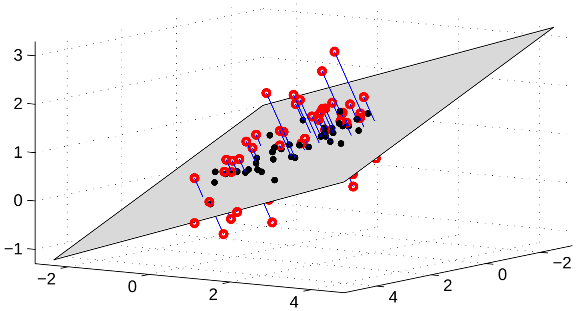
Unsupervised Dimension Reduction

Prof. Dr. H. H. Takada

Quantitative Research – Itaú Asset Management
Institute of Mathematics and Statistics – University of São Paulo

High-Dimensional Spaces – Low Dimensional Manifolds

Data is often high dimensional – images, bag-of-word descriptions, gene-expressions *etc.* Provided there is some ‘structure’, data will typically lie close to a much lower dimensional ‘manifold’. Here we concentrate on computationally efficient linear dimension reduction techniques.



Principal Components Analysis

We express the approximation for datapoint $\mathbf{x}^n$ as

$$\mathbf{x}^n \approx \mathbf{c} + \sum_{j=1}^{M} y_j^n \mathbf{b}^j \equiv \tilde{\mathbf{x}}^n$$

Here the vector $\mathbf{c}$ is a constant and defines a point in the subspace.

Basis

The $\mathbf{b}^j$ are 'basis' vectors that span the subspace. Collectively we can write $\mathbf{B} = \left[\mathbf{b}^1, \ldots, \mathbf{b}^M\right]$.

Low dimensional coordinates

The y_i^n are the low dimensional coordinates of the data.

Minimal square loss approximation

To determine the best lower dimensional representation it is convenient to use the square distance error between $\mathbf{x}$ and its reconstruction $\tilde{\mathbf{x}}$:

$$E(\mathbf{B}, \mathbf{Y}, \mathbf{c}) = \sum_{n=1}^{N} \sum_{i=1}^{D} [x_i^n - \tilde{x}_i^n]^2$$

Centering

The optimal bias $\mathbf{c}$ is given by the mean of the data $\sum_n \mathbf{x}^n / N$. We therefore assume that the data has been centered (has zero mean $\sum_n \mathbf{x}^n = \mathbf{0}$), so that we can set $\mathbf{c}$ to zero, and concentrate on finding the optimal basis $\mathbf{B}$.

Minimal square loss approximation

We wish to minimize the sum of squared differences between each vector $\mathbf{x}$ and its reconstruction $\tilde{\mathbf{x}}$:

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \left[x_i^n - \sum_{j=1}^{M} y_j^n b_i^j \right]^2 = \mathrm{trace}\left((\mathbf{X} - \mathbf{B}\mathbf{Y})^\mathsf{T} (\mathbf{X} - \mathbf{B}\mathbf{Y}) \right)$$

where $\mathbf{X} = \left[\mathbf{x}^1, \ldots, \mathbf{x}^N\right]$.

Orthonormality constraint

Consider an invertible transformation $\mathbf{Q}$ of the basis $\mathbf{B}$ so that $\tilde{\mathbf{B}} \equiv \mathbf{B}\mathbf{Q}$ is an orthonormal matrix, $\tilde{\mathbf{B}}^\mathsf{T}\tilde{\mathbf{B}} = \mathbf{I}$. Since $\mathbf{Q}$ is invertible, we may write $\mathbf{B}\mathbf{Y} = \tilde{\mathbf{B}}\mathbf{Q}^{-1}\mathbf{Y} \equiv \tilde{\mathbf{B}}\tilde{\mathbf{Y}}$, which is of the same form as $\mathbf{B}\mathbf{Y}$, albeit with an orthonormality constraint on $\tilde{\mathbf{B}}$. Hence, without loss of generality, we may impose the orthonormality constraint $\mathbf{B}^\mathsf{T}\mathbf{B} = \mathbf{I}$, namely that the basis vectors are mutually orthogonal and of unit length.

