Matrix Computations in Data Mining

Rafikul Alam
Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati - 781039, INDIA

Outline

Data mining is truly an interdisciplinary subject, where techniques from computer science, statistics and data analysis, pattern recognition, linear algebra and optimization are used extensively.

Outline

Data mining is truly an interdisciplinary subject, where techniques from computer science, statistics and data analysis, pattern recognition, linear algebra and optimization are used extensively.

We discuss a few areas of data mining where matrix computations plays an important role:

Outline

Data mining is truly an interdisciplinary subject, where techniques from computer science, statistics and data analysis, pattern recognition, linear algebra and optimization are used extensively.

We discuss a few areas of data mining where matrix computations plays an important role:

- Text mining (information retrieval).

Outline

Data mining is truly an interdisciplinary subject, where techniques from computer science, statistics and data analysis, pattern recognition, linear algebra and optimization are used extensively.

We discuss a few areas of data mining where matrix computations plays an important role:

- Text mining (information retrieval).
- Pattern recognition (classification of handwritten digits, face recognition).

Outline

Data mining is truly an interdisciplinary subject, where techniques from computer science, statistics and data analysis, pattern recognition, linear algebra and optimization are used extensively.

We discuss a few areas of data mining where matrix computations plays an important role:

- Text mining (information retrieval).
- Pattern recognition (classification of handwritten digits, face recognition).
- Google's PageRank algorithm for web search engine.

Matrices and Vectors in Data Mining

Text mining: Term-document matrices are used in information retrieval. Consider the five documents.

Matrices and Vectors in Data Mining

Text mining: Term-document matrices are used in information retrieval. Consider the five documents.

Doc. 1:	The Google matrix G is a model of the Internet.
Doc. 2:	G_{ij} is nonzero if there is a link from web page j to i.
Doc. 3:	The Google matrix G is used to rank all web pages.
Doc. 4:	The ranking is done by solving a matrix eigenvalue problem.
Doc. 5:	England dropped out of the top 10 in the FIFA ranking.

Matrices and Vectors in Data Mining

Text mining: Term-document matrices are used in information retrieval. Consider the five documents.

Doc. 1:	The Google matrix G is a model of the Internet.
Doc. 2:	G_{ij} is nonzero if there is a link from web page j to i.
Doc. 3:	The Google matrix G is used to rank all web pages.
Doc. 4:	The ranking is done by solving a matrix eigenvalue problem.
Doc. 5:	England dropped out of the top 10 in the FIFA ranking.

The key words or terms are colored in blue. The set of terms is called Dictionary. Counting the frequency of terms in each document we obtain a term-document matrix.

Term-document matrix

Term	Doc. 1	Doc. 2	Doc. 3	Doc. 4	Doc. 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
web	0	1	1	0	1

Term-document matrix

Term	Doc. 1	Doc. 2	Doc. 3	Doc. 4	Doc. 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
web	0	1	1	0	1

Each document is represented by a column of the term-document matrix $A \in \mathbb{R}^{10\times 5}$ which is a vector in $\mathbb{R}^{10}$.

Query vector

Suppose that we want to find all documents that are relevant to the query ranking of web pages. This is represented by a query vector, constructed in the same way as the term-document matrix, using the same dictionary:

$$\mathbf{v} := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \in \mathbb{R}^{10}.$$

Mathematical Problem

Thus the query itself is a document. The information retrieval task can now be formulated as a mathematical problem.

Mathematical Problem

Thus the query itself is a document. The information retrieval task can now be formulated as a mathematical problem.

Problem: Find the columns of A that are close to the query vector $\mathbf{v}$.

Mathematical Problem

Thus the query itself is a document. The information retrieval task can now be formulated as a mathematical problem.

> Problem: Find the columns of A that are close to the query vector $\mathbf{v}$.

The $m \times n$ term-document matrix in information retrieval is usually a tall matrix with $m = \mathcal{O}(10^6)$. As most of the documents only contain a small fraction of the terms in the dictionary, the matrix is sparse.

Mathematical Problem

Thus the query itself is a document. The information retrieval task can now be formulated as a mathematical problem.

> Problem: Find the columns of A that are close to the query vector $\mathbf{v}$.

The $m \times n$ term-document matrix in information retrieval is usually a tall matrix with $m = \mathcal{O}(10^6)$. As most of the documents only contain a small fraction of the terms in the dictionary, the matrix is sparse.

