3 Best-Fit Subspaces and Singular Value Decomposition (SVD)

Think of the rows of an $n \times d$ matrix A as n data points in a d-dimensional space and consider the problem of finding the best k-dimensional subspace with respect to the set of points. Here best means minimize the sum of the squares of the perpendicular distances of the points to the subspace. We begin with a special case where the subspace is 1-dimensional, namely a line through the origin. The best fitting k-dimensional subspace is found by repeated applications of the best fitting line algorithm, each time finding the best fitting line perpendicular to the subspace found so far. When k reaches the rank of the matrix, a decomposition of the matrix, called the *Singular Value Decomposition (SVD)*, is obtained from the best fitting lines.

The singular value decomposition of a matrix A is the factorization of A into the product of three matrices, $A = UDV^T$, where the columns of U and V are orthonormal and the matrix D is diagonal with positive real entries. In many applications, a data matrix A is close to a low rank matrix and a low rank approximation to A is desired. The singular value decomposition of A gives the best rank k approximation to A, for any k .

The singular value decomposition is defined for all matrices, whereas the more commonly used eigenvector decomposition requires the matrix A be square and certain other conditions on the matrix to ensure orthogonality of the eigenvectors. In contrast, the columns of V in the singular value decomposition, called the *right-singular vectors* of A, always form an orthonormal set with no assumptions on A. The columns of U are called the *left-singular vectors* and they also form an orthonormal set. A simple consequence of the orthonormality is that for a square and invertible matrix A, the inverse of A is $VD^{-1}U^T$.

Project a point $\mathbf{a_i} = (a_{i1}, a_{i2}, \ldots, a_{id})$ onto a line through the origin. Then

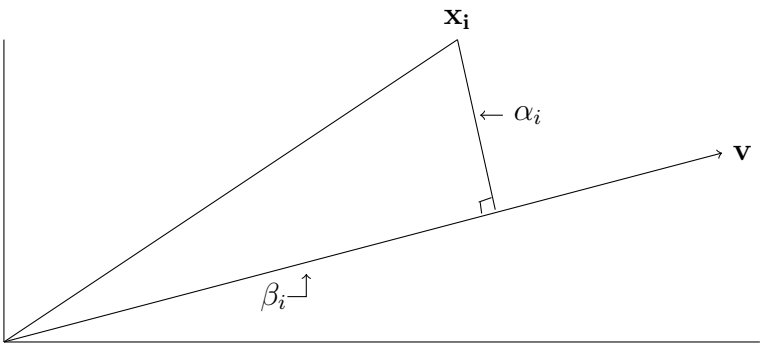
$$a_{i1}^2 + a_{i2}^2 + \cdots + a_{id}^2 = (\text{length of projection})^2 + (\text{distance of point to line})^2 .$$

See Figure 3.1. Thus

$$(\text{distance of point to line})^2 = a_{i1}^2 + a_{i2}^2 + \cdots + a_{id}^2 - (\text{length of projection})^2 .$$

Since $\sum_{i=1}^{n} (a_{i1}^2 + a_{i2}^2 + \cdots + a_{id}^2)$ is a constant independent of the line, minimizing the sum of the squares of the distances to the line is equivalent to maximizing the sum of the squares of the lengths of the projections onto the line. Similarly for best-fit subspaces, maximizing the sum of the squared lengths of the projections onto the subspace minimizes the sum of squared distances to the subspace.

Thus, there are two interpretations of the best-fit subspace. The first is that it minimizes the sum of squared distances of the data points to it. This interpretation and its



Minimizing $\sum_i \alpha_i^2$ is equivalent to maximizing $\sum_i \beta_i^2$

Figure 3.1: The projection of the point $\mathbf{x_i}$ onto the line through the origin in the direction of $\mathbf{v}$.

use are akin to the notion of least-squares fit from calculus. But there is a difference. Here the perpendicular distance to the line or subspace is minimized, whereas, in the calculus notion, given n pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, one finds a line $l = \{(x, y)|y = mx + b\}$ minimizing the vertical distance of the points to it, namely, $\sum_{i=1}^{n}(y_i - mx_i - b)^2$.

The second interpretation of best-fit-subspace is that it maximizes the sum of projections squared of the data points on it. In some sense the subspace contains the maximum content of data among all subspaces of the same dimension.

The reader may wonder why we minimize the sum of squared distances to the line. We could alternatively have defined the best-fit line to be the one that minimizes the sum of distances to the line. There are examples where this definition gives a different answer than the line minimizing the sum of squared distances. The choice of the objective function as the sum of squared distances seems arbitrary, but the square has many nice mathematical properties. The first of these is the use of Pythagoras theorem to say that minimizing the sum of squared distances is equivalent to maximizing the sum of squared projections.

3.1 Singular Vectors

Consider the best fit line through the origin for the points determined by the rows of A. Let $\mathbf{v}$ be a unit vector along this line. The length of the projection of $\mathbf{a_i}$, the i^{th} row of A, onto $\mathbf{v}$ is $|\mathbf{a_i} \cdot \mathbf{v}|$ and the sum of length squared of the projections is $|A\mathbf{v}|^2$. The best fit line is the one maximizing $|A\mathbf{v}|^2$ and hence minimizing the sum of the squared distances of the points to the line.

With this in mind, define the *first singular vector*, $\mathbf{v_1}$, of A, which is a column vector, as the vector defining the best fit line through the origin for the n points in d-space that

are the rows of A. Thus

$$\mathbf{v_1} = \arg\max_{|\mathbf{v}|=1} |A\mathbf{v}|.$$

There may be a tie for the vector attaining the maximum and so technically we should not use the article "the". If there is a tie, arbitrarily pick one of the vectors and refer to it as "the first singular vector" avoiding the more cumbersome "one of the the vectors achieving the maximum". We adopt this terminology for all uses of arg max .

The value $\sigma_1(A) = |A\mathbf{v_1}|$ is called the *first singular value* of A. Note that $\sigma_1^2 = \sum_{i=1}^{n} (\mathbf{a_i} \cdot \mathbf{v_1})^2$ is the sum of the squares of the projections of the points to the line determined by $\mathbf{v_1}$.

If the data points were all either on a line or close to a line, $\mathbf{v_1}$ would give the direction of that line. It is possible that data points are not close to one line, but lie close to a 2-dimensional plane or more generally a low dimensional affine space. A widely applied technique called Principal Component Analysis (PCA) indeed deals with such situations using singular vectors. How do we find the best-fit 2-dimensional plane or more generally the k-dimensional affine space?

The greedy approach to find the best fit 2-dimensional subspace for a matrix A, takes $\mathbf{v_1}$ as the first basis vector for the 2-dimensional subspace and finds the best 2-dimensional subspace containing $\mathbf{v_1}$. The fact that we are using the sum of squared distances helps. For every 2-dimensional subspace containing $\mathbf{v_1}$, the sum of squared lengths of the projections onto the subspace equals the sum of squared projections onto $\mathbf{v_1}$ plus the sum of squared projections along a vector perpendicular to $\mathbf{v_1}$ in the subspace. Thus, instead of looking for the best 2-dimensional subspace containing $\mathbf{v_1}$, look for a unit vector $\mathbf{v_2}$ perpendicular to $\mathbf{v_1}$ that maximizes $|A\mathbf{v}|^2$ among all such unit vectors. Using the same greedy strategy to find the best three and higher dimensional subspaces, define $\mathbf{v_3}, \mathbf{v_4}, \ldots$ in a similar manner. This is captured in the following definitions.

The *second singular vector*, $\mathbf{v_2}$, is defined by the best fit line perpendicular to $\mathbf{v_1}$.

$$\mathbf{v_2} = \arg\max_{\substack{\mathbf{v} \perp \mathbf{v_1} \\ |\mathbf{v}|=1}} |A\mathbf{v}|$$

The value $\sigma_2(A) = |A\mathbf{v_2}|$ is called the *second singular value* of A**.** The *third singular vector* $\mathbf{v}_3$ and *third singular value* are defined similarly by

$$\mathbf{v_3} = \arg\max_{\substack{\mathbf{v} \perp \mathbf{v_1}, \mathbf{v_2} \\ |\mathbf{v}|=1}} |A\mathbf{v}|$$

and

$$\sigma_3(A) = |A\mathbf{v_3}|,$$

and so on. The process stops when we have found singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$, singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$, and

$$\max_{\substack{\mathbf{v} \perp \mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r} \\ |\mathbf{v}|=1}} |A\mathbf{v}| = 0.$$

There is no apriori guarantee that the greedy algorithm gives the best fit. But, in fact, the greedy algorithm does work and yields the best-fit subspaces of every dimension as we will show. If instead of finding the $\mathbf{v_1}$ that maximized $|A\mathbf{v}|$ and then the best fit 2-dimensional subspace containing $\mathbf{v_1}$, we had found the best fit 2-dimensional subspace, we might have done better. This is not the case. We give a simple proof that the greedy algorithm indeed finds the best subspaces of every dimension.

Theorem 3.1 *Let A be an $n \times d$ matrix with singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$. For $1 \leq k \leq r$, let V_k be the subspace spanned by $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$. For each k, V_k is the best-fit k-dimensional subspace for A.*

Proof: The statement is obviously true for $k = 1$. For $k = 2$, let W be a best-fit 2-dimensional subspace for A. For any orthonormal basis $(\mathbf{w_1}, \mathbf{w_2})$ of W, $|A\mathbf{w_1}|^2 + |A\mathbf{w_2}|^2$ is the sum of squared lengths of the projections of the rows of A onto W. Choose an orthonormal basis $(\mathbf{w}_1, \mathbf{w_2})$ of W so that $\mathbf{w_2}$ is perpendicular to $\mathbf{v_1}$. If $\mathbf{v_1}$ is perpendicular to W, any unit vector in W will do as $\mathbf{w_2}$. If not, choose $\mathbf{w_2}$ to be the unit vector in W perpendicular to the projection of $\mathbf{v_1}$ onto W. This makes $\mathbf{w_2}$ perpendicular to $\mathbf{v_1}$. Since $\mathbf{v_1}$ maximizes $|A\mathbf{v}|^2$, it follows that $|A\mathbf{w_1}|^2 \leq |A\mathbf{v_1}|^2$. Since $\mathbf{v_2}$ maximizes $|A\mathbf{v}|^2$ over all $\mathbf{v}$ perpendicular to $\mathbf{v_1}$, $|A\mathbf{w_2}|^2 \leq |A\mathbf{v_2}|^2$. Thus

$$|A\mathbf{w_1}|^2 + |A\mathbf{w_2}|^2 \leq |A\mathbf{v_1}|^2 + |A\mathbf{v_2}|^2.$$

Hence, V_2 is at least as good as W and so is a best-fit 2-dimensional subspace.

