ECE 657A/457B: Representation Learning

Dimensionality Reduction, Manifold Learning

Mark Crowley

February 4, 2025

Outline

Outline

Feature Extraction vs. Feature Selection

- Given a dataset $X : N \times D$ matrix.
- We can **extract** or **transform** new features F to describe the variation, distances, proximity, etc, in the data such that $|F| < D$.
- We can **select** from the existing features the ones which are the most representative F to describe X for $|F| < D$.

Dimensionality Reduction

Can be seen as a preprocessing step that may be required for:

- Reducing costs of training and learning algorithms
- Reducing storage and future measurement costs
- Visualization of the data to *debug* or otherwise *understand* your data.
- Finding the **intrinsic dimensionality** (features)
  - Many applications have a large number of features that may be redundant, irrelevant or sparse (e.g text documents where features are words)
- If the number of samples are much smaller than number of dimensions then this makes learning at a desired resolution difficult $\rightarrow$ **The Curse of Dimensionality**.

Motivation: The Curse of Dimensionality

- A term introduced by Richard Bellman to illustrate the explosion of the samples required when we have higher dimensions.
- The number of samples required to fully represent a 10 valued function with one dimension=10
- For 20 dimensions with same number of values (10 datapoints in each dimension) we will need 10^{20}
- The number of samples **grows exponentially** in terms of the dimensions.
- This poses a problem for **non-parametric** algorithms such as most classification algorithms that require many training samples.

Motivation: The Curse of Dimensionality

Learn a 1-D function from data:

How many data points are needed?

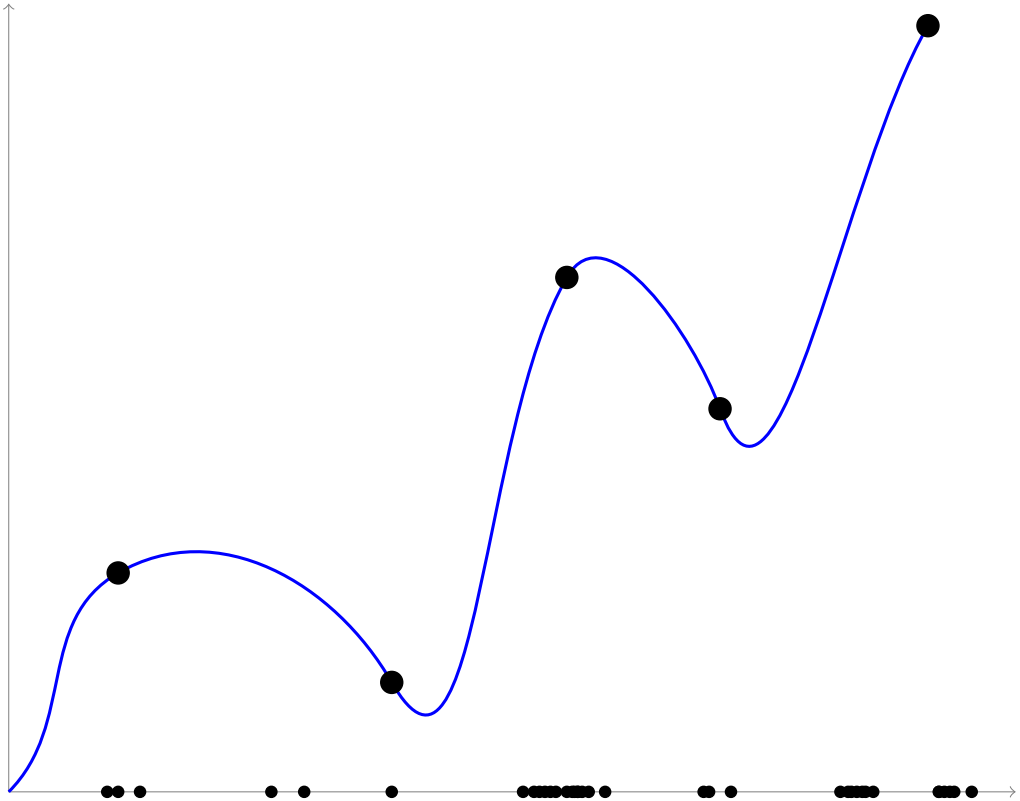
Motivation: The Curse of Dimensionality

Learn a 1-D function from data:

