Everything you wanted to know about the mathematics of dimensionality reduction, but were afraid to ask: Teaching resources and activities

Soumya Banerjee [1]&

1 University of Cambridge, Cambridge, United Kingdom

Corresponding author sb2333@cam.ac.uk

Abstract

This work presents some teaching resources and activities to understand the mathematics of machine learning and unsupervised learning.

Introduction and basics

This work presents some teaching resources and activities to understand the mathematics of machine learning and unsupervised learning.

Some concepts to be covered are:

1. vector spaces

   inner product

   generalization of length and distance

   related to norm

   generalisation of angle

   orthogonal

   linear combination of basis

   projection on to lower dimensional space

2. necker cube

3. mirror image in 4d

4. https://www.coursera.org/learn/pca-machine-learning/lecture/svsZI/inner-product-length-of-vectors

5. dot product
6. eigenvalues and eigenvectors of covariance matrix
7. other and probabilistic interpretations of PCA

   https://www.coursera.org/learn/pca-machine-learning/lecture/qrMP1/other-interpretations-of-pca-optional
8. PCA resources and book

   https://hastie.su.domains/ISLR2/ISLRv2_website.pdf

   https://www.quora.com/What-is-an-intuitive-explanation-for-PCA

   https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
9. eigenvalues

   https://www.youtube.com/watch?v=PFDu9oVAE-g
10. transformation
11. axis of rotation
12. eigenvalue 1 means just fixed in place
13. determinants
14. rotation
15. eigenvalues related to linear system of equations

    https://www.cs.unm.edu/ williams/cs530/kl3.pdf

    https://ncert.nic.in/ncerts/l/lemh103.pdf
16. mathematical basics of machine learning

    linear algebra

    https://mml-book.github.io/book/mml-book.pdf
17. inner product generalization of length

    https://www.youtube.com/watch?v=Ww_aQqWZhz8

Dot product

The dot product is defined as

$$x^T y$$

Length

The length is defined as

$$|x| = \sqrt{x^T x}$$

https://en.wikipedia.org/wiki/Dot_product#Tensors

Distance

The length/distance between two vectors is

$$|x - y| = \sqrt{(x - y)^T (x - y)}$$

https://www.coursera.org/learn/pca-machine-learning/lecture/qQfS4/dot-product

Angle between two vectors

The angle α between two vectors x and y is given by:

$$\alpha = \frac{x^T y}{||x||||y||}$$

But we know that

$$||x|| = \sqrt{x^T x} \tag{1}$$

Hence we get the following relationship for the angle

$$\alpha = \frac{x^T y}{\sqrt{x^T x}\sqrt{y^T y}}$$

Generalizations of dot product: inner product

The inner product generalizes the notion of dot product:

$$< x, y >$$

Length: in terms of inner product

The length of the vector can be defined in terms of the inner product as

$$||x|| = \sqrt{< x, x >}$$

The length is also called the norm.

Distance

Distance: in terms of inner product

$$d(x, y) = \sqrt{< x - y, x - y >}$$

This assumes we can do vector addition and subtraction.

Angle

1. Symmetric
2. Bilinear
3. Positive definite

Other properties are:

1. Triangle inequality

$$||x + y|| \leq ||x|| + ||y||$$

2.

$$||\lambda x|| = ||\lambda|| ||x||$$

3. Cauchy-Schwarz inequality

$$|| < x, y > ||$$

What if not symmetric like K-L divergence?

youtube.com/watch?v=lrSQH_69MTM

orthogonal polynomials

inner product for polynomials

convolution as an inner product

https://www.google.com/searchq=is+convolution+an+inner+product&gs_ivs=1#tts=0

orthogonal functions

https://www.youtube.com/watch?v=ZYf0tz9oVz8

angle between polynomials

https://math.stackexchange.com/questions/556291/angle-between-polynomials

Other generalizations

Orthogonal functions

https://www.youtube.com/watch?v=ZYf0tz9oVz8

Fourier series is a least squares problem

https://www.youtube.com/watch?v=u8ccubUfhKY

polynomials as vectors

https://www.youtube.com/watch?v=xhTciBubSfM

Inner product can be defined for functions as well. This can be defined with integration.

The inner product of two functions u and v is defined as:

$$< u, v >= \int_a^b u(x)v(x)dx$$

Functions can be orthogonal to each other, with respect to this inner product. For example, consider the functions $cos(x)$ and $sin(x)$:

$$< cos(x), sin(x) >= \int_{-\pi}^{\pi} cos(x)sin(x)dx = 0$$

Hence this is a basis.

