors that are used as columns of B give B as

$$B = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The $D_{4\times 3}$ is given

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where nonzero entries $\sqrt{4} = 2$, *and* $\sqrt{2}$ are the *singular values* that are the positive square roots of the nonzero eigenvalues of either AA^t or A^tA. Note that

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix} = BDC.$$

This is an exact factorization, which could be obtained for any $A_{m\times n}$. We can consider an approximation by retaining some largest singular values and ignoring (making them 0) the smaller singular values. For example, here if we ignore $\sqrt{2}$, that is approximate D to

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

then the resulting approximation to A based on the largest singular value is A_1 where

$$A_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Note that the squared Frobenius norm $||A - A_1||_F$ is 2 or the Frobenius norm is $\sqrt{2}$ which is the singular value that is ignored. It is not a coincidence. In general, if a matrix A is approximated to A_K by using the top K singular values in D, then $||A - A_K||_F = \sigma_{K+1}^2$ where σ_{K+1} is the largest of the ignored singular values. This helps in monitoring the possible error in approximating A to A_K for both dimensionality reduction and clustering. A popular application is in document representation, clustering, and classification under *latent semantic analysis (LSA)*.

Table 4.1 Optimization in clustering and dimensionality reduction

Specific task	Criterion function	Solution	Regularizer (Domain knowledge)
PCA	Minimize $E[(x^l - x^K)^t(x^l - x^K)$	Eigenvectors Covar. matrix	Best K (Domain)
KMA	Minimize squared error	Local minimum	Diverse Centers
Hierarchical clustering	Minimum spanning Tree	Dendrogram of clusters	Dendrogram cut appropriately
MI based Feat. Selection *MI*	Maximize features	K Best	Consider all classes
SVD	$A = BDC$	Exact	Approx. A_K

Under the matrix factorization, one can characterize any linear feature extraction including feature selection, hard and soft clustering, and even classification. Note that even nonlinear problems may be viewed as linear in an appropriate high-dimensional space. So, linear algebra in general and matrix factorization in particular are important in several of these topics.

Even the probabilistic variants like *probabilistic latent semantic analysis (PLSA)* are shown to be equivalent to deterministic factorization approaches like *NMF* and the *KMA*. This happens because both the approaches depend on some empirical schemes, based on the given data set in practice. In a more general sense *statistics* is responsible for the equivalence. An important semantic underlying matrix factorization is some kind of criterion function that is optimized with additional constraints to *regularize or reduce the diversity* of the solution space. We summarize the optimization related properties associated with clustering and dimensionality reduction in Table 4.1.

4.2.5 Diversified Clustering

Conventionally in clustering, the points in each cluster are similar to each other and points in different clusters are dissimilar. However, there are applications where each cluster needs to have diverse elements and a pair of clusters are highly similar. In other words there is a higher within cluster entropy and lower between cluster entropy.

Some of these applications are in

- *Peer Learning*: If a collection of students, selected based on some qualifying score, are to be grouped then the conventional clustering will lead to *stratified grouping*. In such a grouping all the students similar in terms of the qualifying score will be put together. This reduces the chance for peer learning. It can be shown to be good if each group has diverse students, that is students with varying qualifying

scores. Further, to avoid discrimination between groups different groups should have similar collective behavior. This means round-robin allotment students to groups is a better deal than stratified grouping.

- *Team formation*: When different soccer teams are to be selected to participate in a cup, there will be diversity in terms of special roles of players like the goalkeeper, wing, center forward, full back, etc., This means there will be diversity in terms of these special roles in each team. Further, every team requires a goalkeeper, two wings, etc., which means a pair of teams are structurally similar. Not only in sports, this kind of grouping is required in the formation of committees and many other team formation scenarios.
- *Groups based on a Standard*: UG programmes offered by various computer science departments typically follow ACM curriculum. So, the similarity between different UG programmes exists because of the standard like the ACM curriculum. At the same time, each programme needs to show enough diversity in terms of representation of theoretical CS, computer systems, and other topics like ML, AI, DBMS, graphics, etc. There other standards like, for example, the Dewey Decimal Classification, Library Congress classification, etc. which are followed by libraries across the globe.