How many data points are needed? N points

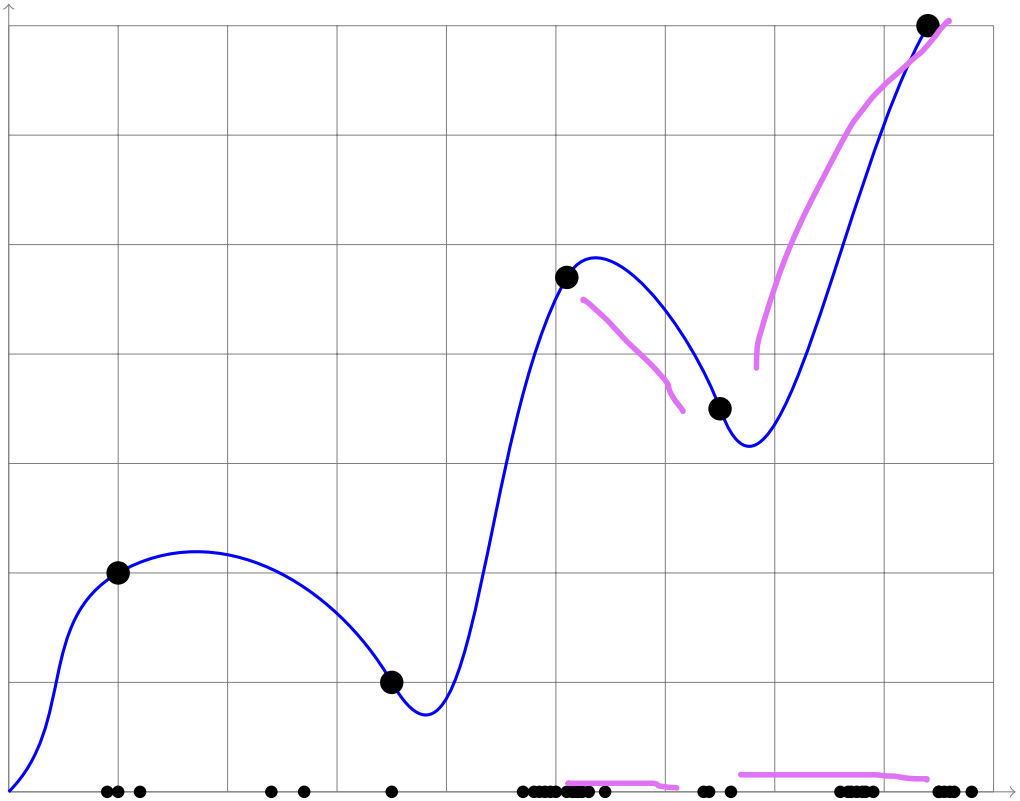
Motivation: The Curse of Dimensionality

What if the data is better explained by a 2-D function?



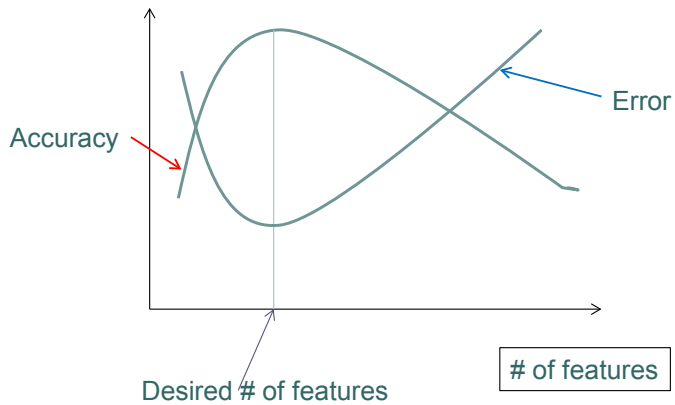
Motivation: The Curse of Dimensionality

What if the data is better explained by a 2-D function? N^2 points



Motivation: Peaking Phenomenon

- For a fixed number of samples increasing the number of features may degrade the performance. (eg. genetics data: very high dimensional, small data sample)
- Classic solution is to assume fixed order of features, only choose most important.
- Alternative approach: use classification algorithms to reduce feature space (see https://www.sciencedirect.com/science/article/abs/pii/S0167865508001



Dimensionality Reduction and Feature Extraction Methods

Feature Extraction and Feature Selection

- Feature Extraction combines original features into a new set of features by transformation or projection.
- Ideally the transformation or the projection is according to an objective that helps finding the significant set of new features of lower dimensions than the original set of features
- Sometimes the transformation produce new features that may have some relevant meaning to the data but many projection cases produce features that are difficult to interpret or relate to the meanings of the original features.