For general k, proceed by induction. By the induction hypothesis, V_{k-1} is a best-fit k-1 dimensional subspace. Suppose W is a best-fit k-dimensional subspace. Choose an orthonormal basis $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_k}$ of W so that $\mathbf{w_k}$ is perpendicular to $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{k-1}}$. Then

$$|A\mathbf{w_1}|^2 + |A\mathbf{w_2}|^2 + \cdots + |A\mathbf{w_k}|^2 \leq |A\mathbf{v_1}|^2 + |A\mathbf{v_2}|^2 + \cdots + |A\mathbf{v_{k-1}}|^2 + |A\mathbf{w_k}|^2$$

since V_{k-1} is an optimal k -1 dimensional subspace. Since $\mathbf{w_k}$ is perpendicular to $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{k-1}}$, by the definition of $\mathbf{v_k}$, $|A\mathbf{w_k}|^2 \leq |A\mathbf{v_k}|^2$. Thus

$$|A\mathbf{w_1}|^2 + |A\mathbf{w_2}|^2 + \cdots + |A\mathbf{w_{k-1}}|^2 + |A\mathbf{w_k}|^2 \leq |A\mathbf{v_1}|^2 + |A\mathbf{v_2}|^2 + \cdots + |A\mathbf{v_{k-1}}|^2 + |A\mathbf{v_k}|^2,$$

proving that V_k is at least as good as W and hence is optimal. ∎

Note that the n-vector $A\mathbf{v_i}$ is a list of lengths with signs of the projections of the rows of A onto $\mathbf{v_i}$. Think of $|A\mathbf{v_i}| = \sigma_i(A)$ as the "component" of the matrix A along $\mathbf{v_i}$. For this interpretation to make sense, it should be true that adding up the squares of the

components of A along each of the $\mathbf{v_i}$ gives the square of the "whole content of the matrix A". This is indeed the case and is the matrix analogy of decomposing a vector into its components along orthogonal directions.

Consider one row, say $\mathbf{a_j}$, of A. Since $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$ span the space of all rows of A, $\mathbf{a_j} \cdot \mathbf{v} = 0$ for all $\mathbf{v}$ perpendicular to $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$. Thus, for each row $\mathbf{a_j}$, $\sum_{i=1}^{r} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = |\mathbf{a_j}|^2$. Summing over all rows j,

$$\sum_{j=1}^{n} |\mathbf{a_j}|^2 = \sum_{j=1}^{n} \sum_{i=1}^{r} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = \sum_{i=1}^{r} \sum_{j=1}^{n} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = \sum_{i=1}^{r} |A\mathbf{v_i}|^2 = \sum_{i=1}^{r} \sigma_i^2(A).$$

But $\sum_{j=1}^{n} |\mathbf{a_j}|^2 = \sum_{j=1}^{n} \sum_{k=1}^{d} a_{jk}^2$, the sum of squares of all the entries of A. Thus, the sum of squares of the singular values of A is indeed the square of the "whole content of A", i.e., the sum of squares of all the entries.

There is an important norm associated with this quantity, the Frobenius norm of A, denoted $||A||_F$ defined as

$$||A||_F = \sqrt{\sum_{j,k} a_{jk}^2}.$$

Lemma 3.2 *For any matrix A, the sum of squares of the singular values equals the square of the Frobenius norm. That is, $\sum \sigma_i^2(A) = ||A||_F^2$.*

Proof: By the preceding discussion. ∎

The vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$ are called the *right-singular vectors*. The vectors $A\mathbf{v_i}$ form a fundamental set of vectors and we normalize them to length one by

$$\mathbf{u_i} = \frac{1}{\sigma_i(A)} A\mathbf{v_i}.$$

The vectors, $\mathbf{u_2}, \ldots, \mathbf{u_r}$ are called the *left-singular vectors*. Later we will show that they are orthogonal and u_i maximizes $|\mathbf{u}^T A|$ over all unit length u perpendicular to allu_j, $j < i$.

3.2 Singular Value Decomposition (SVD)

Let A be an $n \times d$ matrix with singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$ and corresponding singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$. The left-singular vectors of A are $\mathbf{u_i} = \frac{1}{\sigma_i} A\mathbf{v_i}$ where $\sigma_i \mathbf{u_i}$ is a vector whose coordinates correspond to the projections of the rows of A onto $\mathbf{v_i}$. Each $\sigma_i \mathbf{u_i} \mathbf{v_i^T}$ is a rank one matrix whose columns are weighted versions of $\sigma_i \mathbf{u_i}$, weighted proportional to the coordinates of $\mathbf{v_i}$.

We will prove that A can be decomposed into a sum of rank one matrices as

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T.$$

We first prove a simple lemma stating that two matrices A and B are identical if $A\mathbf{v} = B\mathbf{v}$ for all $\mathbf{v}$.

Lemma 3.3 *Matrices A and B are identical if and only if for all vectors $\mathbf{v}$, $A\mathbf{v} = B\mathbf{v}$.*

Proof: Clearly, if $A = B$ then $A\mathbf{v} = B\mathbf{v}$ for all $\mathbf{v}$. For the converse, suppose that $A\mathbf{v} = B\mathbf{v}$ for all $\mathbf{v}$. Let $\mathbf{e_i}$ be the vector that is all zeros except for the i^{th} component which has value one. Now $A\mathbf{e_i}$ is the i^{th} column of A and thus $A = B$ if for each i, $A\mathbf{e_i} = B\mathbf{e_i}$. ∎

Theorem 3.4 *Let A be an $n \times d$ matrix with right-singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$, left-singular vectors $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_r}$, and corresponding singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$. Then*

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T.$$

Proof: We first show that multiplying both A and $\sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$ by $\mathbf{v_j}$ results in quantity Av_j.

$$\left(\sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T \right) \mathbf{v_j} = \sigma_j \mathbf{u_j} = A\mathbf{v_j}$$

Since any vector $\mathbf{v}$ can be expressed as a linear combination of the singular vectors plus a vector perpendicular to the $\mathbf{v_i}$, $A\mathbf{v} = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T \mathbf{v}$ for all $\mathbf{v}$ and by Lemma 3.3, $A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$. ∎

The decomposition $A = \sum_i \sigma_i \mathbf{u_i} \mathbf{v_i^T}$ is called the *singular value decomposition*, *SVD*, of A. In matrix notation $A = UDV^T$ where the columns of U and V consist of the left and right-singular vectors, respectively, and D is a diagonal matrix whose diagonal entries are the singular values of A. To see that $A = UDV^T$, observe that each $\sigma_i \mathbf{u_i} \mathbf{v_i^T}$ is a rank one matrix and $A = \sum_i \sigma_i \mathbf{u_i} \mathbf{v_i^T}$ is a sum of rank one matrices. Each $\sigma_i \mathbf{u_i} \mathbf{v_i^T}$, term contributes $\sigma_i u_{ji} v_{ik}$ to the jk^{th} element of A. Thus, $a_{jk} = \sum_i \sigma_i u_{ji} u_{ik}$ which is correct.

For any matrix A, the sequence of singular values is unique and if the singular values are all distinct, then the sequence of singular vectors is unique also. When some set of singular values are equal, the corresponding singular vectors span some subspace. Any set of orthonormal vectors spanning this subspace can be used as the singular vectors.

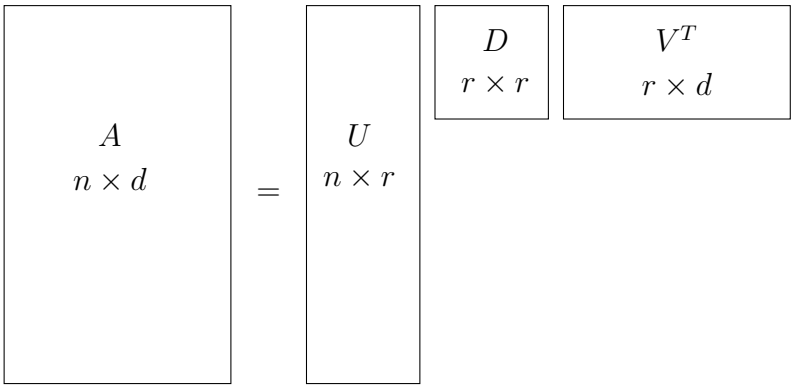


Figure 3.2: The SVD decomposition of an $n \times d$ matrix.

3.3 Best Rank k Approximations

Let A be an $n \times d$ matrix and let

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$$

be the SVD of A. For $k \in \{1, 2, \ldots, r\}$, let

$$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$$

be the sum truncated after k terms. It is clear that A_k has rank k. It is also the case that A_k is the best rank k approximation to A, where error is measured in Frobenius norm.

To show that A_k is the best rank k approximation to A when error is measured by the Frobenius norm, we first show that the rows of $A - A_k$ are the projections of the rows of A onto the subspace V_k spanned by the first k singular vectors of A. This implies that $||A - A_k||_F^2$ equals the sum of squared distances of the rows of A to the subspace V_k.

