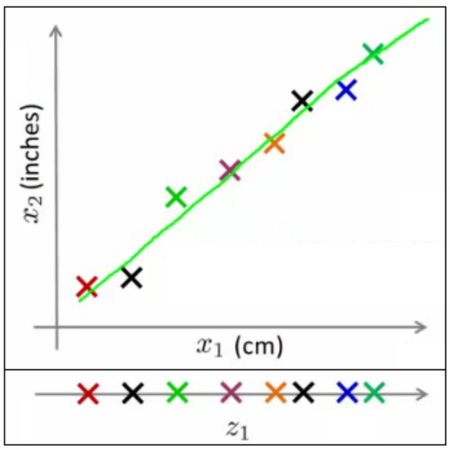
Week 14

August 8, 2021

Compression

- Speeds up algorithms.
- Saves space.
- Dimension reduction: not all features are needed.
- Example: different units for same attribute.



Now we can represent x1 as a 1D number (Z dimension).

Visualization

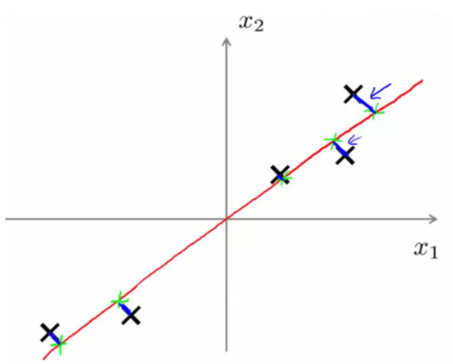
- It is difficult to visualize higher dimensional data.
- Dimensionality reduction can help us show information in a more readable fashion for human consumption.
- Collect a huge data set including numerous facts about a country from around the world.

Country	GDP (trillions of US$)	Per capita GDP (thousands of intl. $)	Human Develop-ment Index	Life expectancy	Poverty Index (Gini as percentage)	Mean household income (thousands of US$)	...
Canada	1.577	39.17	0.908	80.7	32.6	67.293	...
China	5.878	7.54	0.687	73	46.9	10.22	...
India	1.632	3.41	0.547	64.7	36.8	0.735	...
Russia	1.48	19.84	0.755	65.5	39.9	0.72	...
Singapore	0.223	56.69	0.866	80	42.5	67.1	...
USA	14.527	46.86	0.91	78.3	40.8	84.3	...
...	...	...	...	...	...	...	

- Assume each country has 50 characteristics.
- How can we better comprehend this data?
- Plotting 50-dimensional data is quite difficult.
- Create a new feature representation (2 z values) that summarizes these features.
- Reduce $50D -> 2D$ (now possible to plot).

Principle Component Analysis (PCA): Problem Formulation

- Assume we have a 2D data collection that we want to reduce to 1D.
- How can we choose a single line that best fits our data?
- The distance between each point and the projected version should be as little as possible (blue lines below are short).
- PCA tries to find a lower dimensional surface so the sum of squares onto that surface is minimized.
- PCA tries to find the surface (a straight line in this case) which has the minimum projection error.



- PCA is not linear regression.
- For linear regression, fitting a straight line to minimize the straight line between a point and a squared line. VERTICAL distance between point.
- For PCA minimizing the magnitude of the shortest orthogonal distance.
- With PCA there is no y - instead we have a list of features and all features are treated equally.

PCA Algorithm

- Compute the covariance matrix.

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n}(x^{(i)})(x^{(i)})^T$$

- This is an $[nxn]$ matrix (Remember than x^i is a $[n \times 1]$ matrix).
- Next, compute eigenvectors of matrix Σ.

U,S,V = svd(sigma)

- U matrix is also an $[n \times n]$ matrix. Turns out the columns of U are the u vectors we want!
- Just take the first k-vectors from U.
- Next, calculate z.

$$z = (U_{reduce})^T \cdot x$$

Reconstruction from Compressed Representation

- Is it possible to decompress data from a low dimensionality format to a higher dimensionality format?

$$x_{approx} = U_{reduce} \cdot z$$

- We lose some information (everything is now precisely aligned on that line), but it is now projected into 2D space.

Choosing the number of Principle Components

- PCA attempts to minimize the averaged squared projection error.

$$\frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - x_{approx}^{(i)}||^2$$

- Total data variation may be defined as the average over data indicating how distant the training instances are from the origin.

$$\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}||^2$$

- To determine k, we may use the following formula:

$$\frac{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - x_{approx}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}||^2} \leq 0.01$$

Applications of PCA

- Compression: Reduce the amount of memory/disk space required to hold data.
- Visualization: k=2 or k=3 for plotting.
- A poor application of PCA is to avoid over-fitting. PCA discards certain data without understanding what values it is discarding.
- Examine how a system works without PCA first, and then apply PCA only if you have reason to believe it will help.