The task is to extract useful information from large and often unstructured sets of data. Hence the methods used must be efficient and specially designed for large problems.

Query matching

Query matching is the process of finding all documents that are relevant to a particular query $\mathbf{v}$. The cosine distance measure is often used to return relevant documents:

$$\cos\theta_j := \frac{\mathbf{v}^T A e_j}{\|\mathbf{v}\|_2 \|A e_j\|_2} > \mathrm{tol}$$

where tol is a predefined tolerance.

Query matching

Query matching is the process of finding all documents that are relevant to a particular query $\mathbf{v}$. The cosine distance measure is often used to return relevant documents:

$$\cos\theta_j := \frac{\mathbf{v}^T A e_j}{\|\mathbf{v}\|_2 \|A e_j\|_2} > \mathrm{tol}$$

where tol is a predefined tolerance.

Low tolerance $\Longrightarrow$ more documents that are relevant to the query are returned.

Query matching

Query matching is the process of finding all documents that are relevant to a particular query $\mathbf{v}$. The cosine distance measure is often used to return relevant documents:

$$\cos\theta_j := \frac{\mathbf{v}^T A e_j}{\|\mathbf{v}\|_2 \|A e_j\|_2} > \mathrm{tol}$$

where tol is a predefined tolerance.

Low tolerance $\Longrightarrow$ more documents that are relevant to the query are returned.

But at the same time there is a risk that more documents that are not relevant are also returned.

Latent semantic indexing

Latent semantic indexing (LSI) is based on the assumptions:

- There is some underlying latent semantic structure in the data that is corrupted by the wide variety of words used.

Latent semantic indexing

Latent semantic indexing (LSI) is based on the assumptions:

- There is some underlying latent semantic structure in the data that is corrupted by the wide variety of words used.
- The semantic structure can be enhanced by projecting the data (A and $\mathbf{v}$) onto a lower-dimensional space using the singular value decomposition.

Latent semantic indexing

Latent semantic indexing (LSI) is based on the assumptions:

- There is some underlying latent semantic structure in the data that is corrupted by the wide variety of words used.
- The semantic structure can be enhanced by projecting the data (A and $\mathbf{v}$) onto a lower-dimensional space using the singular value decomposition.

Consider SVD $A = U\Sigma V^T$ and the rank-k approximation $A_k := U_k \Sigma_k V_k^T$ so that $\|A - A_k\|_F / \|A\|_F$ small.

Latent semantic indexing

Latent semantic indexing (LSI) is based on the assumptions:

- There is some underlying latent semantic structure in the data that is corrupted by the wide variety of words used.
- The semantic structure can be enhanced by projecting the data (A and $\mathbf{v}$) onto a lower-dimensional space using the singular value decomposition.

Consider SVD $A = U\Sigma V^T$ and the rank-k approximation $A_k := U_k \Sigma_k V_k^T$ so that $\|A - A_k\|_F / \|A\|_F$ small.

Set $\mathbf{v}_k := U_k^T \mathbf{v}$ and $D_k := \Sigma_k V_k^T$. Compute the cosines

$$\cos\theta_j := \frac{\mathbf{v}_k^T D_k(:, j)}{\|\mathbf{v}_k\|_2 \|D_k(; , j)\|_2}.$$

Example

Consider the term-document matrix A and the query ("ranking of web pages") vector $\mathbf{v}$. Then the cosines measures of the query and the original data are given by

$$[0, 0.6667, 0.7746, 0.3333, 0.3333]^T$$

which shows that Doc. 2-3 are relevant and Doc. 1 is totally irrelevant.

Example

Consider the term-document matrix A and the query ("ranking of web pages") vector $\mathbf{v}$. Then the cosines measures of the query and the original data are given by

$$[0,\ 0.6667,\ 0.7746,\ 0.3333,\ 0.3333]^T$$

which shows that Doc. 2-3 are relevant and Doc. 1 is totally irrelevant.

Now consider the LSI method with rank-2 approximation. The cosines measures of the projected query and the projected data are given by

$$[0.7857,\ 0.8332,\ 0.9670,\ 0.4873,\ 0.1819]^T.$$

Note that Doc. 1 is deemed highly relevant!

Clustering

Clustering is a powerful tool in data mining that sorts data into groups called clusters. Clustering is also used in pattern recognition.