Lemma 3.5 *Let V_k be the subspace spanned by the first k singular vectors of A. The rows of A_k are the projections of the rows of A onto the subspace V_k.*

Proof: Let $\mathbf{a}$ be an arbitrary row vector. Since the $\mathbf{v_i}$ are orthonormal, the projection of the vector $\mathbf{a}$ onto V_k is given by $\sum_{i=1}^{k} (\mathbf{a} \cdot \mathbf{v_i}) \mathbf{v_i}^T$. Thus, the matrix whose rows are the projections of the rows of A onto V_k is given by $\sum_{i=1}^{k} A \mathbf{v_i} \mathbf{v_i}^T$. This last expression simplifies to

$$\sum_{i=1}^{k} A \mathbf{v_i} \mathbf{v_i}^T = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^T = A_k.$$

Thus, the rows of A_k are the projections of the rows of A onto the subspace V_k. ∎

We next show that if B is a rank k matrix minimizing $||A - B||_F^2$ among all rank k matrices, that each row of B must be the projection of the corresponding row of A onto the space spanned by rows of B. This implies that $||A - B||_F^2$ is the sum of squared distances of rows of A to the space spanned by the rows of B. Since the space spanned by the rows of B is a k dimensional subspace and since the subspace spanned by the first k singular vectors minimizes the sum of squared distances over all k-dimensional subspaces, it must be that $\|A - A_k\|_F \leq \|A - B\|_F$.

Theorem 3.6 *For any matrix B of rank at most k*

$$\|A - A_k\|_F \leq \|A - B\|_F$$

Proof: Let B minimize $\|A - B\|_F^2$ among all rank k or less matrices. Let V be the space spanned by the rows of B. The dimension of V is at most k. Since B minimizes $\|A - B\|_F^2$, it must be that each row of B is the projection of the corresponding row of A onto V, otherwise replacing the row of B with the projection of the corresponding row of A onto V does not change V and hence the rank of B but would reduce $\|A - B\|_F^2$. Since now each row of B is the projection of the corresponding row of A, it follows that $\|A - B\|_F^2$ is the sum of squared distances of rows of A to V. By Theorem 3.1, A_k minimizes the sum of squared distance of rows of A to any k-dimensional subspace. It follows that $\|A - A_k\|_F \leq \|A - B\|_F$. ∎

There is another matrix norm, called the *2-norm*, that is of interest. To motivate, consider the example of a document-term matrix A. Suppose we have a large database of documents which form the rows of an $n \times d$ matrix A. There are d terms and each document is a d-vector with one component per term which is the number of occurrences of the term in the document. We are allowed to "preprocess" A. After the preprocessing, we receive queries. Each query $\mathbf{x}$ is an d-vector specifying how important each term is to the query. The desired answer is a n-vector which gives the similarity (dot product) of the query to each document in the database, namely, the "matrix-vector" product, $A\mathbf{x}$. Query time should be much less than processing time, one answers many queries for the data base. Näively, it would take $O(nd)$ time to do the product $A\mathbf{x}$. However, if we approximate A by $A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$ we could return $A_k\mathbf{x} = \sum_{i=1}^{k} \sigma_i \mathbf{u_i}(\mathbf{v_i} \cdot \mathbf{x})$ as the approximation to $A\mathbf{x}$. This only takes k dot products of d-vectors and takes time $O(kd)$ which is a win provided k is fairly small. How do we measure the error? Since $\mathbf{x}$ is unknown, the approximation needs to be good for every $\mathbf{x}$. So we should take the maximum over all $\mathbf{x}$ of $|(A_k - A)\mathbf{x}|$. But unfortunately, this is infinite since $|\mathbf{x}|$ can grow without bound. So we restrict to $|\mathbf{x}| \leq 1$.

The *2-norm* or *spectral norm* of a matrix A is

$$||A||_2 = \max_{|\mathbf{x}| \leq 1} |A\mathbf{x}|.$$

Note that the 2-norm of A equals $\sigma_1(A)$.

We will prove in Section 3.4 that A_k is the best rank k, 2-norm approximation to A.

3.4 Left Singular Vectors

In this section we show that he left singular vectors are orthogonal and that A_k is the best 2-norm approximation to A.

Theorem 3.7 *The left singular vectors are pairwise orthogonal.*

Proof: First we show that each $\mathbf{u_i}, i \geq 2$ is orthogonal to $\mathbf{u_1}$. Suppose not, and for some $i \geq 2$, $\mathbf{u_1}^T\mathbf{u_i} \neq 0$. Without loss of generality assume that $\mathbf{u_1}^T\mathbf{u_i} > 0$. The proof is symmetric for the case where $\mathbf{u_1}^T\mathbf{u_i} < 0$. Now, for infinitesimally small $\varepsilon > 0$, the vector

$$A\left(\frac{\mathbf{v}_1 + \varepsilon\mathbf{v_i}}{|\mathbf{v_1} + \varepsilon\mathbf{v_i}|}\right) = \frac{\sigma_1\mathbf{u_1} + \varepsilon\sigma_i\mathbf{u_i}}{\sqrt{1+\varepsilon^2}}$$

has length at least as large as its component along $\mathbf{u_1}$ which is

$$\mathbf{u_1^T}\left(\frac{\sigma_1\mathbf{u_1} + \varepsilon\sigma_i\mathbf{u_i}}{\sqrt{1+\varepsilon^2}}\right) = \left(\sigma_1 + \varepsilon\sigma_i\mathbf{u_1}^T\mathbf{u_i}\right)\left(1 - \tfrac{\varepsilon^2}{2} + O\left(\varepsilon^4\right)\right) = \sigma_1 + \varepsilon\sigma_i\mathbf{u_1}^T\mathbf{u_i} - O\left(\varepsilon^2\right) > \sigma_1,$$

a contradiction. Thus $\mathbf{u_1} \cdot \mathbf{u_i} = 0$ for $i \geq 2$.

The proof for other $\mathbf{u_i}$ and $\mathbf{u_j}, j > i > 1$ is similar. Suppose without loss of generality that $\mathbf{u_i}^T\mathbf{u_j} > 0$.

$$A\left(\frac{\mathbf{v}_i + \varepsilon\mathbf{v_j}}{|\mathbf{v_i} + \varepsilon\mathbf{v_j}|}\right) = \frac{\sigma_i\mathbf{u_i} + \varepsilon\sigma_j\mathbf{u_j}}{\sqrt{1+\varepsilon^2}}$$

has length at least as large as its component along $\mathbf{u_i}$ which is

$$\mathbf{u_i^T}\left(\frac{\sigma_1\mathbf{u_i} + \varepsilon\sigma_j\mathbf{u_j}}{\sqrt{1+\varepsilon^2}}\right) = \left(\sigma_i + \varepsilon\sigma_j\mathbf{u_i}^T\mathbf{u_j}\right)\left(1 - \tfrac{\varepsilon^2}{2} + O\left(\varepsilon^4\right)\right) = \sigma_i + \varepsilon\sigma_j\mathbf{u_i}^T\mathbf{u_j} - O\left(\varepsilon^2\right) > \sigma_i,$$

a contradiction since $\mathbf{v_i} + \varepsilon\mathbf{v_j}$ is orthogonal to $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{i-1}}$ and σ_i is the maximum over such vectors of $|A\mathbf{v}|$. ∎

In Theorem 3.9 we show that $A - k$ is the best 2-norm approximation to A. We first show that the square of the 2-norm of $A - A_k$ is the square of the $(k+1)^{st}$ singular value of A,

Lemma 3.8 $\|A - A_k\|_2^2 = \sigma_{k+1}^2$.

Proof: Let $A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$ be the singular value decomposition of A. Then $A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$ and $A - A_k = \sum_{i=k+1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$. Let $\mathbf{v}$ be the top singular vector of $A - A_k$. Express $\mathbf{v}$ as a linear combination of $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$. That is, write $\mathbf{v} = \sum_{i=1}^{r} \alpha_i \mathbf{v_i}$. Then

$$|(A - A_k)\mathbf{v}| = \left| \sum_{i=k+1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T \sum_{j=1}^{r} \alpha_j \mathbf{v_j} \right| = \left| \sum_{i=k+1}^{r} \alpha_i \sigma_i \mathbf{u_i} \mathbf{v_i}^T \mathbf{v_i} \right|$$

$$= \left| \sum_{i=k+1}^{r} \alpha_i \sigma_i \mathbf{u_i} \right| = \sqrt{\sum_{i=k+1}^{r} \alpha_i^2 \sigma_i^2},$$

since the $\mathbf{u_i}$ are orthonormal. The $\mathbf{v}$ maximizing this last quantity, subject to the constraint that $|\mathbf{v}|^2 = \sum_{i=1}^{r} \alpha_i^2 = 1$, occurs when $\alpha_{k+1} = 1$ and the rest of the α_i are 0. Thus, $\|A - A_k\|_2^2 = \sigma_{k+1}^2$ proving the lemma. ∎

Finally, we prove that A_k is the best rank k, 2-norm approximation to A.