Finding the optimal $\mathbf{Y}$

$$-\frac{1}{2}\frac{\partial}{\partial y_k^n}E(\mathbf{B},\mathbf{Y}) = \sum_i \left[x_i^n - \sum_j y_j^n b_i^j\right] b_i^k = \sum_i x_i^n b_i^k - \sum_j y_j^n \underbrace{\sum_i b_i^j b_i^k}_{\delta_{jk}}$$

The squared error $E(\mathbf{B},\mathbf{Y})$ therefore has zero derivative when

$$y_k^n = \sum_i b_i^k x_i^n, \qquad \text{which can be written as} \quad \mathbf{Y} = \mathbf{B}^\mathsf{T}\mathbf{X}$$

The objective $E(\mathbf{B})$ becomes

$$E(\mathbf{B}) = \mathrm{trace}\left((\mathbf{X}-\mathbf{B}\mathbf{Y})^\mathsf{T}(\mathbf{X}-\mathbf{B}\mathbf{Y})\right) = \mathrm{trace}\left(\mathbf{X}^\mathsf{T}\left(\mathbf{I}-\mathbf{B}\mathbf{B}^\mathsf{T}\right)^2\mathbf{X}\right)$$

Since $\left(\mathbf{I}-\mathbf{B}\mathbf{B}^\mathsf{T}\right)^2 = \mathbf{I}-\mathbf{B}\mathbf{B}^\mathsf{T}$, (using $\mathbf{B}^\mathsf{T}\mathbf{B} = \mathbf{I}$)

$$E(\mathbf{B}) = \mathrm{trace}\left(\mathbf{X}\mathbf{X}^\mathsf{T}\left(\mathbf{I}-\mathbf{B}\mathbf{B}^\mathsf{T}\right)\right)$$

The role of the sample covariance

Hence the objective becomes

$$E(\mathbf{B}) = (N-1)\left[\operatorname{trace}(\mathbf{S}) - \operatorname{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^\mathsf{T}\right)\right]$$

where $\mathbf{S}$ is the sample covariance matrix of the centred data. Since we assumed the data is zero mean, this is

$$\mathbf{S} = \frac{1}{N-1}\sum_{n=1}^{N} \mathbf{x}^n (\mathbf{x}^n)^\mathsf{T} = \frac{1}{N-1}\mathbf{X}\mathbf{X}^\mathsf{T}$$

General case

If we hadn't centered the data, we would have

$$\mathbf{S} = \frac{1}{N-1}\sum_{n=1}^{N} (\mathbf{x}^n - \mathbf{m})(\mathbf{x}^n - \mathbf{m})^\mathsf{T}$$

where $\mathbf{m}$ is the sample mean of the data.

Using Lagrange to enforce orthonormality

To minimize $E(\mathbf{B})$ under the constraint $\mathbf{B}^{\mathsf{T}}\mathbf{B} = \mathbf{I}$ we use a set of Lagrange multipliers $\mathbf{L}$, so that the objective is to minimize

$$-\mathrm{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^{\mathsf{T}}\right) + \mathrm{trace}\left(\mathbf{L}\left(\mathbf{B}^{\mathsf{T}}\mathbf{B} - \mathbf{I}\right)\right)$$

Since the constraint is symmetric, we can assume that $\mathbf{L}$ is also symmetric. Differentiating with respect to $\mathbf{B}$ and equating to zero we obtain that at the optimum

$$\mathbf{S}\mathbf{B} = \mathbf{B}\mathbf{L}$$

We need to find matrices $\mathbf{B}$ and $\mathbf{L}$ that satisfy this equation. One solution is given when $\mathbf{L}$ is diagonal in which case this is a form of eigen-equation and the columns of $\mathbf{B}$ are the corresponding eigenvectors of $\mathbf{S}$.

In this case, $\mathrm{trace}\left(\mathbf{S}\mathbf{B}\mathbf{B}^{\mathsf{T}}\right) = \mathrm{trace}\left(\mathbf{L}\right)$, which is the sum of the eigenvalues corresponding to the eigenvectors forming $\mathbf{B}$.

$$\frac{1}{N-1}E(\mathbf{B}) = -\mathrm{trace}\left(\mathbf{L}\right) + \mathrm{trace}\left(\mathbf{S}\right) = -\sum_{i=1}^{M} \lambda_i + \text{const.}$$

Since we wish to minimize $E(\mathbf{B})$, we therefore define the basis using the eigenvectors with largest corresponding eigenvalues.

