AI-503 Advanced Machine Learning

Recap

- Broad classification of learning
  - Supervised Learning: Learn using both feature vectors targets of the training examples
    - Classification
    - Regression
    - Ranking…
  - Unsupervised Learning: Learn using feature vectors only
    - Feature Extraction/ Selection
    - Clustering
    - …
  - Weakly Supervised learning
    - Semi supervised learning
    - Multiple Instance Learning
    - ….

Recap

- Broad classification of learning
  - Supervised Learning: Learn using both feature vectors targets of the training examples
    - Classification
    - Regression
    - Ranking…
  - Unsupervised Learning: Learn using feature vectors only
    - Feature Extraction/ Selection
    - Clustering
    - …
  - Weakly Supervised learning
    - Semi supervised learning
    - Multiple Instance Learning
    - ….

Unsupervised methods

- Up until now…
  - Supervised Learning
  - Learn models using labeling information
- Unsupervised methods
  - No labeling information used

https://github.com/foxtrotmike/PCA-Tutorial
https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Principal Component Analysis

- fundamentally a dimensionality reduction algorithm
- can also be useful as a tool for
  - visualization
  - feature extraction and engineering,
  - and much more.

Dimensionality Reduction

- Consider the following examples

x1	x2
1	5
1	8
1	4
1	7
1	3
1	2

- Do we really need x1?

Dimensionality Reduction

- How about now?

x1	x2
5	5
8	8
4	4
7	7
3	3
2	2

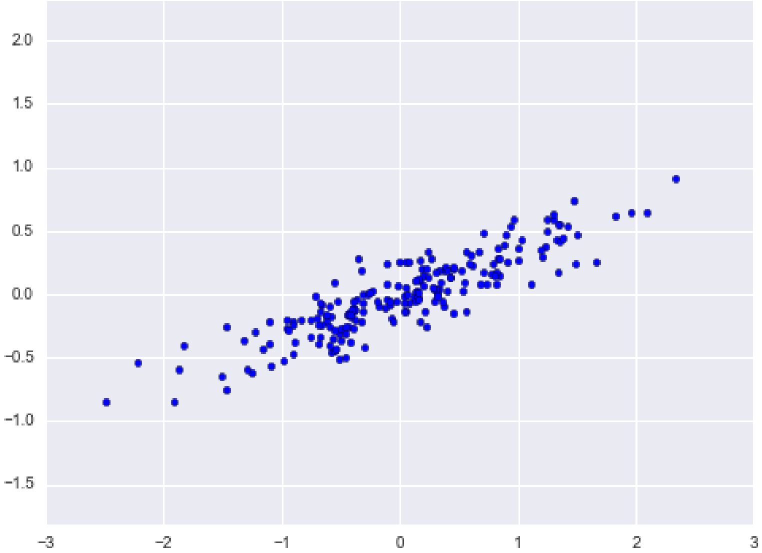
Dimensionality Reduction

- How about now?

x1	x2
5	5
8	8
4	4
7	7
3	3
2	2

What do you see?

- There is a nearly linear relationship between the x and y variables
- So if we know the relationship, we do not need to store both the values for all the examples
- Different from regression
  - rather than attempting to *predict* the y values from the x values, the unsupervised learning problem in PCA attempts to learn about the *relationship* between the x and y values.
- How?
  - In principal component analysis, this relationship is quantified by finding a list of the *principal axes* in the data, and using those axes to describe the dataset.



Dimensionality Reduction

- How can we reduce the dimensionality?
  - Remove dimensions with little or no information
- How is the amount of information quantified?
  - for a given variable, the amount of information in it is proportional to its variance –
    - if all data is constant, then its variance is zero and so is its information content

Why do we want to remove redundant/correlated features?

- Having many correlated features may cause
  - Slow convergence
  - Harmful bias if features not important for the actual problem
  - Worsening of effects of curse of dimensionality
  - Difficulty in interpreting models

PCA

- If we have very high dimensional data, we can reduce its dimensionality by projecting it along directions (or vectors) such that the variance along the chosen direction is maximized in order to preserve the most information in the data.