Projecting other functions onto this basis is the idea behind Fourier series.

Basis

Basis vectors

Any other vector can be composed of a linear combination of these basis vectors

Orthogonal basis: vectors that are orthogonal to each other angle between them is 90 degrees.

$$\alpha = \frac{x^T y}{||x||||y||}$$

For example, consider two vectors b_1 and b_2. They can form an orthonormal basis if their inner product is:

$$< b_1, b_2 >= 0$$

and their length is

$$||b_1|| = 1$$

and

$$||b_2|| = 1$$

Basis of polynomials

https://www.youtube.com/watch?v=pYoGYQOXqTk

why 1, x, x^2 is a terrible basis

Similarity metrics

https://www.researchgate.net/publication/372487573_Similarity_metrics_metrics_and_conditionally_negative_definite_func

Projection onto 1D

When we project x onto U, we are trying to find $\pi_U(x)$. This is the projection of x onto U. We will denote this as

$$\pi_U(x)$$

x is a vector.

Hence we want something that is closest to x. Hence the following quantity is minimal:

$$||x - \pi_U(x)||$$

$x - \pi_U(x)$ is orthogonal to U. Hence $x - \pi_U(x)$ is a basis vector of U. Let us call this basis vector b. Hence b and $x - \pi_U(x)$ are orthogonal to each other.

By definition of orthogonality the following holds for the inner product:

$$< b, x - \pi_U(x) >= 0$$

Also since $\pi_U(x)$ is a projection onto b, it must be a multiple of b. Hence

$$\pi_U(x) = \lambda b$$

Let us now try to find λ.

Let us start from the orthogonality condition:

$$< b, x - \pi_U(x) >= 0$$

Substituting the value of $\pi_U(x) = \lambda b$ we get

$$< b, x - \lambda b >= 0$$

The inner product is bilinear. Hence we get

$$< b, x > -\lambda < b, b >= 0$$

Rearranging we get

$$\lambda = \frac{< b, x >}{< b, b >}$$

Noting that $< b, b >= ||b||^2$ is the norm we have:

$$\lambda = \frac{< b, x >}{||b||^2}$$

If the inner product is the dot product we have

$$\lambda = \frac{b^T x}{||b||^2}$$

For an orthonormal basis, we have $||b|| = 1$. Hence

$$\lambda = b^T x$$

The projected vector $\pi_U(x)$ is

$$\pi_U(x) = \lambda b = \frac{b^T x b}{||b||^2}$$

Generalized projection onto n-dimensions

We now generalize these results to n-dimensions.

$\pi_U(x)$ is the projection and hence must be a linear combination of the basis vectors b_1, b_2, ...

Hence we have the following condition:

$$\pi_U(x) = \sum \lambda_i b_i$$

Concretely if there are m basis vectors:

$$\pi_U(x) = \sum_i^m \lambda_i b_i$$

This can be written in matrix notation as

$$\pi_U(x) = \sum_i^m \lambda_i b_i = B\lambda$$

Now we want the closest or minimum distance. Hence the basis vector b_1 and $\pi_U(x) - x$ must be orthogonal.

Hence the inner product must be 0:

$$< b_1, \pi_U(x) - x >= 0$$

and so on for all the basis vectors

$$< b_2, \pi_U(x) - x >= 0$$

$$< b_3, \pi_U(x) - x >= 0$$

and so on for all the basis vectors.

This can be written in matrix notation as:

$$< B, \pi_U(x) - x >= 0$$

or using $\pi_U(x) = B\lambda$ we have

$$< B, B\lambda - x >= 0$$

We have to now find λ.

For the case of the dot product this becomes

$$B^T(B\lambda - x) = 0$$

Solving we get

$$B^T B\lambda = B^T x$$

Simplifying we get

$$\lambda = (B^T B)^{-1} B^T x$$

Matrix decompositions

The determinant is the measure of volume.

The determinant is also invariant under choice of basis of a linear mapping.

Characteristic polynomial

$$\det(A - \lambda I) = c_0 + c_1\lambda + c_2\lambda^2 + ...$$

The eigenvalue equation is

$$Ax = \lambda x$$

$$\det(A - \lambda I) = 0$$

Eigenvalues, determinant, and trace are key parameters of a linear mapping that are invariant under a change of basis.

https://math.stackexchange.com/questions/507641/show-that-the-determinant-of-a-is-equal-to-the-product-of-its-eigenvalues

You can calculate determinants in your web browser using WolframAlpha. See link below:

https://www.wolframalpha.com/calculators/determinant-calculator

These are a series of transformations.