Theorem 3.9 *Let A be an $n \times d$ matrix. For any matrix B of rank at most k*

$$\|A - A_k\|_2 \leq \|A - B\|_2 .$$

Proof: If A is of rank k or less, the theorem is obviously true since $\|A - A_k\|_2 = 0$. Assume that A is of rank greater than k. By Lemma 3.8, $\|A - A_k\|_2^2 = \sigma_{k+1}^2$. The null space of B, the set of vectors $\mathbf{v}$ such that $B\mathbf{v} = 0$, has dimension at least $d - k$. Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{k+1}}$ be the first $k + 1$ singular vectors of A. By a dimension argument, it follows that there exists a $\mathbf{z} \neq 0$ in

$$\text{Null}\,(B) \cap \text{Span}\left\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{k+1}}\right\}.$$

Let $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{d-k}}$ be $d-k$ independent vectors in Null(B). Now, $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{d-k}}, \mathbf{v_1}$, $\mathbf{v_2}, \ldots, \mathbf{v_{k+1}}$ are $d+1$ vectors in d space and thus are linearly dependent. Let $\alpha_1, \alpha_2, \ldots, \alpha_{d-k}$ and $\beta_1, \beta_2, \ldots, \beta_k$ be such that $\sum_{i=1}^{d-k} \alpha_i \mathbf{u_i} = \sum_{j=1}^{k} \beta_j \mathbf{v_j}$. Let $\mathbf{z} = \sum_{i=1}^{d-k} \alpha_i \mathbf{u_i}$. Scale $\mathbf{z}$ so that $|\mathbf{z}| = 1$. We now show that for this vector $\mathbf{z}$, which lies in the space of the first $k+1$ singular vectors of A, that $(A - B)\,\mathbf{z} \geq \sigma_{k+1}$. Hence the 2-norm of $A - B$ is at least σ_{k+1}. First

$$\|A - B\|_2^2 \geq |(A - B)\,\mathbf{z}|^2 .$$

Since $B\mathbf{z} = 0$,

$$\|A - B\|_2^2 \geq |A\mathbf{z}|^2 .$$

Since $\mathbf{z}$ is in the Span $\left\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{k+1}}\right\}$

$$|A\mathbf{z}|^2 = \left| \sum_{i=1}^{n} \sigma_i \mathbf{u_i} \mathbf{v_i}^T \mathbf{z} \right|^2 = \sum_{i=1}^{n} \sigma_i^2 \left(\mathbf{v_i}^T \mathbf{z}\right)^2 = \sum_{i=1}^{k+1} \sigma_i^2 \left(\mathbf{v_i}^T \mathbf{z}\right)^2 \geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} \left(\mathbf{v_i}^T \mathbf{z}\right)^2 = \sigma_{k+1}^2.$$

It follows that $\|A - B\|_2^2 \geq \sigma_{k+1}^2$ and the theorem is proved. ∎

3.5 Power Method for Computing the Singular Value Decomposition

Computing the singular value decomposition is an important branch of numerical analysis in which there have been many sophisticated developments over a long period of time. Here we present an "in-principle" method to establish that the approximate SVD of a matrix A can be computed in polynomial time. The reader is referred to numerical analysis texts for more details. The method we present, called the *power method*, is simple and is in fact the conceptual starting point for many algorithms. Let A be a matrix whose SVD is $\sum_i \sigma_i \mathbf{u_i}\mathbf{v_i}^T$. We wish to work with a matrix that is square and symmetric. By direct multiplication, since $\mathbf{u_i}^T\mathbf{u_j}$ is the dot product of the two vectors and is zero unless $i = j$

$$\begin{aligned} B = A^T A &= \left(\sum_i \sigma_i \mathbf{v_i}\mathbf{u_i}^T\right)\left(\sum_j \sigma_j \mathbf{u_j}\mathbf{v_j}^T\right) \\ &= \sum_{i,j} \sigma_i\sigma_j \mathbf{v_i}(\mathbf{u}_i^T \cdot \mathbf{u_j})\mathbf{v_j}^T = \sum_i \sigma_i^2 \mathbf{v_i}\mathbf{v_i}^T. \end{aligned}$$

The matrix B is square and symmetric, and has the same left and right-singular vectors. If A is itself square and symmetric, it will have the same right and left-singular vectors, namely $A = \sum_i \sigma_i \mathbf{v_i}\mathbf{v_i}^T$ and computing B is unnecessary.

Now consider computing B^2.

$$B^2 = \left(\sum_i \sigma_i^2 \mathbf{v_i}\mathbf{v_i}^T\right)\left(\sum_j \sigma_j^2 \mathbf{v_j}\mathbf{v_j}^T\right) = \sum_{ij} \sigma_i^2\sigma_j^2 \mathbf{v_i}(\mathbf{v_i}^T\mathbf{v_j})\mathbf{v_j}^T$$

When $i \neq j$, the dot product $\mathbf{v_i}^T\mathbf{v_j}$ equals 0. However the "outer product" $\mathbf{v_i}\mathbf{v_j}^T$ is a matrix and is not zero even for $i \neq j$. Thus, $B^2 = \sum_{i=1}^{r} \sigma_i^4 \mathbf{v_i}\mathbf{v_i}^T$. In computing the k^{th} power of B, all the cross product terms are zero and

$$B^k = \sum_{i=1}^{r} \sigma_i^{2k} \mathbf{v_i}\mathbf{v_i}^T.$$

If $\sigma_1 > \sigma_2$, then

$$\frac{1}{\sigma_1^{2k}} B^k \to \mathbf{v_1}\mathbf{v_1}^T.$$

We do not know σ_1. However, if we divide B^k by $||B^k||_F$ so that the Frobenius norm is normalized to one, the matrix will converge to the rank one matrix $\mathbf{v_1}\mathbf{v_1}^T$ from which $\mathbf{v_1}$ may be computed by normalizing the first column to be a unit vector.

The difficulty with the above method is that A may be a very large, sparse matrix, say a $10^8 \times 10^8$ matrix with 10^9 nonzero entries. Sparse matrices are often represented by just

a list of non-zero entries, say, a list of triples of the form (i, j, a_{ij}). Though A is sparse, B need not be and in the worse case all 10^{16} elements may be non-zero in which case it is impossible to even store B, let alone compute the product B^2. Even if A is moderate in size, computing matrix products is costly in time. Thus, we need a more efficient method.

Instead of computing $B^k = \sigma_1^{2k}\mathbf{v_1}\mathbf{v_1}^T$, select a random vector $\mathbf{x}$ and compute the product $B^k\mathbf{x}$. The way $B^k\mathbf{x}$ is computed is by a series of matrix vector products, instead of matrix products. $B\mathbf{x} = A(A\mathbf{x})$ and $B^k\mathbf{x} = (A^TAB^{k-1}\mathbf{x})$. Thus, we perform $2k$ vector times sparse matrix multiplications. The vector $\mathbf{x}$ can be expressed in terms of the singular vectors of B augmented to a full orthonormal basis as $\mathbf{x} = \sum c_i\mathbf{v_i}$. Then

$$B^k\mathbf{x} \approx (\sigma_1^{2k}\mathbf{v_1}\mathbf{v_1}^T)\left(\sum_{i=1}^{n} c_i\mathbf{v_i}\right) = \sigma_1^{2k}c_1\mathbf{v_1}$$

Normalizing the resulting vector yields $\mathbf{v_1}$, the first singular vector of A.

An issue occurs if there is no significant gap between the first and second singular values of a matrix. If $\sigma_1 = \sigma_2$, then the above argument fails. Theorem 3.10 below states that even with ties, the power method converges to some vector in the span of those singular vectors corresponding to the "nearly highest" singular values. The theorem needs a vector $\mathbf{x}$ that has a component of at least δ along the first right singular vector $\mathbf{v_1}$ of A. Lemma 3.11 establishes that a random vector satisfies this condition.

Theorem 3.10 *Let A be an $n \times d$ matrix and $\mathbf{x}$ a unit length vector in $\mathbf{R}^d$ with $|\mathbf{x}^T\mathbf{v_1}| \geq \delta$, where, $\delta > 0$. Let V be the space spanned by the right singular vectors of A corresponding to singular values greater than $(1-\varepsilon)\,\sigma_1$. Let $\mathbf{w}$ be unit vector after $k = \ln(1/\varepsilon\delta)/\varepsilon$ iterations of the power method, namely,*

$$\mathbf{w} = \frac{\left(A^TA\right)^k\mathbf{x}}{\left|\left(A^TA\right)^k\mathbf{x}\right|}.$$

Then $\mathbf{w}$ has a component of at most ε perpendicular to V.

Proof: Let

$$A = \sum_{i=1}^{r} \sigma_i\mathbf{u_i}\mathbf{v_i}^T$$

be the SVD of A. If the rank of A is less than d, then complete $\{\mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v}_r\}$ into an orthonormal basis $\{\mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v}_d\}$ of d-space. Write $\mathbf{x}$ in the basis of the $\mathbf{v_i}'s$ as

$$\mathbf{x} = \sum_{i=1}^{n} c_i\mathbf{v_i}.$$

Since $(A^TA)^k = \sum_{i=1}^{n} \sigma_i^{2k} \mathbf{v_i}\mathbf{v_i}^T$, it follows that $(A^TA)^k\mathbf{x} = \sum_{i=1}^{n} \sigma_i^{2k} c_i \mathbf{v_i}$. By hypothesis, $|c_1| \geq \delta$.

Suppose that $\sigma_1, \sigma_2, \ldots, \sigma_m$ are the singular values of A that are greater than or equal to $(1-\varepsilon)\,\sigma_1$ and that $\sigma_{m+1}, \ldots, \sigma_n$ are the singular values that are less than $(1-\varepsilon)\,\sigma_1$. Then

$$|(A^TA)^k\mathbf{x}|^2 = \left|\sum_{i=1}^{d} \sigma_i^{2k} c_i \mathbf{v_i}\right|^2 = \sum_{i=1}^{n} \sigma_i^{4k} c_i^2 \geq \sigma_1^{4k} c_1^2 \geq \sigma_1^{4k}\delta^2.$$

The square of the component of $|(A^TA)^k\mathbf{x}|^2$ perpendicular to the space V is

$$\sum_{i=m+1}^{n} \sigma_i^{4k} c_i^2 \leq (1-\varepsilon)^{4k}\,\sigma_1^{4k} \sum_{i=m+1}^{n} c_i^2 \leq (1-\varepsilon)^{4k}\,\sigma_1^{4k}$$

since $\sum_{i=1}^{d} c_i^2 = |\mathbf{x}| = 1$. Thus, the component of $\mathbf{w}$ perpendicular to V is at most

$$\frac{(1-\varepsilon)^{2k}\sigma_1^{2k}}{\delta\sigma_1^{2k}} = (1-\varepsilon)^{2k}/\delta \leq e^{-2k\varepsilon - \ln\delta} = \varepsilon.$$

∎

Lemma 3.11 *Let $\mathbf{y} \in \mathbf{R}^n$ be a random vector with the unit variance spherical Gaussian as its probability density. Let $\mathbf{x} = \mathbf{y}/|\mathbf{y}|$. Let $\mathbf{v}$ be any fixed unit length vector. Then*

$$Prob(|\mathbf{x}^T\mathbf{v}| \leq \frac{1}{20\sqrt{d}}) \leq \frac{1}{10} + 3e^{-d/64}.$$

Proof: By Theorem 2.11 of Chapter 2 with $c = \sqrt{d}$ substituted in that theorem, we see that the probability that $|\mathbf{y}| \geq 2\sqrt{d}$ is at most $3e^{-d/64}$. Further, $\mathbf{y}^T\mathbf{v}$ is a random variable with the distribution of a unit variance Gaussian with zero mean. Thus, the probability that $|\mathbf{y}^T\mathbf{v}| \leq \frac{1}{10}$ is at most $1/10$. Combining these two and using the union bound, proves the lemma. ∎

THE FOLLOWING MATERIAL IS NOT IN PUBLIC VERSION OF BOOK

3.6 Laplacian

Different versions of the adjacency matrix are used for various purposes. Here we consider an undirected graph G with adjacency matric A.