Feature Extraction

- **Feature Extraction:** ways to transform the original features in the data into new features which have some advantage such as
  - lower dimensionality
  - better description of the variance in the data
  - better ability to discriminate (distinguish) data points or clusters of data points

Feature Selection

- **Feature Selection** is attempt to find a subset of the original features which satisfy some criteria.
- Ideally the selected subset includes the significant features and eliminates irrelevant and redundant features.
- The selected features being a subset of the original are interpretable and keep their original meaning.
- Approaches include:
  - Feature Ranking
  - Subset Selection which includes:
  - Wrapper approach and Filter approach
  - Both use search strategies such as:
    - Exhaustive
    - Sequential (Forward or Backward)

Outline

Feature Selection : General Approach

- Search the feature space for a *subset of features* that optimizes a selection criterion, an objective function (ie. a quality index or a classifier output).
- Challenge: If we have n samples and d features then there are 2^d possible subsets of d.

- Components:
  1. Selection Criteria
  2. Search Strategy

Feature Selection : General Approach

- Search the feature space for a *subset of features* that optimizes a selection criterion, an objective function (ie. a quality index or a classifier output).
- Challenge: If we have n samples and d features then there are 2^d possible subsets of d. Even if we pick subsets of a fixed size this won't avoid an exponential search
- Components:
  1. Selection Criteria
  2. Search Strategy

Feature Ranking

We need to use more feasible methods.
Idea: Treat the problem as a search problem and find ways to reduce the search space.
Feature Ranking:

1. (also called Best Individual Features (BIF) and Naive Selection)
2. Evaluate the individual features according to a quality measure (how good the feature is for discriminating the class)
3. Sort the features according to their value of the quality measure
4. Select the best m features

Advantage: search complexity is $O(m)$ after sorting. It can be used for very large number of features
Problem: features are considered in isolation

Limitations of Correlation

Correlation is good at finding linear noise relationships and it's direction(pos or neg), but not non-linearerities or orientation.

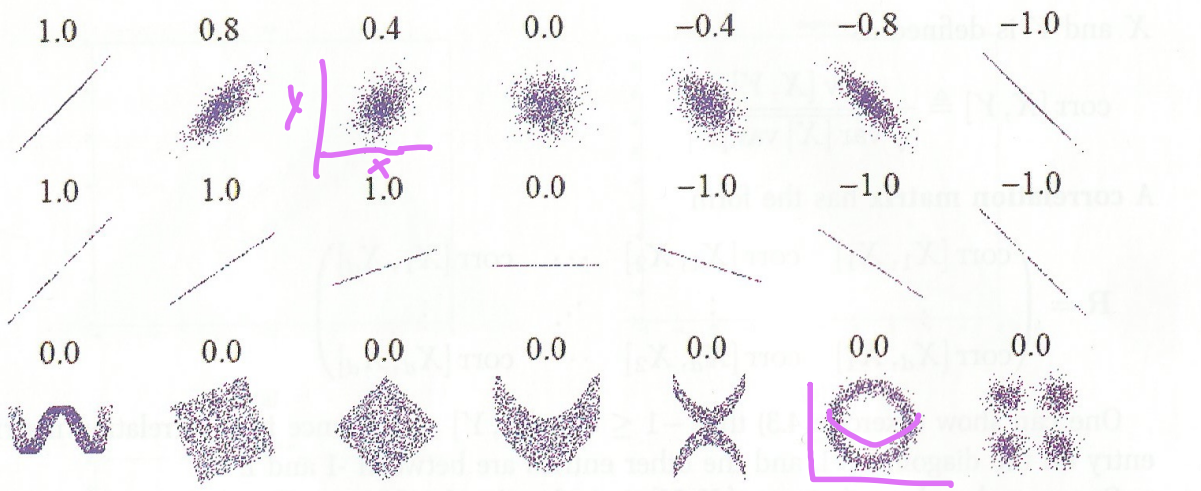


Figure: From (Murphy, 2012) Fig 2.12

So what could we use that's better?