PCA

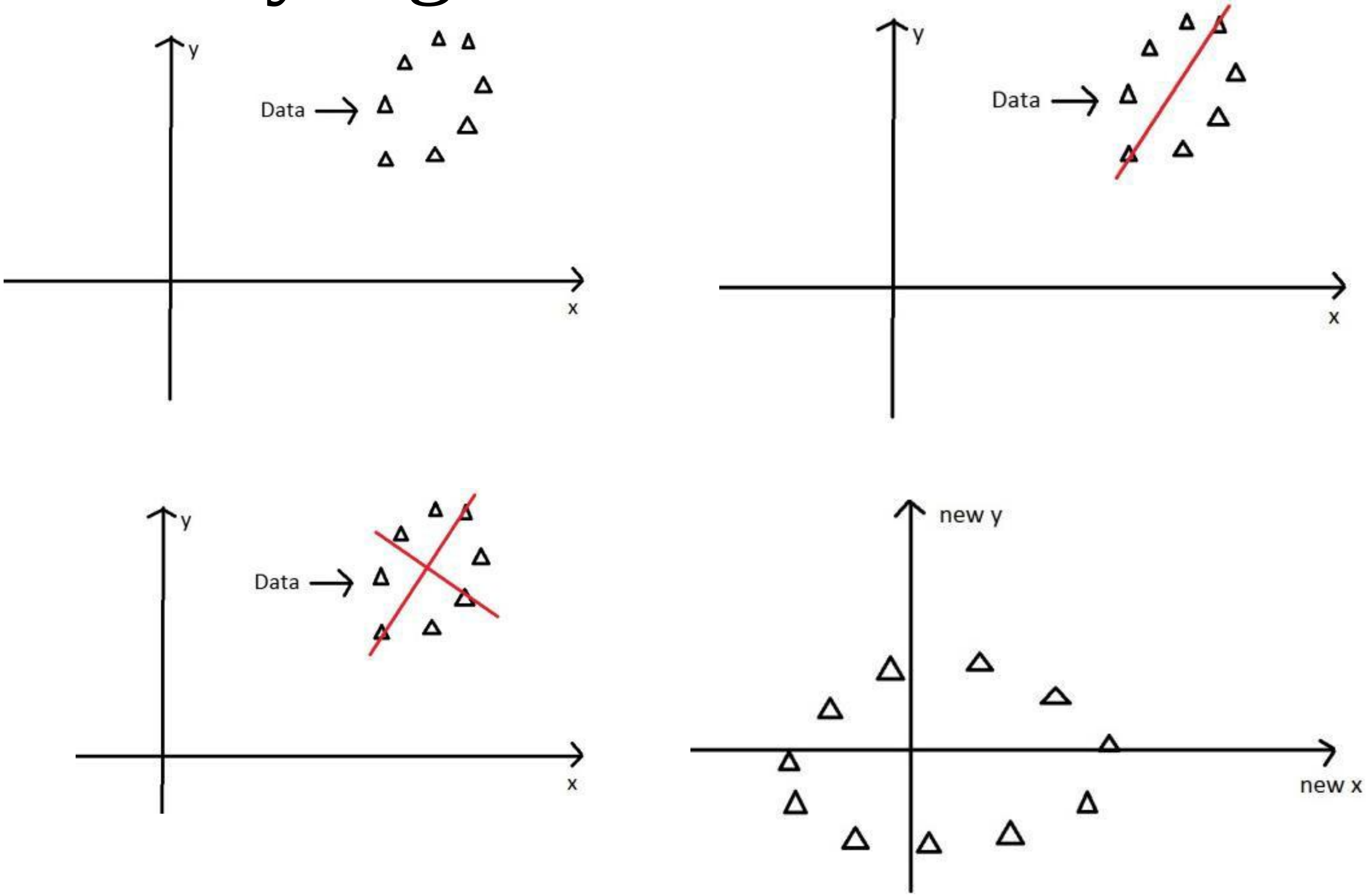
- Principal component analysis is a method for finding orthogonal directions of maximum variance in the data such that if the data is projected in those directions, the variance of the projected data is maximum.

- The dimensionality of the data can be reduced by projecting it along those directions. This projection along the direction of maximum variance gives minimum information loss.

- Finding the direction of maximum variance for a given data set corresponds to **finding the eigen vector of the covariance matrix of the data**

- leads to a minimum loss in information if the data is projected in that direction.