Residual error

If we order the eigenvalues $\lambda_1 \geq \lambda_2, \ldots$, the squared error is then given by

$$\frac{1}{N-1}E(\mathbf{B}) = \mathrm{trace}\left(\mathbf{S}\right) - \mathrm{trace}\left(\mathbf{L}\right) = \sum_{i=1}^{D} \lambda_i - \sum_{i=1}^{M} \lambda_i = \sum_{i=M+1}^{D} \lambda_i$$

Hence the sum of the neglected eigenvalues equals the squared reconstruction error.

Uniqueness

Whilst the solution to this eigen-problem is unique, this only serves to define the solution subspace since one may rotate and scale $\mathbf{B}$ and $\mathbf{Y}$ such that the value of the squared loss is exactly the same

Maximum variance criterion

To break the invariance of least squares projection with respect to rotations and rescaling, we need an additional criterion. One such is given by first searching for the single direction $\mathbf{b}$ such that the variance of the data projected onto this direction is maximal amongst all possible such projections. For a single vector $\mathbf{b}$ we have

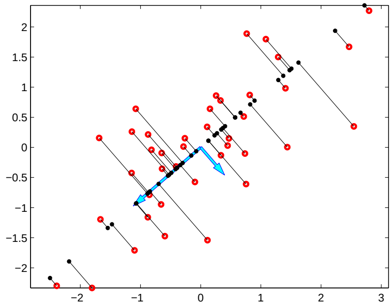
$$y^n = \sum_i b_i x_i^n$$

The projection of a datapoint onto a direction $\mathbf{b}$ is $\mathbf{b}^\mathsf{T}\mathbf{x}^n$ for a unit length vector $\mathbf{b}$. Hence the sum of squared projections is

$$\sum_n \left(\mathbf{b}^\mathsf{T}\mathbf{x}^n\right)^2 = \mathbf{b}^\mathsf{T}\left[\sum_n \mathbf{x}^n\left(\mathbf{x}^n\right)^\mathsf{T}\right]\mathbf{b} = (N-1)\mathbf{b}^\mathsf{T}\mathbf{S}\mathbf{b}$$

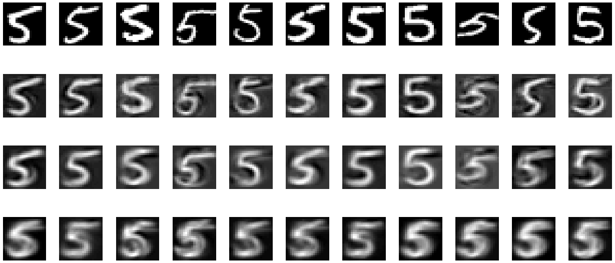
The optimal single $\mathbf{b}$ which maximizes the projection variance is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}$. Under the criterion that the next optimal direction $\mathbf{b}^{(2)}$ should be orthonormal to the first, one can readily show that $\mathbf{b}^{(2)}$ is given by the second largest eigenvector, and so on. These maximal variance directions found by PCA are called the principal directions.

PCA



The original datapoints $\mathbf{x}$ (larger rings) and their reconstructions $\tilde{\mathbf{x}}$ (small dots) using 1 dimensional PCA. The lines represent the orthogonal projection of the original datapoint onto the first eigenvector. The arrows are the two eigenvectors scaled by the square root of their corresponding eigenvalues. The data has been centered to have zero mean. For each 'high dimensional' datapoint $\mathbf{x}$, the 'low dimensional' representation $\mathbf{y}$ is given in this case by the distance (possibly negative) from the origin along the first eigenvector direction to the corresponding orthogonal projection point.

Reducing the dimension of digits



Top row: a selection of the digit 5 taken from the database of 892 examples. Plotted beneath each digit is the reconstruction using 100, 30 and 5 eigenvectors (from top to bottom). Note how the reconstructions for fewer eigenvectors express less variability from each other, and resemble more a mean 5 digit.

PCA via Singular Value Decomposition

Consider the SVD decomposition of the data matrix:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$$

where $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}_D$ and $\mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I}_N$ and $\mathbf{D}$ is a diagonal matrix of the (positive) singular values. We assume that the decomposition has ordered the singular values so that the upper left diagonal element of $\mathbf{D}$ contains the largest singular value. The matrix $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ can then be written as

$$\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}\mathbf{V}\mathbf{D}\mathbf{U}^{\mathsf{T}} = \mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathsf{T}}$$

Since $\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathsf{T}}$ is in the form of an eigen-decomposition, the PCA solution is equivalently given by performing the SVD decomposition of $\mathbf{X}$, for which the eigenvectors are then given by $\mathbf{U}$, and corresponding eigenvalues by the square of the singular values.