Clustering

Clustering is a powerful tool in data mining that sorts data into groups called clusters. Clustering is also used in pattern recognition.

There are several methods for clustering data. The choice of clustering method is influenced by the type of questions asked and data analyzed.

Clustering

Clustering is a powerful tool in data mining that sorts data into groups called clusters. Clustering is also used in pattern recognition.

There are several methods for clustering data. The choice of clustering method is influenced by the type of questions asked and data analyzed.

The singular value decomposition (SVD) of term-document matrix also provides a clustering algorithm.

Clustering

Clustering is a powerful tool in data mining that sorts data into groups called clusters. Clustering is also used in pattern recognition.

There are several methods for clustering data. The choice of clustering method is influenced by the type of questions asked and data analyzed.

The singular value decomposition (SVD) of term-document matrix also provides a clustering algorithm.

Idea: Given a column partitioning of a term-document matrix A into k clusters

$$A := [A_1, \ldots, A_k] \text{ with } A_j \in \mathbb{R}^{m \times n_j},$$

compute the centroid $\mathbf{c}_j := \frac{1}{n_j} A_j e^{(j)}$ and $C := [\mathbf{c}_1, \ldots, \mathbf{c}_k]$, where $e^{(j)} := [1, 1, \cdots, 1]^T \in \mathbb{R}^{n_j}$.

Classification of handwritten digits

Question: Can we create an algorithm to automatically recognize handwritten digits?

Classification of handwritten digits

Question: Can we create an algorithm to automatically recognize handwritten digits?

Computer classification of handwritten digits is a standard problem in pattern recognition. A typical application is automatic reading of pin codes on envelopes.

Classification of handwritten digits

Question: Can we create an algorithm to automatically recognize handwritten digits?

Computer classification of handwritten digits is a standard problem in pattern recognition. A typical application is automatic reading of pin codes on envelopes.

The image of a handwritten digit can be treated as 16×16 matrix representing grey-scale. It can also be represented as a vector in $\mathbb{R}^{256}$ by stacking the columns of the matrix.

Classification of handwritten digits

Question: Can we create an algorithm to automatically recognize handwritten digits?

Computer classification of handwritten digits is a standard problem in pattern recognition. A typical application is automatic reading of pin codes on envelopes.

The image of a handwritten digit can be treated as 16×16 matrix representing grey-scale. It can also be represented as a vector in $\mathbb{R}^{256}$ by stacking the columns of the matrix.

Problem: Given a set of of manually classified digits (the training set), classify a set of unknown digits (the test set).

Classification of handwritten digits

Question: Can we create an algorithm to automatically recognize handwritten digits?

Computer classification of handwritten digits is a standard problem in pattern recognition. A typical application is automatic reading of pin codes on envelopes.

The image of a handwritten digit can be treated as 16×16 matrix representing grey-scale. It can also be represented as a vector in $\mathbb{R}^{256}$ by stacking the columns of the matrix.

Problem: Given a set of of manually classified digits (the training set), classify a set of unknown digits (the test set).

Example: Training set of US Postal Service database contains 7291 handwritten digits and test set contains 2007 digits.

Classification of handwritten digits

Considering the training set digits as vectors, it is reasonable to assume that all digits of one kind form a cluster of points in $\mathbb{R}^{256}$.

Classification of handwritten digits

Considering the training set digits as vectors, it is reasonable to assume that all digits of one kind form a cluster of points in $\mathbb{R}^{256}$.

Ideally the clusters would be well separated and the separation would depend on how well written the training digits are.

Classification of handwritten digits

Considering the training set digits as vectors, it is reasonable to assume that all digits of one kind form a cluster of points in $\mathbb{R}^{256}$.

Ideally the clusters would be well separated and the separation would depend on how well written the training digits are.

Example:

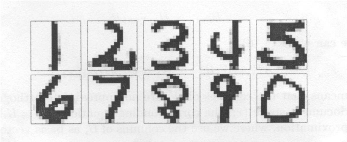


Figure: Handwritten digits from the US Postal service database.

Classification of handwritten digits

The mean digits (centroids) of the training set are given below.

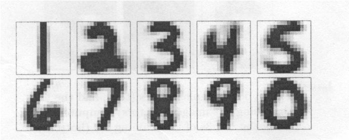


Figure: The mean digits of all digits in the training set.

Classification of handwritten digits

The mean digits (centroids) of the training set are given below.