Adjacency matrix

Since the graph is undirected the adjacency matrix is symmetric. For a random undirected graph with edge probability p, the eigenvalues obey Wigner's semi circular law and

all but the largest have a semicircular distribution between $\pm 2\sigma\sqrt{n}$. The largest eigenvalue is approximately np.

The largest eigenvalue of a symmetric matrix A is at most the maximum sum of absolute values of the elements in any row of A. To see this let λ be an eigenvalue of A and x the corresponding eigenvector. Let x_i be the maximum component of x. Now $\sum_{j=1}^{n} a_{ij}x_j = \lambda x_i$. Thus $\sum_{j\neq i} a_{ij}x_j = \lambda x_i - a_{ii}x_i$ and hence

$$|\lambda - a_{ii}| = \left| \frac{\sum_{j\neq i} a_{ij}x_j}{x_i} \right| \leq \sum_{j\neq i} \left| \frac{a_{ij}x_j}{x_i} \right| \leq \sum_{j\neq i} |a_{ij}|.$$

It follows that $\lambda \leq \sum_{j=1}^{n} |a_{ij}|$.

Laplacian

The Laplacian is defined to be $L = D - A$ where D is a diagonal matrix with the vertex degrees on its diagonal.

$$l_{ij} = \begin{cases} d_{ij} & i = j \\ -1 & i \neq j \text{ and there is an edge from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

The smallest eigenvalue of L is zero since each row of L sums to zero and thus the all 1's vector is an eigenvector with eigenvalue zero. The number of eigenvalues equal to zero equals the number of connected components of the graph G. All other eigenvalues are greater than zero. Thus, the matric L is positive semi definite. The maximum eigenvalue is at most twice the maximum of any row sum of L which is at most twice the maximum degree.

To see that all eigenvalues of L are nonnegative define an incidence matrix B whose rows correspond to edges of the graph G and whose columns correspond to vertices of G.

$$b_{ij} = \begin{cases} 1 & i^{th} \text{ edge is } (k, j) \\ -1 & i^{th} \text{ edge is } (j, k) \\ 0 & \text{otherwise} \end{cases}$$

The Laplacian matrix can be expressed as $L = M^T M$. This follows since each row of M^T gives the edges incident to the corresponding vertex. Some entries are +1 and some -1. The diagonal entries of $M^T M$ are the length of the corresponding vectors and the off diagonal ij^{th} entry will be 0 or -1 depending on whether an edge incident to vertex i is also incident to vertex j. If v is an eigenvector of L with eigenvalue λ, then $\lambda = (Mv)^T Mv \geq 0$. Thus L is positive semi definite and hence all eigenvalues are nonnegative.

Symmetric normalized Laplacian
Sometimes the Laplacian is normalized. Define $L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$. L_{sym} is symmetric and all eigenvalues are nonnegative since

$$L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = D^{-\frac{1}{2}}M^TMD^{-\frac{1}{2}} = (D^{-\frac{1}{2}}M)^T(MD^{-\frac{1}{2}})$$

is positive semi definite. The eigenvalues of L_{sym} are in the range $0 \leq \lambda \leq 2$.
Spectral gap

Adjacency matrix normalized for a random walk
In doing a random walk on a graph one wants each row to sum to one so that the entries are probabilities of taking an edge. To do this one defines a transition matrix $T = D^{-1}A$. In a random walk we have adopted the notation $p^T(t+1) = p^T(t)T$ and this requires the rows instead of the columns to sum to one.

Laplacian normalized for random walk

To use the L for a random walk one needs to normalize the edge probability by the degree. This is done by multiplying by D^{-1} to get $D^{-1}L = D^{-1}(D - A) = I - D^{-1}A = I - T$.

3.7 Applications of Singular Value Decomposition

3.7.1 Principal Component Analysis

The traditional use of SVD is in Principal Component Analysis (PCA). PCA is illustrated by a customer-product data problem where there are n customers buying d products. Let matrix A with elements a_{ij} represent the probability of customer i purchasing product j. One hypothesizes that there are only k underlying basic factors like age, income, family size, etc. that determine a customer's purchase behavior. An individual customer's behavior is determined by some weighted combination of these underlying factors. That is, a customer's purchase behavior can be characterized by a k-dimensional vector where k is much smaller than n or d. The components of the vector are weights for each of the basic factors. Associated with each basic factor is a vector of probabilities, each component of which is the probability of purchasing a given product by someone whose behavior depends only on that factor. More abstractly, A is an $n \times d$ matrix that can be expressed as the product of two matrices U and V where U is an $n \times k$ matrix expressing the factor weights for each customer and V is a $k \times d$ matrix expressing the purchase probabilities of products that correspond to that factor. Finding the best rank k approximation A_k by SVD gives such a U and V. One twist is that A may not be exactly equal to UV, but close to it since there may be noise or random perturbations in which case $A - UV$ is treated as noise.

In the above setting, A was available fully and we wished to find U and V to identify the basic factors. If n and d are very large, on the order of thousands or even millions,

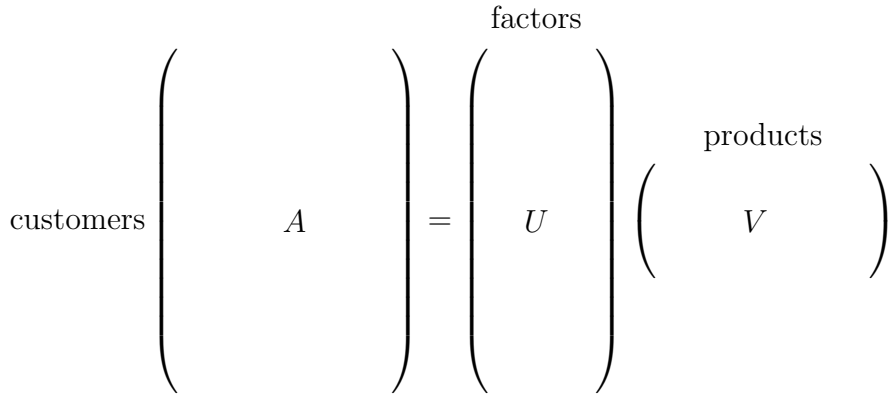


Figure 3.3: Customer-product data

there is probably little one could do to estimate or even store A. In this setting, we may assume that we are given just a few elements of A and wish to estimate A. If A was an arbitrary matrix of size $n \times d$, this would require $\Omega(nd)$ pieces of information and cannot be done with a few entries. But again hypothesize that A was a small rank matrix with added noise. If now we also assume that the given entries are randomly drawn according to some known distribution, then there is a possibility that SVD can be used to estimate the whole of A. This area is called collaborative filtering and one of its uses is to target an ad to a customer based on one or two purchases. We do not describe it here.

3.7.2 Clustering a Mixture of Spherical Gaussians

Clustering, is the task of partitioning a set of points in d-space into k subsets or clusters where each cluster consists of "nearby" points. Different definitions of the goodness of a clustering lead to different solutions. Clustering is an important area which we will study in detail in Chapter **??**. Here we solve a particular clustering problem using singular value decomposition.

In general, a solution to any clustering problem comes up with k *cluster centers* that define the k clusters. A cluster is the set of data points that are closest to a particular cluster center. Hence the Vornoi cells of the cluster centers determine the clusters. Using this observation, it is relatively easy to cluster points in two or three dimensions. However, clustering is not so easy in higher dimensions. Many problems have high-dimensional data and clustering problems are no exception.

Clustering problems tend to be NP-hard, so we there are no polynomial time algorithms to solve them. One way around this is to assume stochastic models of input data and devise algorithms to cluster data generated by such models. Mixture models are a very important class of stochastic models. A mixture is a probability density or distribution that is the weighted sum of simple component probability densities. It is of the

form

$$F = w_1 p_1 + w_2 p_2 + \cdots + w_k p_k,$$

where $p_1, p_2, \ldots, p_k$ are the basic probability densities and $w_1, w_2, \ldots, w_k$ are positive real numbers called weights that add up to one. Clearly, F is a probability density, it integrates to one.

The *model fitting problem* is to fit a mixture of k basic densities to n independent, identically distributed samples, each sample drawn according to the same mixture distribution F. The class of basic densities is known, but various parameters such as their means and the component weights of the mixture are not. Here, we deal with the case where the basic densities are all spherical Gaussians. There are two equivalent ways of thinking of the sample generation process which is hidden, only the samples are given.

1. Pick each sample according to the density F on $\mathbf{R}^d$.

2. Pick a random i from $\{1, 2, \ldots, k\}$ where probability of picking i is w_i. Then, pick a sample according to the density F_i.

The model-fitting problem can be broken up into two sub problems:

- The first sub problem is to cluster the set of samples into k clusters $C_1, C_2, \ldots, C_k$, where, C_i is the set of samples generated according to F_i, see (2) above, by the hidden generation process.

- The second sub problem is to fit a single Gaussian distribution to each cluster of sample points.

The second problem is easier than the first and in Chapter (2) we showed that taking the empirical mean, the mean of the sample, and the empirical standard deviation gives the best-fit Gaussian. The first problem is harder and this is what we discuss here.