4.3 Classification

We have seen in earlier chapters how search and representation impact the classifiers. Knowledge is used in the form of prior densities, selection of representation schemes for patterns and classes. We can search for how knowledge can be exploited in modeling, selecting the correct model, and even selection of the values of the hyperparameters. Search takes different forms including searching for a solution to an optimization problem based on some constraints. In this section, we will examine how optimization can be used in modeling and selecting classifier models.

A good number of classifiers are explicitly modeled or can be interpreted as solutions to some *intuitively appealing and convenient* optimization problems. We will look into some of the classifiers.

4.3.1 Perceptron

It may be viewed as minimizing the sum of the violations of the training patterns, their distances from the wrong side of the decision boundary, using the current w, the weight vector of perceptron. This happens because w has misclassified some training patterns. Noting that each such pattern, x satisfies $w^t x < 0$, the perceptron criterion function based on w is, *PCF(w)* is

$$PCF(w) = -\sum_{x:w^t x<0} w^t x.$$

$w^t x$ captures the extent of violation of x because of w. Because $w^t x < 0$ for such an x, we minimize $-w^t x$ for every x that is misclassified by w so that sum of the extent of violations is minimized.

If we consider the gradient $\nabla_w PCF(w)$, we get $-\sum_{x:w^t x<0} x$. So, if we use the gradient descent method to minimize $PCF(w)$, then the updates to w are given, using the negative of the gradient with a suitable scaling factor η, by

$$w_{k+1} = w_k + \eta \sum_{x:w_k^t x<0} x \tag{4.1}$$

This update rule is called *batch mode update*. There are several simplifications to this equation.

1. One variant is to use $\eta = 1$ and consider one x that is misclassified at a time rather than the sum of all the patterns x that are misclassified by w_k. This is popularly called the *fixed increment rule* that we discussed in the previous chapter.
2. Another variant is to insist that the w obtained is a simple sparse vector, minimum possible nonzero entries, that can be effectively used for classification which is useful in high-dimensional spaces. This is specified by

$$PCF(w) = -\sum_{x:w^t x<0} w^t x + \lambda' w^t w, \tag{4.2}$$

so that while minimizing the sum of violations, we reduce the nonzero entries in w as well. There is a scaling factor λ'. Noting that the gradient of $w^t w$ is $2w$, we have the the corresponding *incremental update rule*, one pattern at a time, to be

$$w_{k+1} = w_k + \eta x^k - \lambda w_k \rightarrow w_{k+1} = (1-\lambda) w_k + \eta x^k$$

where $\lambda = 2\eta\lambda'$ and x^k is the first pattern misclassified by w_k.

Note that both these variants are constraining or regularizing the optimization solution, w.

4.3.2 Support Vector Machine (SVM)

In SVMs, the criterion function that is considered is *margin between the two classes*. This may be detailed using Fig. 4.1. In *SVM* margin between the positive and negative classes is maximized. In the figure, there are negative class patterns in the left side. These are labeled by using $-$. Similarly, on the right side we have the positive class patterns. These are labeled by $+$.

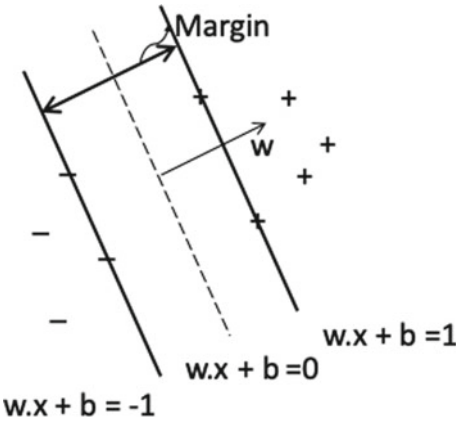

Fig. 4.1 Margin between the two classes

In this two-dimensional case, there are two parallel lines (in higher dimensions they will be parallel hyperplanes) called *support lines*. The respective class *boundary patterns* are located on these support lines. The negative class patterns satisfy the property that $w^t x + b \leq 1$ where the boundary vectors, xs, or *support vectors* (SVs) of the negative class satisfy $w^t x + b = -1$. Similarly, the positive class patterns satisfy $w^t x + b \geq 1$ with the respective SVs satisfying the property $w^t x + b = +1$.