Spectral theorem

As a consequence of the spectral theorem, for a symmetric matrix A, we can get:

$$A = PDP^T$$

where D is a diagonal matrix and P is a matrix such that the columns have the eigenvectors.

The determinant of a matrix A is the product of its eigenvalues

$$\det(A) = \prod_i \lambda_i$$

The trace of a matrix is the sum of its eigenvalues:

$$\mathrm{tr}(A) = \sum_i \lambda_i$$

As a result of this, since

$$Ax_1 = \lambda x_1$$

where x_1 is the eigenvector and λ is the eigenvalue

the eigenvalue scales and stretches the original length by λ.

Hence the area increases by $\lambda_1 \lambda_2$ and the perimeter becomes $2(\lambda_1 + \lambda_2)$.

This is how the procedure transforms the original system.

CONCEPT:

A is a matrix of transformations. Eigenvectors give us the direction of transformation. Eigenvalues show us how much to stretch.

Cholesky decomposition

This is a square root like operation for matrices:

$$A = LL^T$$

Eigendecomposition

$$A = PDP^{-1}$$

where D is a diagonal matrix whose diagonal entries are the eigenvalues and P is a matrix such that the columns have the eigenvectors.

This represents a series of transformations:

1. P^{-1} performs a change of basis into the eigenbasis.

2. The diagonal matrix D scales the vectors by the eigenvalues λ

3. P scales the vectors back into the original co-ordinate system.

Singular value decomposition

$$A = U\Sigma V^T$$

where A is a rectangular matrix.

This represents a series of transformations:

1. V^T performs a change of basis.

2. The singular matrix Σ scales the vectors and performs dimensionality reduction (or augmentation).

3. U scales the vectors back into the original co-ordinate system.

Singular value equation

Similar to the eigenvalue equation, there is a singular value equation:

$$Av = \sigma u$$

Compare this to the eigenvalue equation:

$$Ax = \lambda x$$

where x is the same on both sides.

This yields $AV = U\Sigma$ which upon rearrangement gives us $A = U\Sigma V^{-1}$

Applications of SVD

SVD can be used to find structure in matrices, eg. structure in movie ratings, consumer behaviour, matrix approximation, etc.

It can be used to find the low-rank representation. The assumption is that movie preferences are linear combinations.

Other extensions and applications include multi-dimensional scaling, isomap and PCA.

Generalization of SVD to tensors

Tucker decomposition generalizes SVD to tensors (multi-dimensional arrays):

https://www.alexejgossmann.com/tensor_decomposition_tucker/

Other reading material

Non-negative matrix factorization

https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

Matrix norm and Frobenius norm

PCA

PCA is the diagonalization of the covariance matrix.

The covariance between two random variables is the expected value of the product of the deviation from their means.

Let x_n be the data points with mean 0.

The data covariance matrix is given by

$$S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T$$

The data points x_n can be compressed or projected onto a lower dimensional space z_n

$$z_n = B^T x_n$$

where B is a projection matrix.

$$B = [b_1, b_2, ...]$$

where the columns of B are orthonormal.

$$b_i^T b_i = 1$$

$$b_i^T b_j = 0$$

if $i \neq j$.

1. PCA assumes a linear relationship between the data points and the lower dimensional representation, since $z_n = B^T x_n$.
2. PCA also assumes a squared reconstruction error (dot product).

Derivations

PCA aims to maximize the variance of the coordinates. Let z_1 be the first projected coordinate.

We aim to maximize the variance

$$\frac{1}{N}\sum_{n=1}^{N} z_{1n}^2$$

where $z_{1n} = b_1^T x_n$ (the projected coordinate).

Substituting this into the equation above, we get the following.

$$\frac{1}{N}\sum_{n=1}^{N} (b_1^T x_n)^2$$

Again we note that PCA uses the dot product. The dot product is symmetric with respect to arguments i.e.

$$b_1^T x_n = x_n^T b_1$$

Using this, we get the following

$$\frac{1}{N}\sum_{n=1}^{N} b_1^T x_n x_n^T b_1$$

Pulling in the summation inside (since b_1 does not depend on n), we get

$$\frac{1}{N} b_1^T (\sum_{n=1}^{N} x_n x_n^T) b_1$$

$$b_1^T (\frac{1}{N}\sum_{n=1}^{N} x_n x_n^T) b_1$$

The term in the summation (within parentheses) is the data covariance matrix S. Hence we get

$$b_1^T S b_1$$

This is the quantity we aim to maximize.

$$\max_{b_1} b_1^T S b_1$$

where we aim to find b_1 such that the quantity is maximized.