If the component Gaussians in the mixture have their centers very close together, then the clustering problem is unresolvable. In the limiting case where a pair of component densities are the same, there is no way to distinguish between them. What condition on the inter-center separation will guarantee unambiguous clustering? First, by looking at 1-dimensional examples, it is clear that this separation should be measured in units of the standard deviation, since the density is a function of the number of standard deviation from the mean. In one dimension, if two Gaussians have inter-center separation at least six times the maximum of their standard deviations, then they hardly overlap.

How far apart must the means be to determine which Gaussian a point belongs to. In one dimension, if the distance is at least six standard deviations, we separate the Gaussians. What is the analog of this in higher dimensions?

We discussed in Chapter (2) distances between two sample points from the same Gaussian as well the distance between two sample points from two different Gaussians. Recall from that discussion that if

- If $\mathbf{x}$ and $\mathbf{y}$ are two independent samples from the same spherical Gaussian with standard deviation[1] σ, then

$$|\mathbf{x} - \mathbf{y}|^2 \approx 2(\sqrt{d} \pm c)^2\sigma^2.$$

- If $\mathbf{x}$ and $\mathbf{y}$ are samples from different spherical Gaussians each of standard deviation σ and means separated by distance δ, then

$$|\mathbf{x} - \mathbf{y}|^2 \approx 2(\sqrt{d} \pm c)^2\sigma^2 + \delta^2.$$

Now we would like to assert that points from the same Gaussian are closer to each other than points from different Gaussians. To ensure this, we need

$$2(\sqrt{d} - c)^2\sigma^2 + \delta^2 > 2(\sqrt{d} + c)^2\sigma^2.$$

Expanding the squares, the high order term $2d$ cancels and we need that

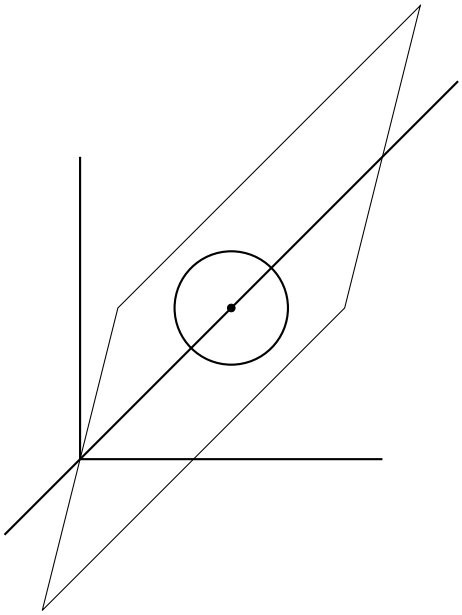
$$\delta > c' d^{1/4}.$$

While this was not a completely rigorous argument, it can be used to show that a distance based clustering approach requires an inter-mean separation of at least $c' d^{1/4}$ standard deviations to succeed, thus unfortunately not keeping within a constant number of standard deviations separation of the means. Here, indeed, we will show that $\Omega(1)$ standard deviations suffice, provided $k \in O(1)$.

The central idea is the following. Suppose we can find the subspace spanned by the k centers and project the sample points to this subspace. The projection of a spherical Gaussian with standard deviation σ remains a spherical Gaussian with standard deviation σ, Lemma 3.12. In the projection, the inter-center separation remains the same. So in the projection, the Gaussians are distinct provided the inter-center separation in the whole space is $\Omega(k^{1/4}\,\sigma)$ which is a lot smaller than the $\Omega(d^{1/4}\,\sigma)$ for $k << d$. Interestingly, we will see that the subspace spanned by the k-centers is essentially the best-fit k-dimensional subspace that can be found by singular value decomposition.

Lemma 3.12 *Suppose p is a d-dimensional spherical Gaussian with center μ and standard deviation σ. The density of p projected onto a k-dimensional subspace V is a spherical Gaussian with the same standard deviation.*

[1]Since a spherical Gaussian has the same standard deviation in every direction, we call it the standard deviation of the Gaussian.



1. The best fit 1-dimension subspace to a spherical Gaussian is the line through its center and the origin.
2. Any k-dimensional subspace containing the line is a best fit k-dimensional subspace for the Gaussian.
3. The best fit k-dimensional subspace for k spherical Gaussians is the subspace containing their centers.

Figure 3.4: Best fit subspace to a spherical Gaussian.

Proof: Rotate the coordinate system so V is spanned by the first k coordinate vectors. The Gaussian remains spherical with standard deviation σ although the coordinates of its center have changed. For a point $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, we will use the notation $\mathbf{x}' = (x_1, x_2, \ldots x_k)$ and $\mathbf{x}'' = (x_{k+1}, x_{k+2}, \ldots, x_n)$. The density of the projected Gaussian at the point $(x_1, x_2, \ldots, x_k)$ is

$$ce^{-\frac{|\mathbf{x}'-\boldsymbol{\mu}'|^2}{2\sigma^2}} \int_{\mathbf{x}''} e^{-\frac{|\mathbf{x}''-\boldsymbol{\mu}''|^2}{2\sigma^2}} d\mathbf{x}'' = c'e^{-\frac{|\mathbf{x}'-\boldsymbol{\mu}'|^2}{2\sigma^2}}.$$

This clearly implies the lemma. ∎

We now show that the top k singular vectors produced by the SVD span the space of the k centers. First, we extend the notion of best fit to probability distributions. Then we show that for a single spherical Gaussian whose center is not the origin, the best fit 1-dimensional subspace is the line though the center of the Gaussian and the origin. Next, we show that the best fit k-dimensional subspace for a single Gaussian whose center is not the origin is any k-dimensional subspace containing the line through the Gaussian's center and the origin. Finally, for k spherical Gaussians, the best fit k-dimensional subspace is the subspace containing their centers. Thus, the SVD finds the subspace that contains the centers.

Recall that for a set of points, the best-fit line is the line passing through the origin that minimizes the sum of squared distances to the points. We extend this definition to probability densities instead of a set of points.

Definition 3.1 *If p is a probability density in d space, the best fit line for p is the line l passing through the origin that minimizes the expected squared perpendicular distance to the line, namely,*

$$\int dist(\mathbf{x}, l)^2 \, p(\mathbf{x}) \, d\mathbf{x}.$$

■

A word of caution: The integral may not exist. We assume that it does when we write it down.

For the uniform density on the unit circle centered at the origin, it is easy to see that any line passing through the origin is a best fit line for the probability distribution. Our next lemma shows that the best fit line for a spherical Gaussian centered at $\boldsymbol{\mu} \neq 0$ is the line passing through $\boldsymbol{\mu}$ and the origin.

Lemma 3.13 *Let the probability density p be a spherical Gaussian with center $\boldsymbol{\mu} \neq 0$. The unique best fit 1-dimensional subspace is the line passing through $\boldsymbol{\mu}$ and the origin. If $\boldsymbol{\mu} = 0$, then any line through the origin is a best-fit line.*

Proof: For a randomly chosen $\boldsymbol{x}$ (according to p) and a fixed unit length vector $\mathbf{v}$,

$$\begin{aligned}
E\left[(\mathbf{v}^T\mathbf{x})^2\right] &= E\left[\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu}) + \mathbf{v}^T\boldsymbol{\mu}\right)^2\right] \\
&= E\left[\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right)^2 + 2\left(\mathbf{v}^T\boldsymbol{\mu}\right)\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right) + \left(\mathbf{v}^T\boldsymbol{\mu}\right)^2\right] \\
&= E\left[\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right)^2\right] + 2\left(\mathbf{v}^T\boldsymbol{\mu}\right) E\left[\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right] + \left(\mathbf{v}^T\boldsymbol{\mu}\right)^2 \\
&= E\left[\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right)^2\right] + \left(\mathbf{v}^T\boldsymbol{\mu}\right)^2 \\
&= \sigma^2 + \left(\mathbf{v}^T\boldsymbol{\mu}\right)^2
\end{aligned}$$

since $E\left[\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right)^2\right]$ is the variance in the direction $\mathbf{v}$ and $E\left(\mathbf{v}^T(\mathbf{x}-\boldsymbol{\mu})\right) = 0$. The lemma follows from the fact that the best fit line $\mathbf{v}$ is the one that maximizes $\left(\mathbf{v}^T\boldsymbol{\mu}\right)^2$ which is maximized when $\mathbf{v}$ is aligned with the center $\boldsymbol{\mu}$. To see the uniqueness, just note that if $\boldsymbol{\mu} \neq 0$, then $\mathbf{v}^T\boldsymbol{\mu}$ is strictly smaller when $\mathbf{v}$ is not aligned with the center. ■

Recall that a k-dimensional subspace is the best-fit subspace if the sum of squared distances to it is minimized or equivalently, the sum of squared lengths of projections onto it is maximized. This was defined for a set of points, but again it can be extended to a density as we did for best-fit lines.

Definition 3.2 *If p is a probability density in d-space and V is a subspace, then the expected squared perpendicular distance of V to p, denoted $f(V,p)$, is given by*

$$f(V,p) = \int \left(dist\left(\mathbf{x}, V\right)\right)^2 p\left(\mathbf{x}\right) d\mathbf{x},$$

where dist$(\mathbf{x}, V)$ denotes the perpendicular distance from the point $\mathbf{x}$ to the subspace V.

∎

Lemma 3.14 *For a spherical Gaussian with center $\boldsymbol{\mu}$, a k-dimensional subspace is a best fit subspace if and only if it contains $\boldsymbol{\mu}$.*

Proof: If $\boldsymbol{\mu} = \mathbf{0}$, then by symmetry any k-dimensional subspace is a best-fit subspace. If $\boldsymbol{\mu} \neq \mathbf{0}$, then the best-fit line must pass through $\boldsymbol{\mu}$ by Lemma 3.13. Now, as in the greedy algorithm for finding subsequent singular vectors, we would project perpendicular to the first singular vector. But after the projection, the mean of the Gaussian becomes $\mathbf{0}$ and then any vectors will do as subsequent best-fit directions. ∎

This leads to the following theorem.