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}} \approx \mathbf{U}_M\mathbf{D}_M\mathbf{V}_M^{\mathsf{T}}$$

where $\mathbf{U}_M$, $\mathbf{D}_M$, $\mathbf{V}_M$ correspond to taking only the first M singular values of the full matrices.

Eigenfaces

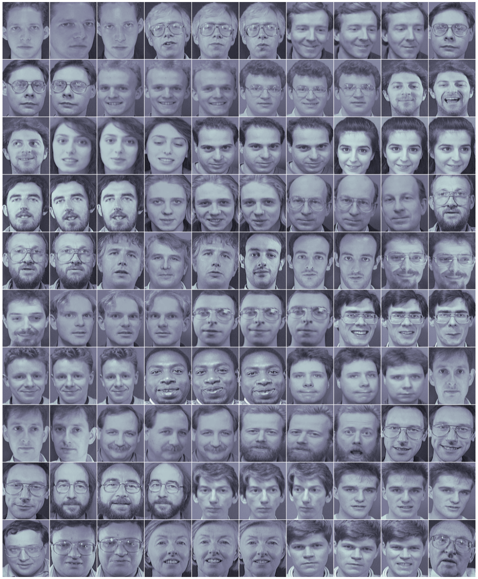
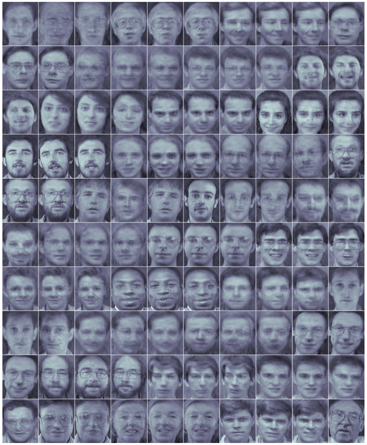


Figure: 100 of the 120 training images (40 people, with 3 images of each person). Each image consists of $92 \times 112 = 10304$ non-negative greyscale pixels. The data is scaled so that, represented as an image, the components of each image sum to 1. The average value of each pixel across all images is 9.70×10^{-5}. This is a subset of the 400 images in the full Olivetti Research Face Database

Eigenfaces


(a)

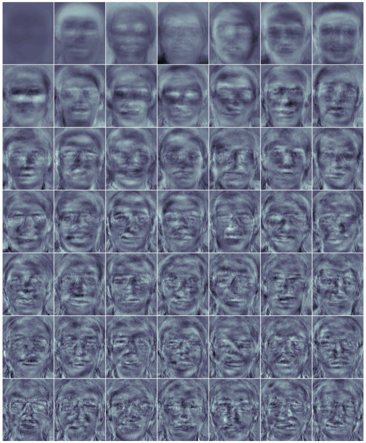

(b)

Figure: **(a)**: SVD reconstruction of the images using a combination of the 49 eigen-images. **(b)**: The eigen-images are found using SVD of the above data and taking the 49 eigenvectors with largest eigenvalues. The images corresponding to the largest eigenvalues are contained in the first row, and the next 7 in the row below, *etc.*

PCA With Missing Data

Unfortunately, there is no ‘quick fix’ PCA solution when some of the x_i^n are missing. A naive approach is to require the squared reconstruction error to be small only for the existing elements of $\mathbf{X}$. That is

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_i^j \right]^2$$

where $\gamma_i^n = 1$ if the i^{th} entry of the n^{th} vector is available, and is zero otherwise. The optimal weights satisfy (assuming $\mathbf{B}^\mathsf{T}\mathbf{B} = \mathbf{I}$),

$$y_n^k = \sum_i \gamma_i^n x_i^n b_i^k$$

One then substitutes this expression into the squared error, and minimizes the error with respect to $\mathbf{B}$ under the orthonormality constraint. An alternative iterative optimization procedure is as follows: First select a random $D \times M$ matrix $\hat{\mathbf{B}}$. Then iterate until convergence the following two steps:

Optimize $\mathbf{Y}$ for fixed $\mathbf{B}$

$$E(\hat{\mathbf{B}}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n \hat{b}_i^j \right]^2$$