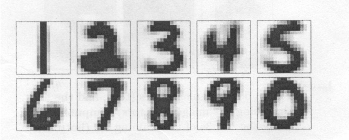


Figure: The mean digits of all digits in the training set.

It follows that a majority of the digits are well written (badly written digits would demonstrate as diffuse means).

Classification of handwritten digits

The mean digits (centroids) of the training set are given below.

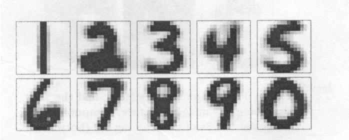


Figure: The mean digits of all digits in the training set.

It follows that a majority of the digits are well written (badly written digits would demonstrate as diffuse means).

Hence a simple algorithm that computes the distance from each unknown digit to the mean digits is likely to work well.

Classification of handwritten digits

A simple classification algorithm

- Training. Given the training set, compute the mean digit (centroid) of all digits of one kind.
- Classification. For each digit in the test set, compute the distance to all ten mean digits, and classify as the closest.

Classification of handwritten digits

A simple classification algorithm

- Training. Given the training set, compute the mean digit (centroid) of all digits of one kind.
- Classification. For each digit in the test set, compute the distance to all ten mean digits, and classify as the closest.

The success rate of this algorithm for the training set (US Postal Service) is around 75%, which is not good enough.

Classification of handwritten digits

A simple classification algorithm

- Training. Given the training set, compute the mean digit (centroid) of all digits of one kind.
- Classification. For each digit in the test set, compute the distance to all ten mean digits, and classify as the closest.

The success rate of this algorithm for the training set (US Postal Service) is around 75%, which is not good enough.

The reason is that the algorithm does not use any information about the variation of the digits of one kind. This variation can be modelled using the SVD.

SVD based classification of handwritten digits

Let $A \in \mathbb{R}^{m \times n}$, $m = 256$, be the matrix consisting of all the training digits of one kind, say 3. The column space $R(A)$ cannot be expected to have a large dimension because the subspaces of the different kinds of digits would intersect.

SVD based classification of handwritten digits

Let $A \in \mathbb{R}^{m \times n}$, $m = 256$, be the matrix consisting of all the training digits of one kind, say 3. The column space $R(A)$ cannot be expected to have a large dimension because the subspaces of the different kinds of digits would intersect.

The idea is to model the variation within the set of training digits of one kind using an orthonormal basis of $R(A)$.

SVD based classification of handwritten digits

Let $A \in \mathbb{R}^{m \times n}$, $m = 256$, be the matrix consisting of all the training digits of one kind, say 3. The column space $R(A)$ cannot be expected to have a large dimension because the subspaces of the different kinds of digits would intersect.

The idea is to model the variation within the set of training digits of one kind using an orthonormal basis of $R(A)$.

An orthonormal basis of $R(A)$ can be computed using the SVD

$$A = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T.$$

Also A can be approximated by a low rank matrix.

SVD based classification of handwritten digits

Each column in A is an image of the digit 3. The left singular vectors $\mathbf{u}_j$'s are an orthogonal basis of $R(A)$, the image space of 3.

SVD based classification of handwritten digits

Each column in A is an image of the digit 3. The left singular vectors $\mathbf{u}_j$'s are an orthogonal basis of $R(A)$, the image space of 3.

The left singular vectors are referred to as singular images. The first singular vector represents the dominating direction of the data matrix.

SVD based classification of handwritten digits

Each column in A is an image of the digit 3. The left singular vectors $\mathbf{u}_j$'s are an orthogonal basis of $R(A)$, the image space of 3.

The left singular vectors are referred to as singular images. The first singular vector represents the dominating direction of the data matrix.

We expect that the first singular image (image corresponding to the first singular vector) to look like 3 and the remaining singular images should represent the dominating variations of the training set around the first singular image.

SVD based classification of handwritten digits

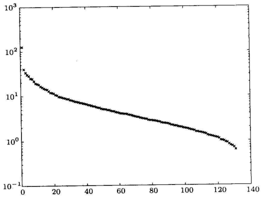


Figure: Singular values computed using 131 images of 3 from training set.

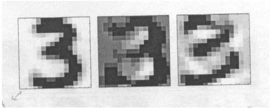


Figure: The first three singular images of 3.