The decision boundary between the two classes is characterized by points x such that $w^t x + b = 0$. Points to its right are from positive class and left side patterns are of negative class. If two points x_1 and x_2 are points on the decision boundary, then

$$w^t x_1 + b = w^t x_2 + b = 0 \Rightarrow w^t (x_1 - x_2) = 0.$$

This means vector w is orthogonal to $x_1 - x_2$ or the line on which they are located which is the decision boundary itself. So, w is orthogonal to the decision boundary as shown in the figure.

Another property is that w points towards the positive side. Consider a problem where the origin is on the decision boundary. So, $w^t 0 + b = 0 \Rightarrow b = 0$. Now if we consider a point $x_1 \in c_+$ the positive class, then $w^t x_1 > 0$. The cosine of the angle, θ, between w and x_1 is given by

$$cos\theta = \frac{w^t x_1}{||w||||x_1||}.$$

The denominator terms are positive here and the numerator is positive as $x_1 \in c_+$. So, $cos\theta > 0 \Rightarrow$ the angle between w and x_1 is acute. So, w points towards the positive side.

Any point $x \in c_+$ may be written as $x = x_d + p\frac{w}{||w||}$ where x_d is point on the decision boundary at which the normal projection of x onto the decision boundary meets it. If the distance between x and x_d is p units, then the corresponding vector is

$p\frac{w}{||w||}$ because w is orthogonal or normal to the decision boundary. But as $x \in c_+$,

$$w^t x + b = w^t(x_d + p\frac{w}{||w||}) + b = w^t x_d + b + p||w|| = p||w|| > 0$$

as $w^t x_d + b = 0$, where $\frac{w}{||w||}$ is a unit vector in the direction of w. So,

$$w^t x + b = p||w|| \Rightarrow p = \frac{w^t x + b}{||w||}.$$

So, normal distance between any point x on the positive support line and the decision boundary is $\frac{w^t x+b}{||w||} = \frac{1}{||w||}$. Similarly, from any point x on the negative support line to the decision boundary the modulus of the distance is again $\frac{1}{||w||}$. So,

$$margin = \frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||}.$$

In SVM, we find w that *maximizes the margin*. Equivalently, we minimize $\frac{1}{2}||w||^2$ which maximizes the margin. The constraints are $y_i(w^t x_i + b) \geq 1$ where y_i is the class label of x_i; $y_i = 1$ *or* -1 based on whether $x_i \in c_+$ or $x_i \in c_-$ respectively.

We can express the corresponding Lagrangian by taking into account the constraints as

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 + \sum_{i=1}^{n} \alpha_i(1 - y_i(w^t x_i + b)),$$

where we would like to find the vectors $w, \alpha = \{\alpha_1, \ldots, \alpha_n\}$ and the scalar b. Optimal values of these variables can be obtained by equating the gradient to zero which is given by

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i.$$

$$\nabla_b \mathcal{L} = \sum_{i=1} \alpha_i y_i = 0.$$

$$\nabla_{\alpha_i} \mathcal{L} = 1 - y_i(w^t x_i + b) = 0 \Rightarrow y_i(w^t x_i + b) = 1 \Rightarrow w^t x_i + b = y_i$$

There are other conditions also including $\alpha_i \geq 0$ and $\alpha_i(1 - y_i(w^t x_i + b)) = 0$.