Maximizing this quantity we get the following relationships

$$S b_1 = \lambda b_1$$

which is the eigenvector/eigendecomposition equation and

$$b_1^T b_1 = 1$$

which implies that the basis vector has length 1.

Hence we choose the basis vector corresponding to the largest eigenvalue of the data covariance matrix.

λ represents the variance explained by the first coordinate b_1. $\sqrt{\lambda}$ is called the loading.

In summary, PCA is

1. linear in its variables.

2. uses the dot product (one form of the inner product).

3. uses the squared error as the loss function.

Alternative derivation

We outline an alternative derivation here.

Let x_n be a point in the original space and z_m is the corresponding point in lower-dimensional space. z_m can be projected back into the original space. Let us call that $\hat{x}_n$ (the corresponding point in a lower-dimensional space).

If b_m denotes the basis vectors, then since PCA is linear the point in lower-dimensional space (z_m) is a linear combination of the basis vectors:

$$\hat{x}_n = \sum_{m=1}^{M} z_m b_m$$

where z_m is the corresponding point in lower-dimensional space.

We intend to minimize the sum of squared error (loss function) between the original points and the reprojected points (reconstruction loss)

$$J = \frac{1}{N} \sum_{n=1}^{N} ||x_n - \hat{x_n}||^2$$

This is the reconstruction loss.

We can now use use gradient descent to find the minima or find an explicit formula.

Taking partial derivatives:

$$\frac{\partial J}{\partial z_m} = \frac{\partial J}{\partial x_n} \frac{\partial x_n}{\partial z_m}$$

where each term on the RHS is:

$$\frac{\partial J}{\partial x_n} = -\frac{2}{N}(x_n - \hat{x_n})^T$$

Now we use the relationship $\hat{x}_n = \sum_{m=1}^{M} z_m b_m$ to find $\frac{\partial x_n}{\partial z_m}$

$$\frac{\partial x_n}{\partial z_m} = b_m$$

Setting $\frac{\partial J}{\partial z_m} = \frac{\partial J}{\partial x_n}\frac{\partial x_n}{\partial z_m}$ to 0, we get

$$-\frac{2}{N}(x_n - \hat{x_n})^T b_m = 0$$

Rearranging we get

$$x_n b_m = \hat{x_n}^T b_m$$

On the RHS, we can write $\hat{x}_n$ as $z_m b_m$

Hence we get

$$x_n b_m = (z_m b_m)^T b_m$$

Simplifying we get

$$x_n b_m = z_m^T b_m^T b_m$$

The basis vectors are orthonormal $b_m^T b_m = 1$. So we get

$$x_n b_m = z_m^T$$

$$z_m^T = x_n b_m$$

The optimal coordinate is a normal projection.

MDS

MDS (multi-dimensional scaling) is the diagonalization of the distance matrix (each pair).

tSNE

distance matrix

gram matrix

https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

UMAP

https://pair-code.github.io/understanding-umap/

https://pair-code.github.io/understanding-umap/supplement.html

Autoencoders

Activities

1. Activities related to PCA

   https://www.quora.com/What-is-an-intuitive-explanation-for-PCA

   https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

2. Eigenvalues

   https://www.youtube.com/watch?v=PFDu9oVAE-g

3. Play with these tools (PCA, tSNE) in the browser

   http://projector.tensorflow.org/

Applications and case studies

removing outliers using PCA

t-test downstream

[1]

tsne hypothesis generation

Declarations

Acknowledgements

We acknowledge the help and support of the Accelerate team.

Funding statement

This work was funded by Accelerate Research Fellowship. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The views expressed are those of the authors and not necessarily those of the funders.

Conflicts of interests

All authors declare they have no conflicts of interest to disclose.

Ethics

No ethics approval was necessary.

Data accessibility

This study does not generate any clinical data.

Author contributions

SB carried out the analysis and implementation, participated in the design of the study and wrote the manuscript. All authors gave final approval for publication.

References

1. Aschenbrenner D, Quaranta M, Banerjee S, Ilott N, Jansen J, et al. (2019) Systems-level analysis of monocyte responses in inflammatory bowel disease identifies il-10 and il-1 cytokine networks that regulate il-23. bioRxiv : 719492.