Quality Measures for Feature Ranking

Some quality measures include:

- Mutual Information: between a feature and class
- Information Gain
- Chi-squared test (χ^2)

See the probstatsreview video for more on this if you've missed it.

Recall: Mutual Information (MI)

The **mutual information (MI)** between two vectors X, Y measures how similar the joint distribution $p(X, Y)$ is to the factored distribution $p(X)p(Y)$:

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

- MI(X,Y) is always nonnegative
- Equals 0 iff X, Y are independent
- Notice this is just the KL-Divergence between the distributions $p(X, Y)$ and $p(X)p(Y)$

Mutual Information For Feature Ranking

In our case, mutual information is:

$$MI(X_f, Y) = \sum_{x_{fi} \in X_f} \sum_{y_k \in Y} p(x_{fi}, y_k) \log \frac{p(x_{fi}, y_k)}{p(x_{fi})p(y_k)}$$

- where $x_{fi} \in X_f$ is the value of feature (column) f for datapoint i
- and $y_k \in Y$ are the class labels
- a feature with high MI may not have high probability but it's better at identifying the classes.

Feature Selection Methods

- Feature Ranking
  - Quality Measures for Feature Ranking
- Choosing Feature Subsets
  - Filter Approach: evaluate feature subsets based on information content
  - Wrapper Approach: evaluate feature subsets based on maximizing accuracy for final classifier being used
  - Objective Functions and Distance Measures: Embedded vs. Hybrid
- Search Strategies
  - Exhaustive Search
  - Sequential Forward Selection
  - Sequential Backward Selection
  - Sequential Floating Selection
  - Oscillating Search

Outline (S)

Feature Extraction Methods

- Feature Extraction combines original features into a new set of features by **transformation** or **projection**.
- Projection is according to some objective function or concept.
  - objective should find the *most significant set of new features*
  - used as a **lower dimensional subspace** than the original set of features
- Interpretation:
  - Transformation can produce new features have interpretable meaning to the data
  - But many projections produce features that are difficult to relate to the original features.

Methods of Feature Extraction

- Linear methods:
  - Unsupervised (no class label information, e.g .PCA, ICA)
  - Supervised (class labels know, e.g LDA)
- Nonlinear methods
  - Global (preserves global properties, eg. MDS, Isomap, Kernel PCA)
  - Local (preserves properties within local neighborhood, e.g. LLE)

Outline

Feature Extraction Methods

- Feature Extraction combines original features into a new set of features by **transformation** or **projection**.
- Projection is according to some objective function or concept.
  - objective should find the *most significant set of new features*
  - used as a **lower dimensional subspace** than the original set of features
- Interpretation:
  - Transformation can produce new features have interpretable meaning to the data
  - But many projections produce features that are difficult to relate to the original features.

Methods of Feature Extraction

- Linear methods:
  - Unsupervised (no class label information, e.g .PCA, ICA)
  - Supervised (class labels know, e.g LDA)
- Nonlinear methods
  - Global (preserves global properties, eg. MDS, Isomap, Kernel PCA)
  - Local (preserves properties within local neighborhood, e.g. LLE)

Feature Selection

- Feature Selection is attempt to find a subset of the original features which satisfy some criteria.
- Ideally the selected subset includes the significant features and eliminates irrelevant and redundant features.
- The selected features being a subset of the original are interpretable and keep their original meaning.
- Approaches include:
  - Feature Ranking
  - Subset Selection which includes:
  - Wrapper approach and Filter approach
  - Both use search strategies such as:
    - Exhaustive
    - Sequential (Forward or Backward)

Summary of Introduction

[Dunham, Data Mining Intro and Advanced Topics, 2003]
Margaret Dunham, Data Mining Introductory and Advanced Topics, ISBN:0130888923, Prentice Hall, 2003.