Let’s try to visualize what we are trying to achieve



https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/

- Finding the direction of maximum variance for a given data set corresponds to **finding the eigen vector of the covariance matrix of the data**

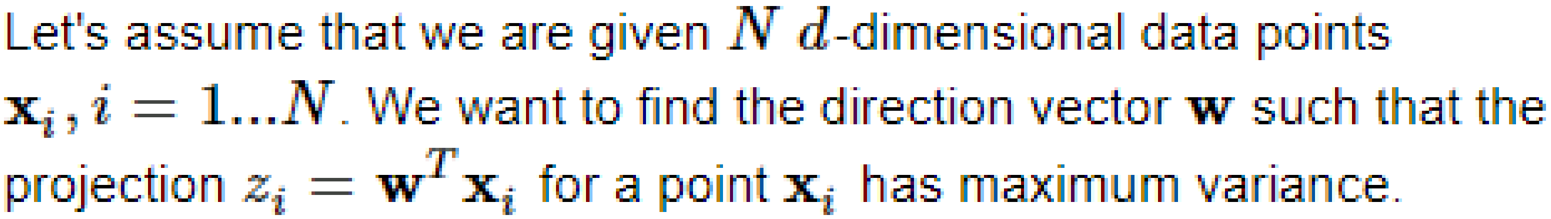
- Why? How?

Goals

- Find the axes, over which if I project the data, the variance is maximized
- Project the data over those axes

How to find the max variance axes?

- Formulate maximization of variance as an optimization problem
- Solve the problem

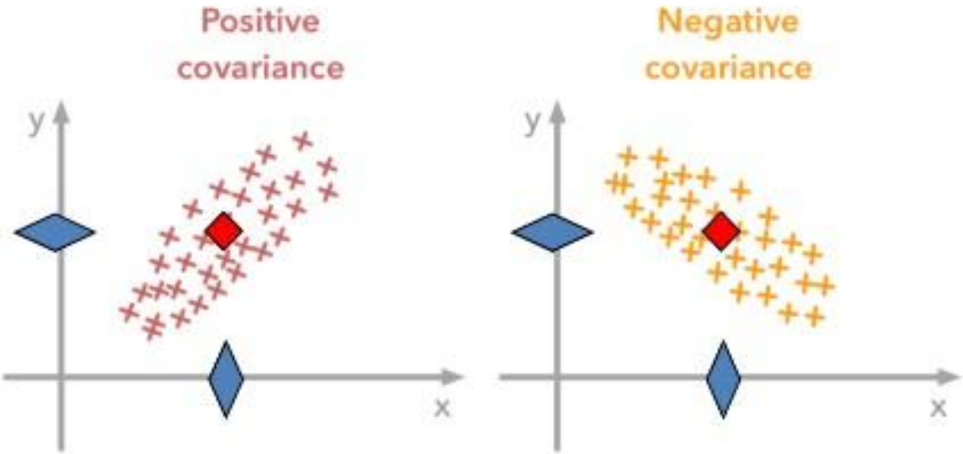


Let's assume that we are given N d-dimensional data points $\mathbf{x}_i, i = 1...N$. We want to find the direction vector $\mathbf{w}$ such that the projection $z_i = \mathbf{w}^T\mathbf{x}_i$ for a point $\mathbf{x}_i$ has maximum variance.

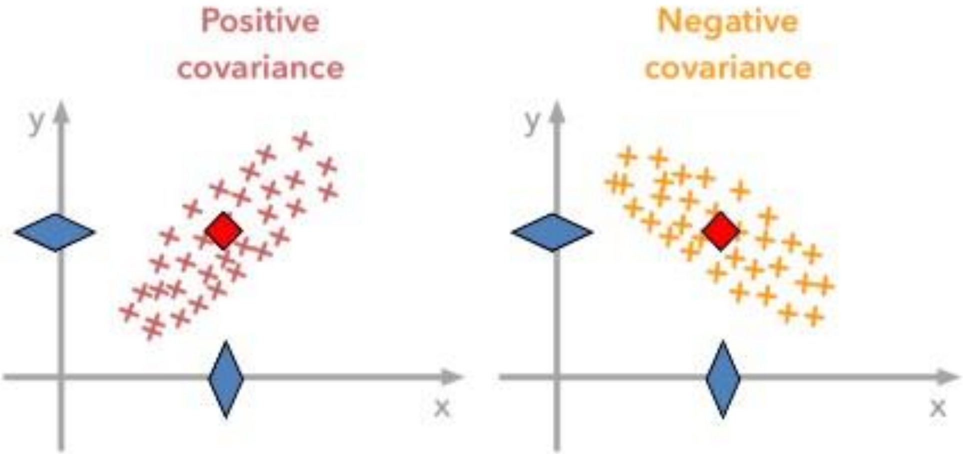
Let's assume that we are given N d-dimensional data points $\mathbf{x}_i, i = 1...N$. We want to find the direction vector $\mathbf{w}$ such that the projection $z_i = \mathbf{w}^T\mathbf{x}_i$ for a point $\mathbf{x}_i$ has maximum variance.