Theorem 3.15 *If p is a mixture of k spherical Gaussians , then the best fit k-dimensional subspace contains the centers. In particular, if the means of the Gaussians are linearly independent, the space spanned by them is the unique best-fit k dimensional subspace.*

Proof: Let p be the mixture $w_1 p_1 + w_2 p_2 + \cdots + w_k p_k$. Let V be any subspace of dimension k or less. The expected squared perpendicular distance of V to p is

$$\begin{aligned} f(V,p) &= \int \text{dist}^2(\mathbf{x}, V) p(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i=1}^{k} w_i \int \text{dist}^2(\mathbf{x}, V) p_i(\mathbf{x}) d\mathbf{x} \\ &\geq \sum_{i=1}^{k} w_i (\text{ distance squared of } p_i \text{ to its best fit } k\text{-dimensional subspace}). \end{aligned}$$

If a subspace V contains the centers of the densities p_i, by Lemma **??** the last inequality becomes an equality proving the theorem. Indeed, for each i individually, we have equality which is stronger than just saying we have equality for the sum. ∎

For an infinite set of points drawn according to the mixture, the k-dimensional SVD subspace gives exactly the space of the centers. In reality, we have only a large number of samples drawn according to the mixture. However, it is intuitively clear that as the number of samples increases, the set of sample points approximates the probability density and so the SVD subspace of the sample is close to the space spanned by the centers. The details of how close it gets as a function of the number of samples are technical and we do not carry this out here.

3.7.3 Spectral Decomposition

Let B be a square matrix. If the vector $\mathbf{x}$ and scalar λ are such that $B\mathbf{x} = \lambda\mathbf{x}$, then $\mathbf{x}$ is an *eigenvector* of the matrix B and λ is the corresponding *eigenvalue*. We present here a spectral decomposition theorem for the special case where B is of the form $B = AA^T$ for some possibly rectangular matrix A. If A is a real valued matrix, then B is symmetric and positive definite. That is, $\mathbf{x}^T B\mathbf{x} > 0$ for all nonzero vectors $\mathbf{x}$. The spectral decomposition theorem holds more generally and the interested reader should consult a linear algebra book.

Theorem 3.16 (Spectral Decomposition) *If $B = AA^T$ then $B = \sum_i \sigma_i^2 u_i u_i^T$ where $A = \sum_i \sigma_i u_i v_i^T$ is the singular valued decomposition of A.*

Proof:

$$\begin{aligned} B = AA^T &= \left(\sum_i \sigma_i \mathbf{u_i}\mathbf{v_i}^T\right)\left(\sum_j \sigma_j \mathbf{u_j} v_j^T\right)^T \\ &= \sum_i \sum_j \sigma_i \sigma_j \mathbf{u_i}\mathbf{v_i}^T \mathbf{v_j}\mathbf{u_j}^T \\ &= \sum_i \sigma_i^2 \mathbf{u_i}\mathbf{u_i}^T. \end{aligned}$$

∎

When the σ_i are all distinct, the $\mathbf{u_i}$ are the eigenvectors of B and the σ_i^2 are the corresponding eigenvalues. If the σ_i are not distinct, then any vector that is a linear combination of those $\mathbf{u_i}$ with the same eigenvalue is an eigenvector of B.

3.7.4 Singular Vectors and Ranking Documents

An important task for a document collection is to rank the documents according to their intrinsic relevance to the collection. A good candidate is a document's projection onto the best-fit direction for the collection of term-document vectors, namely the top left-singular vector of the term-document matrix. An intuitive reason for this is that this direction has the maximum sum of squared projections of the collection and so can be thought of as a synthetic term-document vector best representing the document collection.

Ranking in order of the projection of each document's term vector along the best fit direction has a nice interpretation in terms of the power method. For this, we consider a different example, that of the web with hypertext links. The World Wide Web can be represented by a directed graph whose nodes correspond to web pages and directed edges to hypertext links between pages. Some web pages, called *authorities*, are the most prominent sources for information on a given topic. Other pages called *hubs*, are ones

that identify the authorities on a topic. Authority pages are pointed to by many hub pages and hub pages point to many authorities. One is led to what seems like a circular definition: a hub is a page that points to many authorities and an authority is a page that is pointed to by many hubs.

One would like to assign hub weights and authority weights to each node of the web. If there are n nodes, the hub weights form a n-dimensional vector $\mathbf{u}$ and the authority weights form a n-dimensional vector $\mathbf{v}$. Suppose A is the adjacency matrix representing the directed graph. Here a_{ij} is 1 if there is a hypertext link from page i to page j and 0 otherwise. Given hub vector $\mathbf{u}$, the authority vector $\mathbf{v}$ could be computed by the formula

$$v_j = \sum_{i=1}^{d} u_i a_{ij}$$

since the right hand side is the sum of the hub weights of all the nodes that point to node j. In matrix terms,

$$\mathbf{v} = A^T \mathbf{u}.$$

Similarly, given an authority vector $\mathbf{v}$, the hub vector $\mathbf{u}$ could be computed by $\mathbf{u} = A\mathbf{v}$. Of course, at the start, we have neither vector. But the above discussion suggests a power iteration. Start with any $\mathbf{v}$. Set $\mathbf{u} = A\mathbf{v}$; then set $\mathbf{v} = A^T\mathbf{u}$ and repeat the process. We know from the power method that this converges to the left and right-singular vectors. So after sufficiently many iterations, we may use the left vector $\mathbf{u}$ as hub weights vector and project each column of A onto this direction and rank columns (authorities) in order of their projections. But the projections just form the vector $A^T\mathbf{u}$ which equals $\mathbf{v}$. So we can rank by order of the v_j. This is the basis of an algorithm called the HITS algorithm, which was one of the early proposals for ranking web pages.

A different ranking called *page rank* is widely used. It is based on a random walk on the graph described above. We will study random walks in detail in Chapter 5.

3.7.5 An Application of SVD to a Discrete Optimization Problem

In Gaussian clustering the SVD was used as a dimension reduction technique. It found a k-dimensional subspace containing the centers of the Gaussians in a d-dimensional space and made the Gaussian clustering problem easier by projecting the data to the subspace. Here, instead of fitting a model to data, we have an optimization problem. Again applying dimension reduction to the data makes the problem easier. The use of SVD to solve discrete optimization problems is a relatively new subject with many applications. We start with an important NP-hard problem, the maximum cut problem for a directed graph $G(V, E)$.

The maximum cut problem is to partition the node set V of a directed graph into two subsets S and $\bar{S}$ so that the number of edges from S to $\bar{S}$ is maximized. Let A be the

adjacency matrix of the graph. With each vertex i, associate an indicator variable x_i. The variable x_i will be set to 1 for $i \in S$ and 0 for $i \in \bar{S}$. The vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is unknown and we are trying to find it or equivalently the cut, so as to maximize the number of edges across the cut. The number of edges across the cut is precisely

$$\sum_{i,j} x_i(1 - x_j)a_{ij}.$$

Thus, the maximum cut problem can be posed as the optimization problem

$$\text{Maximize} \sum_{i,j} x_i(1 - x_j)a_{ij} \quad \text{subject to } x_i \in \{0, 1\}.$$

In matrix notation,

$$\sum_{i,j} x_i(1 - x_j)a_{ij} = \mathbf{x}^T A(\mathbf{1} - \mathbf{x}),$$

where $\mathbf{1}$ denotes the vector of all 1's . So, the problem can be restated as

$$\text{Maximize } \mathbf{x}^T A(\mathbf{1} - \mathbf{x}) \quad \text{subject to } x_i \in \{0, 1\}. \tag{3.1}$$

The SVD is used to solve this problem approximately by computing the SVD of A and replacing A by $A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^T$ in (3.1) to get

$$\text{Maximize } \mathbf{x}^T A_k(\mathbf{1} - \mathbf{x}) \quad \text{subject to } x_i \in \{0, 1\}. \tag{3.2}$$

Note that the matrix A_k is no longer a 0-1 adjacency matrix.

We will show that:

1. For each 0-1 vector $\mathbf{x}$, $\mathbf{x}^T A_k(\mathbf{1} - \mathbf{x})$ and $\mathbf{x}^T A(\mathbf{1} - \mathbf{x})$ differ by at most $\frac{n^2}{\sqrt{k+1}}$. Thus, the maxima in (3.1) and (3.2) differ by at most this amount.

2. A near optimal $\mathbf{x}$ for (3.2) can be found by exploiting the low rank of A_k, which by Item 1 is near optimal for (3.1) where near optimal means with additive error of at most $\frac{n^2}{\sqrt{k+1}}$.

First, we prove Item 1. Since $\mathbf{x}$ and $\mathbf{1} - \mathbf{x}$ are 0-1 n-vectors, each has length at most $\sqrt{n}$. By the definition of the 2-norm, $|(A - A_k)(\mathbf{1} - \mathbf{x})| \leq \sqrt{n}||A - A_k||_2$. Now since $\mathbf{x}^T(A - A_k)(\mathbf{1} - \mathbf{x})$ is the dot product of the vector $\mathbf{x}$ with the vector $(A - A_k)(\mathbf{1} - \mathbf{x})$,

$$|\mathbf{x}^T(A - A_k)(\mathbf{1} - \mathbf{x})| \leq n||A - A_k||_2.$$

By Lemma 3.8, $||A - A_k||_2 = \sigma_{k+1}(A)$. The inequalities,

$$(k + 1)\sigma_{k+1}^2 \leq \sigma_1^2 + \sigma_2^2 + \cdots \sigma_{k+1}^2 \leq ||A||_F^2 = \sum_{i,j} a_{ij}^2 \leq n^2$$

imply that $\sigma_{k+1}^2 \leq \frac{n^2}{k+1}$ and hence $||A - A_k||_2 \leq \frac{n}{\sqrt{k+1}}$ proving Item 1.

Next we focus on Item 2. It is instructive to look at the special case when k=1 and A is approximated by the rank one matrix A_1. An even more special case when the left and right-singular vectors $\mathbf{u}$ and $\mathbf{v}$ are required to be identical is already NP-hard to solve exactly because it subsumes the problem of whether for a set of n integers, $\{a_1, a_2, \ldots, a_n\}$, there is a partition into two subsets whose sums are equal. So, we look for algorithms that solve the maximum cut problem approximately.