[Han,Kamber and Pei. Data Mining, 2011]
Jiawei Han, Micheline Kamber and Jian Pei, *Data Mining: Concepts and Techniques*, 3rd ed, Morgan Kaufmann Publishers, May 2011.

[Duda, Pattern Classification, 2001]
R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification (2nd ed.)*, John Wiley and Sons, 2001.

[Jain and Dubes. Algs for Clustering Data, 1988]
A. K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, ISBN: 0-13-022278-x, Prentice Hall, 1988.

[Cohen,Empirical Methods for Artificial Intelligence, 1995]
P. Cohen, Empirical Methods for Artificial Intelligence, ISBN:0-262-03225-2, MIT Press, 1995.

[Ackoff, From Data to Wisdom, 1989]

Ackoff, *From Data to Wisdom*, Journal of Applied Systems Analysis, 1989.

[Sima and Dougherty, 2008]
Sima, C. and Dougherty, E. R. *The Peaking Phenomenon in the Presence of Feature Selection.* Pattern Recognition Letters, 29, 1667-1674, 2008.

[Zhu and Ghodsi, 2006]
Mu Zhu, Ali Ghodsi, *Automatic dimensionality selection from the scree plot via the use of profile likelihood"*, Computational Statistics & Data Analysis 51 918 930, 2006.

[Cox, 2000]
Trevor Cox and M.A.A Cox, *Multidimensional Scaling*, Chapman and Hall/CRC, Second Edition, 2000.

Nonlinear Methods For Dimensionality Reduction

Additional material for the MDS section is based on the following references:

- Cox, T.F., Cox, M.A.A. Multidimensional Scaling. Chapman and Hall, 2001
- Tenenbaum, J. B., de Silva, V, Langford, J.C., "A global geometric framework for nonlinear dimensionality reduction " Science 290(5500): 2319-2323, 2000
- de Silva, V., Tenenbaum, J.B., Global versus local methods in nonlinear dimensionality reduction. In Neural Information Processing Systems. 15, 721-728, 2003.
- Roweis, S.T. and Saul, L.K., Nonlinear dimensionality reduction by Locally Linear Embedding. Science, 290(5500):2323-2326, 2000.
- van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J., Dimensionality reduction: a comparative review. Tilburg University Technical Report, TiCC-TR 2009-005, 2009.

Principal Component Analysis

- A way to linearly transform a set of d-dimensional vectors
- $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ into another set of m-dimensional vectors
- Has the property that most of the information content is stored in the first few dimensions, so we can have $m < d$
- The main idea is that high information corresponds to high variance (more discriminating).
- The direction of max variance is parallel to the eigenvector corresponding to the largest eigenvalue of the covariance matrix of the sample matrix A.

Intuitive View

- Find new axes which will be better in terms of variance and errors than original ones.
- PCA is equivalent to minimizing the mean-square error. It also maximizes the scatter.

Visual Explanation:
`http://setosa.io/ev/principal-component-analysis/`

Implementing PCA

- Let R be the $d \times d$ covariance matrix of A
- A is normalized by subtracting mean

$$x'_{ij} = (x_{ij} - \bar{x}_j), i = 1, \ldots, n; j = 1, \ldots, d$$

- R is symmetric positive definite, its eigenvalues are real and positive
- Now we apply an orthogonal transformation to R to diagonalize it.

$$\mathrm{CRC}^T = \Lambda_d \tag{1}$$

- Where Λ_d is a diagonal matrix of the d *eigenvalues* of R
- and C is a matrix with columns corresponding to the *eigenvectors* of R

Implementing PCA

We can sort the eigenvalues of R such that

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots \geq \lambda_d \geq 0 \tag{2}$$

and $\hat{c_1}, \hat{c_2}, \hat{c_3}, \ldots, \hat{c_d}$ are the corresponding eigenvectors, called the **Principal Components** (axes)

Selecting m dimensions

- If we want to reduce the dimensions but keep a large percentage of the variance in the data
- Then we can select the 1st m eigenvalues and eigenvectors

Let $H_m = \begin{bmatrix} \hat{c}_1^T \\ \hat{c}_2^T \\ \vdots \\ \hat{c}_m^T \end{bmatrix}$ be an $m \times d$ matrix.