SVD based classification of handwritten digits

We compute how well an unknown digit can be represented in the ten different singular bases. This can be done by computing the residual vector in least squares problems:

SVD based classification of handwritten digits

We compute how well an unknown digit can be represented in the ten different singular bases. This can be done by computing the residual vector in least squares problems:

Let U_k be k-singular basis. Given a test digit $\mathbf{t}$

$$\text{solve } \mathbf{t}_k := \operatorname{argmin}_{\mathbf{x}} \|\mathbf{t} - U_k\mathbf{x}\|_2$$

and compute the residual

$$r_k := \min_{\mathbf{x}} \|\mathbf{t} - U_k\mathbf{x}\|_2$$

for the 10 digits $k = 0, 1, \ldots, 9$.

SVD based classification of handwritten digits

We compute how well an unknown digit can be represented in the ten different singular bases. This can be done by computing the residual vector in least squares problems:

Let U_k be k-singular basis. Given a test digit $\mathbf{t}$

$$\text{solve } \mathbf{t}_k := \operatorname{argmin}_{\mathbf{x}} \|\mathbf{t} - U_k \mathbf{x}\|_2$$

and compute the residual

$$r_k := \min_{\mathbf{x}} \|\mathbf{t} - U_k \mathbf{x}\|_2$$

for the 10 digits $k = 0, 1, \ldots, 9$.

Then $\mathbf{t}_k = U_k^T \mathbf{t}$ and $r_k = \|(I - U_k U_k^T)\mathbf{t})$.
r_k depends on the number of basis elements in k-singular basis.

Example

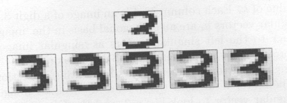


Figure: Unknown digit (nice 3) and 1, 3, 5, 7, and 9 basis elements in 3-basis.

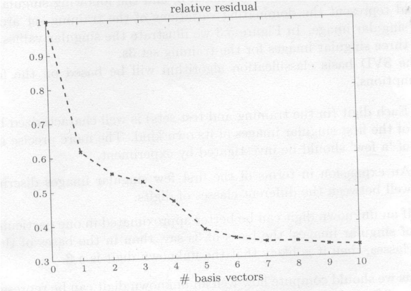


Figure: Relative residuals: $r_k/\|\mathbf{t}\|_2$.

Example

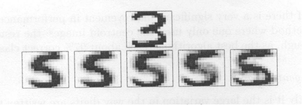


Figure: Unknown digit (nice 3) and $1, 3, 5, 7$, and 9 basis elements in 5-basis.

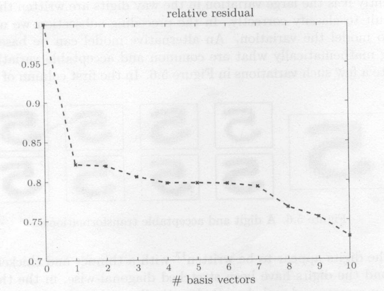


Figure: Relative residuals $r_k/\|\mathbf{t}\|_2$.

SVD based classification of handwritten digits

An SVD based classification algorithm

- Training: For the training set of known digits, compute the SVD of each class of digits and use k basis vectors for each class.
- Classification: For a given test digit, compute its relative residual in all ten bases. If one residual is significantly smaller than all the others, classify as that. Else give up.

SVD based classification of handwritten digits

An SVD based classification algorithm

- Training: For the training set of known digits, compute the SVD of each class of digits and use k basis vectors for each class.
- Classification: For a given test digit, compute its relative residual in all ten bases. If one residual is significantly smaller than all the others, classify as that. Else give up.

Test results for the US Postal Service data set with 7291 training digits and 2007 test digits. Result for each class.

# basis vector	1	2	4	6	8	10
correct (%)	80	86	90	90.5	92	93

Face recognition

Figure: Library of twelve Presidents of the United States.

We store these images as vectors or matrices and apply PCA (Principal Component Analysis) to process the images.

Face recognition

Figure: Library of twelve Presidents of the United States.

We store these images as vectors or matrices and apply PCA (Principal Component Analysis) to process the images.

Face recognition

PCA uses eigenvectors (eigenfaces) to recapture an approximation to the original faces. Usually 50 or 100 eigenfaces are required to create strong approximations to a set of $10,000$ faces.

Face recognition

PCA uses eigenvectors (eigenfaces) to recapture an approximation to the original faces. Usually 50 or 100 eigenfaces are required to create strong approximations to a set of $10,000$ faces.