For Item 2, we want to maximize $\sum_{i=1}^{k} \sigma_i(\mathbf{x}^T\mathbf{u_i})(\mathbf{v_i}^T(\mathbf{1} - \mathbf{x}))$ over 0-1 vectors $\mathbf{x}$. A piece of notation will be useful. For any $S \subseteq \{1, 2, \ldots n\}$, write $\mathbf{u_i}(S)$ for the sum of coordinates of the vector $\mathbf{u_i}$ corresponding to elements in the set S and also for $\mathbf{v_i}$. That is, $\mathbf{u}_i(S) = \sum_{j \in S} u_{ij}$. We will maximize $\sum_{i=1}^{k} \sigma_i \mathbf{u_i}(S)\mathbf{v_i}(\bar{S})$ using dynamic programming.

For a subset S of $\{1, 2, \ldots, n\}$, define the $2k$-dimensional vector

$$\mathbf{w}(S) = (\mathbf{u_1}(S), \mathbf{v_1}(\bar{S}), \mathbf{u_2}(S), \mathbf{v_2}(\bar{S}), \ldots, \mathbf{u_k}(S), \mathbf{v_k}(\bar{S})).$$

If we had the list of all such vectors, we could find $\sum_{i=1}^{k} \sigma_i \mathbf{u_i}(S)\mathbf{v_i}(\bar{S})$ for each of them and take the maximum. There are 2^n subsets S, but several S could have the same $\mathbf{w}(S)$ and in that case it suffices to list just one of them. Round each coordinate of each $\mathbf{u_i}$ to the nearest integer multiple of $\frac{1}{nk^2}$. Call the rounded vector $\tilde{\mathbf{u}}_\mathbf{i}$. Similarly obtain $\tilde{\mathbf{v}}_\mathbf{i}$. Let $\tilde{\mathbf{w}}(S)$ denote the vector $(\tilde{\mathbf{u}}_\mathbf{1}(S), \tilde{\mathbf{v}}_\mathbf{1}(\bar{S}), \tilde{\mathbf{u}}_\mathbf{2}(S), \tilde{\mathbf{v}}_\mathbf{2}(\bar{S}), \ldots, \tilde{\mathbf{u}}_\mathbf{k}(S), \tilde{\mathbf{v}}_\mathbf{k}(\bar{S}))$. We will construct a list of all possible values of the vector $\tilde{\mathbf{w}}(S)$. Again, if several different S's lead to the same vector $\tilde{\mathbf{w}}(S)$, we will keep only one copy on the list. The list will be constructed by dynamic programming. For the recursive step of dynamic programming, assume we already have a list of all such vectors for $S \subseteq \{1, 2, \ldots, i\}$ and wish to construct the list for $S \subseteq \{1, 2, \ldots, i+1\}$. Each $S \subseteq \{1, 2, \ldots, i\}$ leads to two possible $S' \subseteq \{1, 2, \ldots, i+1\}$, namely, S and $S \cup \{i+1\}$. In the first case, the vector $\tilde{\mathbf{w}}(S') = (\tilde{\mathbf{u}}_\mathbf{1}(S), \tilde{\mathbf{v}}_\mathbf{1}(\bar{S}) + \tilde{v}_{1,i+1}, \tilde{\mathbf{u}}_\mathbf{2}(S), \tilde{\mathbf{v}}_\mathbf{2}(\bar{S}) + \tilde{v}_{2,i+1}, \ldots, ...)$. In the second case, it is $\tilde{\mathbf{w}}(S') = (\tilde{\mathbf{u}}_\mathbf{1}(S) + \tilde{u}_{1,i+1}, \tilde{\mathbf{v}}_\mathbf{1}(\bar{S}), \tilde{\mathbf{u}}_\mathbf{2}(S) + \tilde{u}_{2,i+1}, \tilde{\mathbf{v}}_\mathbf{2}(\bar{S}), \ldots, ...)$ We put in these two vectors for each vector in the previous list. Then, crucially, we prune - i.e., eliminate duplicates.

Assume that k is constant. Now, we show that the error is at most $\frac{n^2}{\sqrt{k+1}}$ as claimed. Since $\mathbf{u_i}, \mathbf{v_i}$ are unit length vectors, $|\mathbf{u_i}(S)|, |\mathbf{v_i}(\bar{S})| \leq \sqrt{n}$. Also $|\tilde{\mathbf{u}}_\mathbf{i}(S) - \mathbf{u_i}(S)| \leq \frac{n}{nk^2} = \frac{1}{k^2}$ and similarly for $\mathbf{v_i}$. To bound the error, we use an elementary fact: if a, b are reals with $|a|, |b| \leq M$ and we estimate a by a' and b by b' so that $|a - a'|, |b - b'| \leq \delta \leq M$, then $a'b'$ is an estimate of ab in the sense

$$|ab - a'b'| = |a(b - b') + b'(a - a')| \leq |a||b - b'| + (|b| + |b - b'|)|a - a'| \leq 3M\delta.$$

Using this, we get that

$$\left| \sum_{i=1}^{k} \sigma_i \tilde{\mathbf{u}}_\mathbf{i}(S)\tilde{\mathbf{v}}_\mathbf{i}(\bar{S}) \quad - \quad \sum_{i=1}^{k} \sigma_i \mathbf{u_i}(S)\mathbf{v_i}(S) \right| \leq 3k\sigma_1\sqrt{n}/k^2 \leq 3n^{3/2}/k \leq n^2/k,$$

and this meets the claimed error bound.

Next, we show that the running time is polynomially bounded. $|\tilde{\mathbf{u}}_\mathbf{i}(S)|, |\tilde{\mathbf{v}}_\mathbf{i}(S)| \leq 2\sqrt{n}$. Since $\tilde{\mathbf{u}}_\mathbf{i}(S), \tilde{\mathbf{v}}_\mathbf{i}(S)$ are all integer multiples of $1/(nk^2)$, there are at most $2/\sqrt{n}k^2$ possible values of $\tilde{\mathbf{u}}_\mathbf{i}(S), \tilde{\mathbf{v}}_\mathbf{i}(S)$ from which it follows that the list of $\tilde{\mathbf{w}}(S)$ never gets larger than $(1/\sqrt{n}k^2)^{2k}$ which for fixed k is polynomially bounded.

We summarize what we have accomplished.

Theorem 3.17 *Given a directed graph $G(V, E)$, a cut of size at least the maximum cut minus $O\left(\frac{n^2}{\sqrt{k}}\right)$ can be computed in polynomial time n for any fixed k.*

It would be quite a surprise to have an algorithm that actually achieves the same accuracy in time polynomial in n and k because this would give an exact max cut in polynomial time.

3.8 Singular Vectors and Eigenvectors

An eigenvector of a square matrix A is a vector $\mathbf{v}$ satisfying $A\mathbf{v} = \lambda\mathbf{v}$, for a non-zero scaler λ which is the corresponding eigenvalue. A square matrix A can be viewed as a linear transformation from a space into itself which transforms an eigenvector into a scaler multiple of itself. The eigenvector decomposition of A is $V^T DV$ where the columns of V are the eigenvectors of A and D is a diagonal matrix with the eigenvalues on the diagonal.

A non square $m \times n$ matrix A also defines a linear transformation, but now from $\mathbf{R}^n$ to $\mathbf{R}^m$. In this case, eigenvectors do not make sense. But singular vectors can be defined. They serve the purpose of decomposing the linear transformation defined by the matrix A into the sum of simple linear transformations, each of which maps $\mathbf{R}_n$ to a one dimensional space, i.e., to a line through the origin.

A positive semi-definite matrix can be decomposed into a product AA^T. Thus, the eigenvector decomposition can be obtained from the singular value decomposition of $A = UDV^T$ since

$$AA^T = UDV^T VDU^T = UD^2U^T = \sum_i \sigma_i(A)^2 \mathbf{u}_\mathbf{i}\mathbf{u}_\mathbf{i}{}^T,$$

where the $\mathbf{u}_\mathbf{i}$, the columns of U, are the eigenvectors of AA^T.

There are many applications of singular vectors and eigenvectors. For square non-symmetric matrices, both singular vectors and eigenvectors are defined but they may be different. In an important application, the pagerank, one represents the web by a $n \times n$ matrix A, where, a_{ij} is one if there is a hypertext link from the i^{th} page in the web to the j^{th} page. Otherwise, it is zero. The matrix is scaled by dividing each entry by the sum of entries in its row to get a stochastic matrix P. A stochastic matrix is one with nonnegative

entries where each row sums to one. Note that P is not necessarily symmetric. Since the row sums of P are all one, the vector $\mathbf{1}$ of all one's is a right eigenvector with eigenvalue one, i.e., $P\mathbf{1} = \mathbf{1}$. This eigenvector contains no information. But the left eigenvector $\mathbf{v}$ with eigenvalue one satisfies $\mathbf{v}^T P = \mathbf{v}^T$ and is the stationary probability of the Markov chain with transition probability matrix P. So, it is the proportion of time a Markov chain spends at each vertex (page) in the long run. A simplified definition of pagerank ranks the page in order of its component in the top left eigenvector $\mathbf{v}$.

3.9 Bibliographic Notes

Singular value decomposition is fundamental to numerical analysis and linear algebra. There are many texts on these subjects and the interested reader may want to study these. A good reference is [GvL96]. The material on clustering a mixture of Gaussians in Section 3.7.2 is from [VW02]. Modeling data with a mixture of Gaussians is a standard tool in statistics. Several well-known heuristics like the expectation-minimization algorithm are used to fit the mixture model to data. Recently, in theoretical computer science, there has been modest progress on provable polynomial-time algorithms for learning mixtures. Some references are [DS07], [AK], [AM05], [MV10]. The application to the discrete optimization problem is from [FK99]. The section on ranking documents/webpages is from two influential papers, one on hubs and authorities by Jon Kleinberg [Kle99] and the other on pagerank by Page, Brin, Motwani and Winograd [BMPW98].