Implementing PCA

Then

$$\bar{y}_i = H_m \bar{x}_i, \qquad i = 1, 2, \ldots, n$$
$$m \times 1 = m \times d \cdot d \times 1$$

The projected matrix B_m

$$B_m = \begin{bmatrix} \bar{y}_1{}^T \\ \bar{y}_2{}^T \\ \vdots \bar{y}_n{}^T \end{bmatrix} = \begin{bmatrix} \bar{x}_1{}^T \\ \bar{x}_2{}^T \\ \vdots \bar{x}_n{}^T \end{bmatrix} H_m^T = A H_m^T$$

where

$$\bar{x}_m^T = [x_{k1}, x_{k2}, \ldots, x_{kd}]$$
$$\bar{y}_m^T = [x_{k1}, x_{k2}, \ldots, x_{kd}]$$

Implementing PCA

The covariance matrix in the new space can be defined as

$$\begin{aligned}\frac{1}{n}B_m^T B_m &= \frac{1}{n}\sum_{i=1}^{n}\bar{y}_i\bar{y}_i^T = H_m R H_m^T \\ &= H_m(C^T\Lambda C)H_m^T = H_m C^T \Lambda (H_m C^T)^T \\ &= \Lambda_m = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)\end{aligned}$$

$$H_m C^T = \begin{bmatrix}\bar{c}_1^T \\ \bar{c}_2^T \\ \vdots \\ \bar{c}_m^T\end{bmatrix}[\bar{c}_1\,\bar{c}_2\ldots\bar{c}_m] = \begin{bmatrix}100\cdots0 \\ 010\cdots0 \\ \cdot\cdot1\cdots0 \\ \vdots \\ 10\cdots1\cdot\cdot0\end{bmatrix}$$

Which means the m new features are uncorrelated.

Sum of Eigenvalues

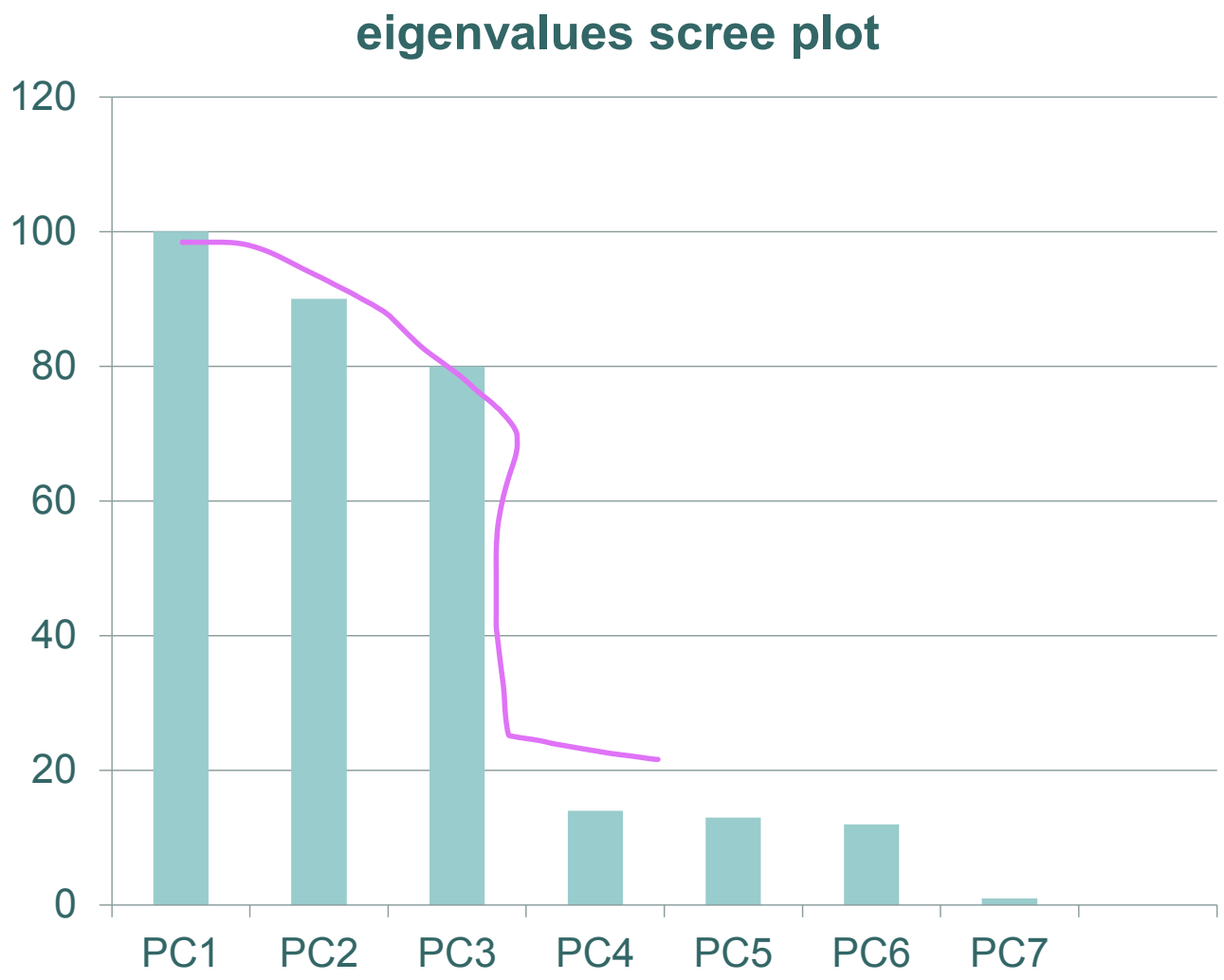
The sum of the Eigenvalues of R are the sample variance in the new space
One would choose m such that