By using these equations all the variables w, α, and b can be determined using different approaches. Here, the optimization problem was chosen such that it is a well-behaved problem guaranteeing a globally optimal solution to the minimization. However, we face a difficulty when there is no margin between the two classes. This can happen, for example in Fig. 4.1, if a point from the positive class falls to the left of the decision boundary or equivalently a point from the negative class falls to the

right of the decision boundary. Such points are called *violators*. This can be the case in most of the real-world problems.

To overcome this problem, a popularly used solution is to formulate it as a *soft margin problem*. This is achieved by weighing each of the violators using a weight C based on the extent of violation. If we do not want to permit any violator then $C \rightarrow \infty$. This amounts to the soft margin formulation to converge to the hard margin formulation. On the other extreme, a value of $C = 0$ means every point can be a violator. However, this will not solve the problem in practice.

Typically, a positive finite nonzero value is used for C to accommodate some violators. The corresponding problem is

$$\min_{w} \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i$$

$s.t.\ y_i(w^t x_i + b) \geq 1 - \xi_i$. and $\xi_i \geq 0$. It is seen that there is no change in the form of the variables. Only change is that in the hard margin formulation, $\alpha_i \geq 0$. In the soft margin case, $0 \leq \alpha_i \leq C$.

An important practical consideration is the right value of C. This shifts the attention from getting the global optimal solution to getting the right value of C. So, tuning of hyperparameter C occupies the central stage in practice. Some professionally developed software packages have helped in realizing this practically.

4.3.3 Summary

In this chapter, we have seen the role of optimization in dimensionality reduction, clustering, and classification. We have considered only some of the algorithms. There are potentially a large variety of other machine learning platforms like neural networks. In a sense optimization based solutions exhibit diversity which is controlled using regularization to provide more central or less variance solutions.

Note the following about optimization. The set of constraints specify the *feasible region*. This may typically characterize potentially infinite solutions or diverse possible solutions. The criterion function being optimized will force the selection of one or more of these diverse points in the solution space increasing the centrality. A regularizer will shrink this collection of possible solutions further.

Consider, for example, a data set of the following four patterns drawn from 2 classes as shown in Table 4.2. Let patterns 1 and 2 be training points from class 1 and let the other 2, that is patterns 3 and 4 be from class 2.

Let a classifier gave two w vectors given by
$w^1 = (1, 0, 1, 0, -2)$
$w^2 = (0.5, 0.6, 0.5, 0.4, -2)$. Verify that both these weight vectors classify all the four patterns correctly. For example, using w^1 on pattern 2 gives us

Table 4.2 4-dimensional data from two classes

Pattern	x_1	x_2	x_3	x_4
1	0.6	0.4	0.7	0.4
2	0.5	0.3	0.7	0.5
3	1.2	1.4	1.5	1.6
4	1.3	1.3	1.4	1.5

$0.5 + 0 + 0.7 + 0 - 2 < 0$. Similarly w^2 with pattern 4 gives us $0.65 + 0.78 + 0.7 + 0.9 - 2 > 0$. Similarly one can verify other patterns.

Among these two weight vectors, if we require a sparse vector, then w^1 will be selected and w^2 will be left out.

Bibliography

1. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press
2. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J (2008) LIBLINEAR: a library for large linear classification. JMLR 9:1871–1874
3. Hsu CW, Lin C-J (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
4. Lay DC (2012) Linear algebra and its applications. Addison-Wesley
5. Murty MN, Devi VS (2015) Introduction to Pattern recognition and machine learning. IISc Press
6. Ding C, Li T, Peng W (2008) On the equivalence between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing. Comput Stat Data Anal 52:3913–3917
7. Chaudhuri AR, Murty MN (2012) On the relation between K-means and PLSA. In: Proceedings of ICPR, Nov 11–15 2012: pp 2298–2301, Japan
8. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley-Interscience
9. Sambaran B, Sharad N, Rishabh D, Murty MN (2018) DivGroup: a diversified approach to divide collection of patterns into uniform groups. ICPR 2018:964–969
10. Rakesh A, Sharad N, Murty MN (2017) Grouping students for maximizing learning from peers. EDM