- Variance
  - Mean of the spread of a variable around its mean
  - $var(z) = \frac{1}{N}\sum_{i=1}^{N}(z_i - \mu_z)^2 = \frac{1}{N}(\mathbf{z} - \mu_z)^T(\mathbf{z} - \mu_z)$
    - $\boldsymbol{z}$ is an N-dimensional vector composed of the values of all data points in the sample
  - If mean is zero then $var(z) = \frac{1}{N}\mathbf{z}^T\boldsymbol{z} = \frac{1}{N}||\mathbf{z}||^2$
  - $var(z) = E[(z - \mu_z)^2]$
- Variance as an information measure
  - How is variance related to information content?

- Co-Variance
  - Given two random variables, to what extent are they linearly related to each other
  - $cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) = \frac{1}{N}(\boldsymbol{x} - \mu_x)^T(\boldsymbol{y} - \mu_y)$
    - Covariance is positive if, on average,
      - When one variable is above its mean then the other variable is above its mean too
      - When one variable is below its mean then the other variable is below its mean too
    - Covariance is negative if, on average,
      - When one variable is above the mean, the other is below its mean
  - Assume that the means are zero: $cov(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{N}\boldsymbol{x}^T\boldsymbol{y}$
    - Maximum when the vectors are co-linear or parallel
  - $cov(\boldsymbol{x},\boldsymbol{y}) = E[(y - \mu_y)(x - \mu_x)]$
  - Thus,$var(z) = cov(z,z)$

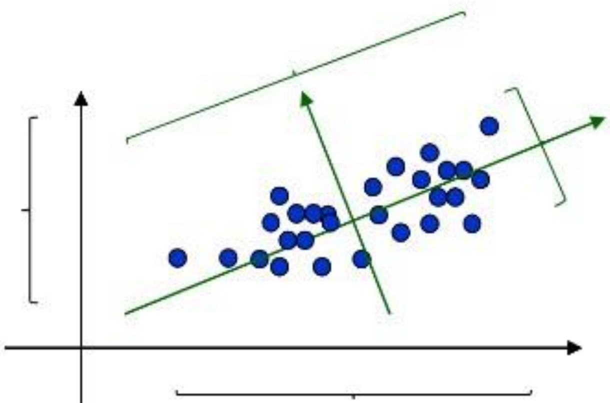


- Co-Variance
  - Given two random variables, to what extent are they linearly related to each other
  - $cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) = \frac{1}{N}(\boldsymbol{x} - \mu_x)^T(\boldsymbol{y} - \mu_y)$
    - Covariance is positive if, on average,
      - When one variable is above its mean then the other variable is above its mean too
      - When one variable is below its mean then the other variable is below its mean too
    - Covariance is negative if, on average,
      - When one variable is above the mean, the other is below its mean
  - Assume that the means are zero: $cov(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{N}\boldsymbol{x}^T\boldsymbol{y}$
    - Maximum when the vectors are co-linear or parallel
  - $cov(\boldsymbol{x},\boldsymbol{y}) = E\left[(y - \mu_y)(x - \mu_x)\right]$
  - Thus,$var(z) = cov(z,z)$



PCA

- A method for transforming the data
  - Projecting the data onto orthogonal vectors such that the variance of the projected data is maximum
  - Projection of *x* on the direction of *w*: $z = w^T x$
  - Find *w* such that Var(*z*) is maximized