$$r_m = \left(\sum_{i=1}^{m} \lambda_i\right) / \left(\sum_{i=1}^{d} \lambda_i\right) \geq \tau < 1$$

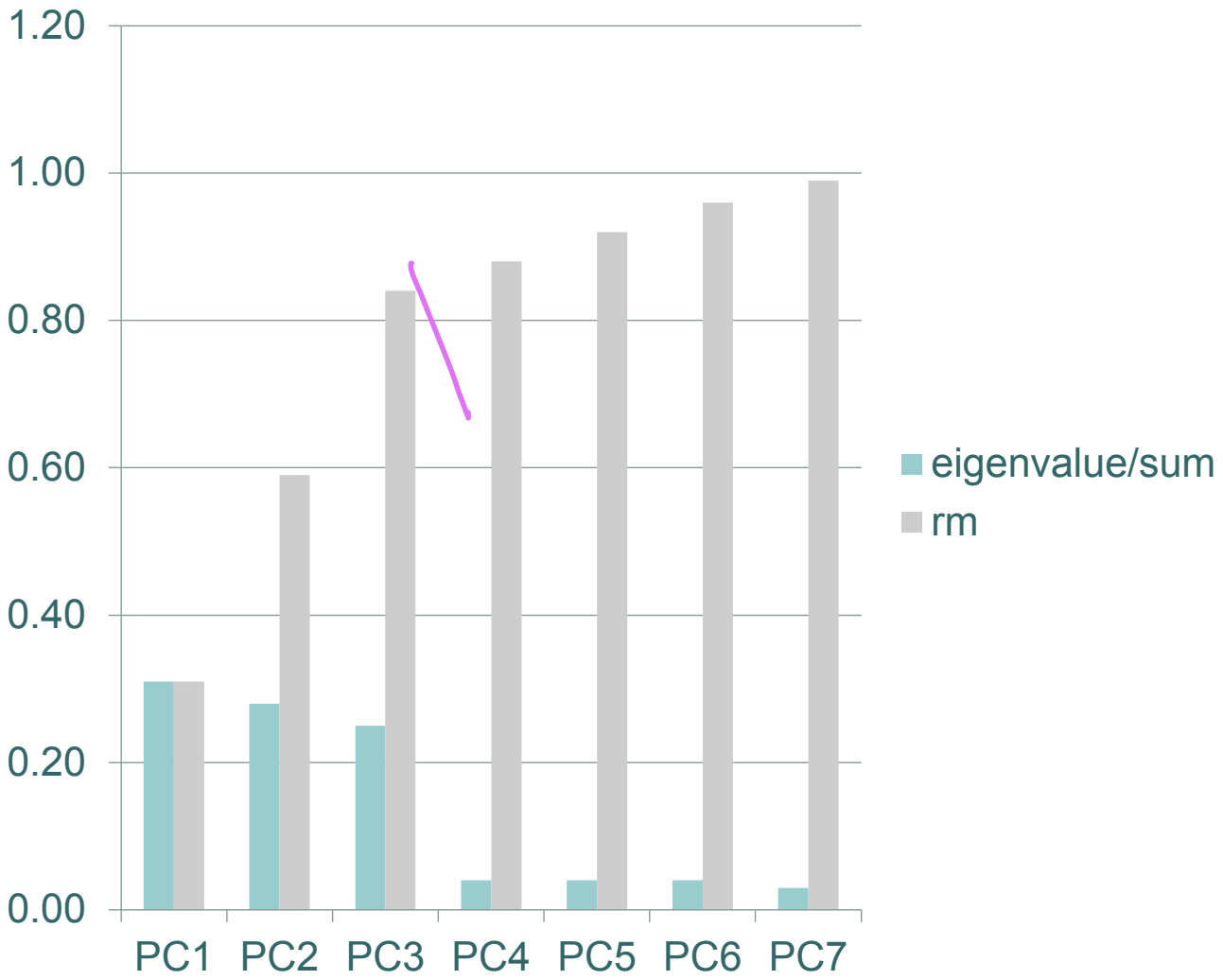
e.g. Choosing $\tau = 0.95$ will ensures that 95% of the variance is retained in the new space.
One way to know the right value is to use a **scree plot**.
The scree plot can be done in different ways:

- simply plotting the eigenvalues of the components (in descending order) and look for a gap or a knee in the plot.
- or plot r_m as a function of m and look for a knee in curve.
- could also plot the cumulative variance.

Scree Plot (Descending Eigenvectors)



Scree Plot (Normalized Eigenvectors)



Cost of Computation of PCA Optimization

- The approach we've shown so far is the most direct way to do PCA, but not the most efficient.
- The covariance matrix $R = A^T A$ is a $d \times d$ matrix and d (features) may be much larger than n (samples).
- Using this approach could be too complex.

Breaking Things Down...

A matrix A can be decomposed using **Singular Value Decomposition (SVD)** into $A_{n\times d} = USV^T$, where:

- U is a $n \times d$ matrix of orthonormal columns $U^T U = I$
  - the left singular vectors
- V is $d \times d$ matrix of orthonormal columns $V^T V = I$
  - the rght singular vectors
- S is $d \times d$ diagonal matrix of singular values.

PCA Using Singular Value Decomposition

SVD can be used to obtain PCA.
Now $AA^T = USV^T(VSU^T) = US^2U^T$
and $A^TA = VSU^T(USV^T) = VS^2V^T$
Which leads to the following facts:

- The singular values are the square root of the eigenvalues of the covariance matrix
- The right singular vectors are the eigenvectors of the covariance matrix.
- So, the SVD gives us the d eigenvalues (ordered) values as well as the principle components.
- Now we can *reduce the dimensions* by selecting the largest m.

Interpretation of PCA

- PCA is Optimal in the sense of min. sum of square of errors.
- It obtains max variance projection by finding orthogonal linear combinations of the original variables.
- It mainly rotates the coordinates (for zero mean data) to find new axes that have the max variance.
- It de-correlates the axes. For uni-modal Gaussian data this will amount to finding independent axes.
- For data that doesn't follow this distribution, the axes may not necessarily be independent.
- The principle components may not be the best for discriminating between classes.

Interpretation of PCA

- PCA is also called Karhunen-Loeve transform (KLT) or the Hotelling transform.
- The transformed points $\bar{y_1}, \bar{y_2}, \ldots, \bar{y_n}$ are sometimes called **scores**
- The eigenvectors or principal components $\bar{c_1}, \bar{c_2}, \ldots, \bar{c_d}$ are called **loadings** represented by **coefficients**
- The eigenvalues of the covariance matrix are sometimes called the **latent representation**
- Hotelling's T^2 value: measures the distance of the projected points from the centre of the entire entire projected space.