For fixed $\hat{\mathbf{B}}$ the above $E(\hat{\mathbf{B}}, \mathbf{Y})$ is a quadratic function of the matrix $\mathbf{Y}$, which can be optimized directly. By differentiating and equating to zero, one obtains the fixed point condition

$$\sum_i \gamma_i^n \left(x_i^n - \sum_l y_l^n \hat{b}_i^l \right) \hat{b}_i^k = 0$$

Defining

$$\left[\mathbf{y}^{(n)}\right]_l = y_n^l, \qquad \left[\mathbf{M}^{(n)}\right]_{kl} = \sum_i \hat{b}_i^l \hat{b}_i^k \gamma_i^n, \qquad [\mathbf{c}^n]_k = \sum_i \gamma_i^n x_i^n \hat{b}_i^k,$$

in matrix notation, we then have a set of linear systems:

$$\mathbf{c}^{(n)} = \mathbf{M}^{(n)} \mathbf{y}^{(n)}, \qquad n = 1, \dots, N$$

Optimize $\mathbf{B}$ for fixed $\mathbf{Y}$

One now freezes $\hat{\mathbf{Y}}$ and considers the function

$$E(\mathbf{B},\hat{\mathbf{Y}}) = \sum_{n=1}^{N}\sum_{i=1}^{D}\gamma_i^n\left[x_i^n - \sum_j \hat{y}_j^n b_i^j\right]^2$$

For fixed $\hat{\mathbf{Y}}$ the above expression is quadratic in the matrix $\mathbf{B}$, which can again be optimized using linear algebra. This corresponds to solving a set of linear systems for the i^{th} row of $\mathbf{B}$:

$$\mathbf{m}^{(i)} = \mathbf{F}^{(i)}\mathbf{b}^{(i)}$$

where

$$\left[\mathbf{m}^{(i)}\right]_k = \sum_n \gamma_i^n x_i^n \hat{y}_k^n, \qquad \left[\mathbf{F}^{(i)}\right]_{kj} = \sum_n \gamma_i^n \hat{y}_j^n \hat{y}_k^n$$

Mathematically, this is $\mathbf{b}^{(i)} = {\mathbf{F}^{(i)}}^{-1}\mathbf{m}^{(i)}$. In this manner one is guaranteed to iteratively decrease the value of the squared error loss until a minimum is reached.

PCA With Missing Data: matrix completion

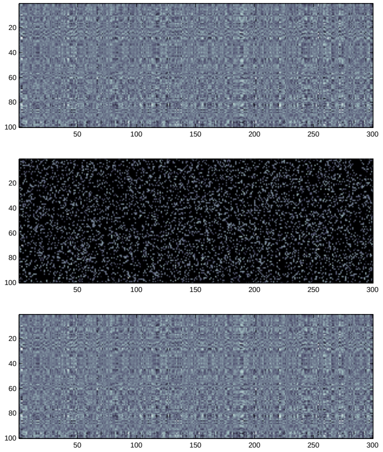


Figure: Top: original data matrix $\mathbf{X}$. The data is constructed from a set of only 5 basis vectors. Middle: $\mathbf{X}$ with missing data (black)(80% sparsity). Bottom : reconstruction found using `svdm.m`, SVD for missing data. This problem is essentially easy since, despite there being many missing elements, the data is indeed constructed from a model for which SVD is appropriate. Such techniques have application in collaborative filtering and recommender systems where one wishes to 'fill in' missing values in a matrix.

Matlab

```
setup; demoSVDmissing;
```

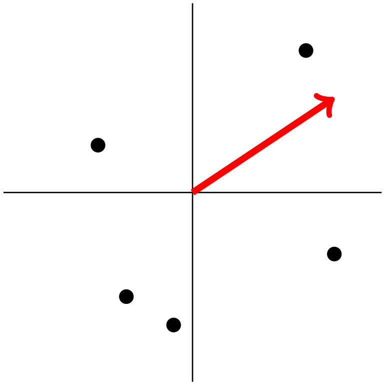
Matrix Decompositions

Given a data matrix $\mathbf{X}$ for which each column represents a datapoint, an approximate matrix decomposition is of the form $\mathbf{X} \approx \mathbf{BY}$ into a basis matrix $\mathbf{B}$ and weight (or coordinate) matrix $\mathbf{Y}$. Symbolically, matrix decompositions are of the form

$$\underbrace{\left(\qquad X \ : \text{Data} \qquad \right)}_{D\times N}$$
$$\approx \underbrace{\left(\quad B \ : \text{Basis} \quad \right)}_{D\times M} \underbrace{\left(\qquad Y \ : \text{Weights/Components} \qquad \right)}_{M\times N}$$