Principal Component Analysis

- Relation between variance of projection and covariance matrix

$$
\begin{aligned}
\mathrm{Var}(z) = \mathrm{Var}(w^{T}x) &= \mathrm{E}[(w^{T}x - w^{T}\mu)^{2}] \\
&= \mathrm{E}[(w^{T}x - w^{T}\mu)(w^{T}x - w^{T}\mu)] \\
&= \mathrm{E}[w^{T}(x - \mu)(x - \mu)^{T}w] \\
&= w^{\mathrm{T}}\,\mathrm{E}[(x - \mu)(x - \mu)^{T}]w = w^{T}\,\mathrm{C}\,w
\end{aligned}
$$

where $\mathrm{Cov}(x) = \mathrm{E}[(x - \mu)(x - \mu)^{T}] = \mathrm{C}$

Covariance Matrix

- Representing covariance among dimensions as a matrix, e.g., for 3 dimensions:

$$C = \begin{bmatrix} \mathrm{cov}(X,X) & \mathrm{cov}(X,Y) & \mathrm{cov}(X,Z) \\ \mathrm{cov}(Y,X) & \mathrm{cov}(Y,Y) & \mathrm{cov}(Y,Z) \\ \mathrm{cov}(Z,X) & \mathrm{cov}(Z,Y) & \mathrm{cov}(Z,Z) \end{bmatrix}$$

- Properties:
  - Diagonal: variances of the variables
  - $\mathrm{cov}(X,\mathrm{Y})=\mathrm{cov}(Y,X)$, hence matrix is symmetrical about the diagonal (upper triangular)

Connection to Eigen Vectors

- Note that $\boldsymbol{Cw} = \alpha \boldsymbol{w}$
- Thus, our solution means that $\boldsymbol{w}$ is, in essence, an eigen vector of $\boldsymbol{C}$ with eigen value α
  - The eigen vectors of the covariance matrix (Called Principal Components) of the given dataset are along the direction of maximum variance of the data!!
  - Typically, the eigen vectors are normalized as unit vectors $\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$

PCA Process – STEP 1

- Subtract the mean from each of the dimensions
- This produces a data set whose mean is zero.
- Subtracting the mean makes variance and covariance calculation easier by simplifying their equations.
- The variance and co-variance values are not affected by the mean value.
- Suppose we have two measurement types $\mathbf{X}_1$ and $\mathbf{X}_2$, hence $m = 2$, and ten samples each, hence $n = 10$.

PCA Process – STEP 1

X_1	X_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.2	0.9

$$\Rightarrow \quad \begin{array}{l} \overline{X_1} = 1.81 \\ \overline{X_2} = 1.91 \end{array} \quad \Rightarrow$$

X_1'	X'_2
0.69	0.49
−1.31	−1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	−0.31
−0.81	−0.81
−0.31	−0.31
−0.71	−1.01

PCA Process – STEP 2

- Calculate the covariance matrix

$$S_X = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

- Since the non-diagonal elements in this covariance matrix are positive, we should expect that both the $\mathbf{X}_1$ and $\mathbf{X}_2$ variables increase together.
- Since it is symmetric, we expect the eigenvectors to be orthogonal.

PCA Process – STEP 3

- Calculate the eigen vectors **V** and eigen values **D** of the covariance matrix

$$D = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

PCA Process – STEP 3

- Calculate the eigen vectors **V** and eigen values **D** of the covariance matrix

$$D = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

- Reduce dimensionality and form *feature vector*

  The eigenvector with the *highest* eigenvalue is the *principal component* of the data set.

  In our example, the eigenvector with the largest eigenvalue is the one that points down the middle of the data.

  Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance.

- Reduce dimensionality and form *feature vector*

The eigenvector with the *highest* eigenvalue is the *principal component* of the data set.

In our example, the eigenvector with the largest eigenvalue is the one that points down the middle of the data.

Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance.

PCA Process – STEP 4

Now, if you'd like, you can decide to *ignore* the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don't lose much

PCA Process – STEP 4

- When the λ_i's are sorted in descending order, the proportion of variance explained by the *r* principal components is:

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{m} \lambda_i} = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_r}{\lambda_1 + \lambda_2 + \ldots + \lambda_p + \ldots + \lambda_m}$$

- If the dimensions are highly correlated, there will be a small number of eigenvectors with large eigenvalues and *r* will be much smaller than *m*.

- If the dimensions are not correlated, *r* will be as large as *m* and PCA does not help.

PCA Process – STEP 5

- Derive the new data

PCA

- 1. Find the principal components
- 2. Transform data by projecting over principal components
- But where is the dimensionality reduction?
  - Select a smaller number of principal components for transformation than the actual dimensionality
- Implemented in Sklearn
  - https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- Demo: https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.09-Principal-Component-Analysis.ipynb#scrollTo=6tCGHfiDfmeN

Required reading

- Please go through the following tutorial
- https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html