Assume that each image has the same pixel resolution with m rows and n columns. Then, converting each picture to a vector, we have an $mn \times 12$ matrix P.

Face recognition

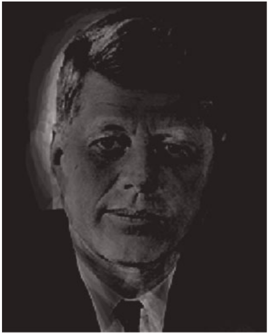
PCA uses eigenvectors (eigenfaces) to recapture an approximation to the original faces. Usually 50 or 100 eigenfaces are required to create strong approximations to a set of $10,000$ faces.

Assume that each image has the same pixel resolution with m rows and n columns. Then, converting each picture to a vector, we have an $mn \times 12$ matrix P.

Consider the average image $\mathbf{p} := Ae/12$, where $e := [1, 1, \ldots, 1]^T \in \mathbb{R}^{12}$. Then $A := P - \mathbf{p}e^T$ is the mean-subtracted image and $C := AA^T$ is the covariance matrix.

Face recognition

(a) label 1



(b) label 2

Figure: (a) The average image of twelve U.S. Presidents and (b) the average image subtracted from the image of President Kennedy .

Face recognition

For the presidential library images, C is a 50000×50000 matrix. Finding eigenvectors would be prohibitively time-consuming even for a supercomputer.

Face recognition

For the presidential library images, C is a 50000×50000 matrix. Finding eigenvectors would be prohibitively time-consuming even for a supercomputer.

However, when there are N images in the library, there will be N linearly independent eigenvectors of C.

Face recognition

For the presidential library images, C is a 50000×50000 matrix. Finding eigenvectors would be prohibitively time-consuming even for a supercomputer.

However, when there are N images in the library, there will be N linearly independent eigenvectors of C.

Since $A^T A$ and C have the same nonzero eigenvalues, we solve $A^T Au = \lambda u$. If $\lambda \neq 0$ then $AA^T(Au) = \lambda(Au)$ and $Au \neq 0$. Hence $v := Au$ is an eigenvector of C corresponding to λ.

Face recognition

For the presidential library images, C is a 50000×50000 matrix. Finding eigenvectors would be prohibitively time-consuming even for a supercomputer.

However, when there are N images in the library, there will be N linearly independent eigenvectors of C.

Since $A^T A$ and C have the same nonzero eigenvalues, we solve $A^T Au = \lambda u$. If $\lambda \neq 0$ then $AA^T(Au) = \lambda(Au)$ and $Au \neq 0$. Hence $v := Au$ is an eigenvector of C corresponding to λ.

Eigenvectors of C are eigenfaces. These eigenfaces are obtained from eigenvectors of a 12×12 matrix $A^T A$.

Face recognition

Figure: A half dozen eigenfaces of the library of twelve U.S. Presidents.

Nonzero singular values of A are square roots of the nonzero eigenvalues of $A^T A$. The singular values have tendency to drop off in value quickly.

Face recognition

Figure: A half dozen eigenfaces of the library of twelve U.S. Presidents.

Nonzero singular values of A are square roots of the nonzero eigenvalues of $A^T A$. The singular values have tendency to drop off in value quickly.

This allows us to approximate a large collection of images, like 5,000, with only a small subset of eigenfaces.

Face recognition

The eigenfaces can be used to recognize a new face as follows. Subtract the average presidential image from the new image.

Face recognition

The eigenfaces can be used to recognize a new face as follows. Subtract the average presidential image from the new image.

Project the mean-subtracted image $\mathbf{v}$ on the space spanned by the eigenfaces $\mathbf{u}_j$. .

$$\mathbf{v}_{proj} := \sum_{j=1}^{12} (\mathbf{u}_j^T \mathbf{v})\mathbf{u}_j.$$

Then adding the average presidential image we obtain a recognizable face (or an approximation) $\mathbf{v}_{proj} + \mathbf{p}$.

Face recognition

(a) label 1

(b) label 2

Figure: (a) An altered image of President Kennedy and (b) the image recreated using the six eigenfaces.

Face recognition

This is part of the power of PCA in facial recognition.

Face recognition

This is part of the power of PCA in facial recognition.

One can use a large library of images, reduce it to a much smaller set of eigenfaces, and then use it to recognize a face or create an approximation, even with some disguising.

***