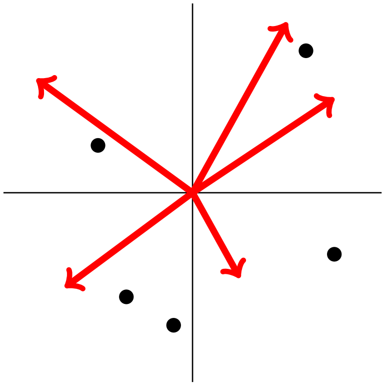
Based on the SVD of the data matrix, we see that PCA is in this class. Many methods can be considered as matrix decompositions under specific constraints.

Under-complete decompositions



When $M < D$, there are fewer basis vectors than dimensions. The matrix $\mathbf{B}$ is then called ‘tall’ or ‘thin’. In this case the matrix $\mathbf{Y}$ forms a lower dimensional approximate representation of the data $\mathbf{X}$, PCA being a classic example.

Over-complete decompositions



For $M > D$ the basis is over-complete, there being more basis vectors than dimensions. In such cases additional constraints are placed on either the basis or components. For example, one might require that only a small number of the large number of available basis vectors is used to form the representation for any given $\mathbf{x}$. Such sparse-representations are common in theoretical neurobiology where issues of energy efficiency, rapidity of processing and robustness are of interest.

Probabilistic Latent Semantic Analysis (PLSA)

Consider two objects, x and y, where $\mathrm{dom}(x) = \{1, \dots, I\}$ and $\mathrm{dom}(y) = \{1, \dots, J\}$ and a dataset $(x^n, y^n, n = 1, \dots, N)$. We have a count matrix with elements C_{ij} which describes the number of times the joint state $x = i, y = j$ was observed in the dataset. We can transform this count matrix into a frequency matrix p with elements

$$p(x = i, y = j) = \frac{C_{ij}}{\sum_{ij} C_{ij}}$$

Our interest is to find a decomposition of this frequency matrix of the form

$$\underbrace{p(x = i, y = j)}_{X_{ij}} \approx \sum_k \underbrace{\tilde{p}(x = i|z = k)}_{B_{ik}} \underbrace{\tilde{p}(y = j|z = k)\tilde{p}(z = k)}_{Y_{kj}} \equiv \tilde{p}(x = i, y = j)$$

where all quantities $\tilde{p}$ are distributions. This is then a form of matrix decomposition into positive basis $\mathbf{B}$ and positive coordinates $\mathbf{Y}$. This has the interpretation of discovering latent topics z that describe the joint behaviour of x and y.

An EM style training algorithm

For probabilities, a useful measure of discrepancy is the Kullback-Leibler divergence

$$\mathrm{KL}(p|\tilde{p}) = \langle \log p \rangle_p - \langle \log \tilde{p} \rangle_p$$

Since p is fixed, minimizing the Kullback-Leibler divergence with respect to the approximation $\tilde{p}$ is equivalent to maximizing the 'likelihood' term $\langle \log \tilde{p} \rangle_p$. This is

$$L \equiv \sum_{x,y} p(x,y) \log \tilde{p}(x,y)$$

It's convenient to derive an EM style algorithm to learn $\tilde{p}(x|z)$, $\tilde{p}(y|z)$ and $\tilde{p}(z)$.

Consider

$$\begin{aligned}&\mathrm{KL}(q(z|x,y)|\tilde{p}(z|x,y))\\&\qquad=\sum_z q(z|x,y)\log q(z|x,y)-\sum_z q(z|x,y)\log\tilde{p}(z|x,y)\geq 0\end{aligned}$$

where $\sum_z$ implies summation over all states of the variable z. Using

$$\tilde{p}(z|x,y)=\frac{\tilde{p}(x,y,z)}{\tilde{p}(x,y)}$$

and rearranging, this gives the bound,

$$\log\tilde{p}(x,y)\geq-\sum_z q(z|x,y)\log q(z|x,y)+\sum_z q(z|x,y)\log\tilde{p}(z,x,y)$$

Plugging this into the 'likelihood' term above, we have the bound

$$\begin{aligned}L\geq&-\sum_{x,y}p(x,y)\sum_z q(z|x,y)\log q(z|x,y)\\&\qquad+\sum_{x,y}p(x,y)\sum_z q(z|x,y)\left[\log\tilde{p}(x|z)+\log\tilde{p}(y|z)+\log\tilde{p}(z)\right]\end{aligned}$$

M-step

For fixed $\tilde{p}(x|z), \tilde{p}(y|z)$, the contribution to the bound from $\tilde{p}(z)$ is

$$\sum_{x,y} p(x,y) \sum_{z} q(z|x,y) \log \tilde{p}(z)$$

Up to a constant, this is $\mathrm{KL}\left(\sum_{x,y} q(z|x,y)p(x,y)|\tilde{p}(z)\right)$ so that, optimally,

$$\tilde{p}(z) = \sum_{x,y} q(z|x,y)p(x,y)$$

Similarly, optimally

$$\tilde{p}(x|z) \propto \sum_{y} p(x,y)q(z|x,y)$$

and

$$\tilde{p}(y|z) \propto \sum_{x} p(x,y)q(z|x,y)$$

E-step

The optimal setting for the q distribution at each iteration is

$$q(z|x, y) = \tilde{p}(z|x, y)$$

which is fixed throughout the M-step.

Convergence

L is guaranteed to increase (and the Kullback-Leibler divergence decrease) under iterating between the E and M-steps, since the method is analogous to an EM procedure.

Matlab

```
>> setup;
>> demoPLSA;
```

Modelling citations

We have a collection of research documents which cite other documents. For example, document 1 might cite documents $3, 2, 10$, *etc.* Given only the list of citations for each document, can we identify key research papers and the communities that cite them?

A probabilistic formulation

We use the variable $d \in \{1, \ldots, D\}$ to index documents and $c \in \{1, \ldots, D\}$ to index citations (both d and c have the same domain, namely the index of a research article). If document $d = i$ cites article $c = j$ then we set the entry of the matrix $C_{ij} = 1$. If there is no citation, C_{ij} is set to zero. We can form a 'distribution' over documents and citations using

$$p(d = i, c = j) = \frac{C_{ij}}{\sum_{ij} C_{ij}}$$

and use PLSA to decompose this matrix into citation-topics.

Modelling citations

The Cora corpus contains an archive of around 30,000 computer science research papers. From this archive, the papers in the machine learning category are extracted, consisting of 4,220 documents and 38,372 citations.

Using PLSA

The joint PLSA method is fitted to the data using $z = 7$ topics. From the trained model the expression $p(c = j|z = k)$ defines how authoritative paper j is according to community $z = k$.

Modelling citations

factor 1	(Reinforcement Learning)
0.0108	Learning to predict by the methods of temporal differences. Sutton.
0.0066	Neuronlike adaptive elements that can solve difficult learning control problems. Barto et al.
0.0065	Practical Issues in Temporal Difference Learning. Tesauro.
factor 2	(Rule Learning)
0.0038	Explanation-based generalization: a unifying view. Mitchell et al.
0.0037	Learning internal representations by error propagation. Rumelhart et al.
0.0036	Explanation-Based Learning: An Alternative View. DeJong et al.
factor 3	(Neural Networks)
0.0120	Learning internal representations by error propagation. Rumelhart et al.
0.0061	Neural networks and the bias-variance dilemma. Geman et al.
0.0049	The Cascade-Correlation learning architecture. Fahlman et al.
factor 4	(Theory)
0.0093	Classification and Regression Trees. Breiman et al.
0.0066	Learnability and the Vapnik-Chervonenkis dimension. Blumer et al.
0.0055	Learning Quickly when Irrelevant Attributes Abound. Littlestone.
factor 5	(Probabilistic Reasoning)
0.0118	Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Pearl.
0.0094	Maximum likelihood from incomplete data via the em algorithm. Dempster et al.
0.0056	Local computations with probabilities on graphical structures. Lauritzen et al.
factor 6	(Genetic Algorithms)
0.0157	Genetic Algorithms in Search, Optimization, and Machine Learning. Goldberg.
0.0132	Adaptation in Natural and Artificial Systems. Holland.
0.0096	Genetic Programming: On the Programming of Computers by Means of Natural Selection. Koza.
factor 7	(Logic)
0.0063	Efficient induction of logic programs. Muggleton et al.
0.0054	Learning logical definitions from relations. Quinlan.
0.0033	Inductive Logic Programming Techniques and Applications. Lavrac et al.

Table: Highest ranked documents according to $p(c|z)$. The factor topic labels are manual assignments based on similarity to the Cora topics.