Data Preprocessing

Kaiqi Zhao
The University of Auckland

Data Reduction

Data Reduction

- Data reduction: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? – A database may store terabytes of data, complex data analysis may take a very long time to run on the complete data set
- The most commonly used data reduction strategy:
  - Dimensionality reduction, e.g. remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
- Other strategies: numerosity reduction, data compression

Dimensionality Reduction

- Curse of dimensionality

x_1	x_2	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	x_n
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	11	12	22	24
0	0	0	0	0	0	0	0	0	0	0	74	38	99	2	4	0	0	0	0	0
0	0	0	0	0	0	0	0	84	69	55	0	0	0	0	0	0	0	0	0	0
0	0	0	0	66	35	14	62	0	0	0	0	0	0	0	0	0	0	0	0	0
32	48	54	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

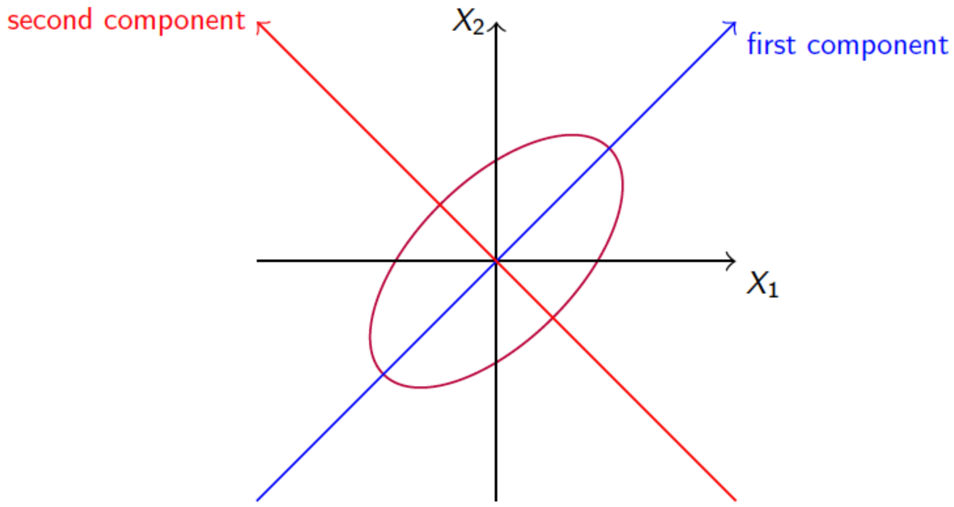
- When dimensionality increases, data becomes increasingly sparse
- Density and distance between points, which is critical to clustering, outlier analysis, classification, regression becomes less meaningful

Dimensionality Reduction

- Why dimensionality reduction?
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce required time and space
  - Allow easier visualization
- Dimensionality reduction techniques
  - Principal Component Analysis
  - Feature selection

Principal Component Analysis – PCA

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction
- How? - We find the eigenvalues and eigenvectors of the covariance matrix of input features
    - Eigenvalue - the amount of variance along the corresponding eigenvector
    - Eigenvector – the directions that variances occur



Question: The eigenvectors are orthogonal, are they correlated?

Demo: https://setosa.io/ev/principal-component-analysis/

PCA – steps

- Given n-dimensional feature vectors X, find $k \leq n$ orthogonal vectors (principal components) that can be best used to represent data
    - Normalize input data: each attribute falls within the same range
    - Compute the unit eigenvectors of the covariance matrix of X, i.e., principal components. The input is a linear combination of the principal components.
    - The principal components are sorted in order of decreasing "significance" or strength
    - Pick the top k principal components and remove the rest, i.e. those with low variance

- Does PCA work for categorical data?

Scikit-learn PCA:
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

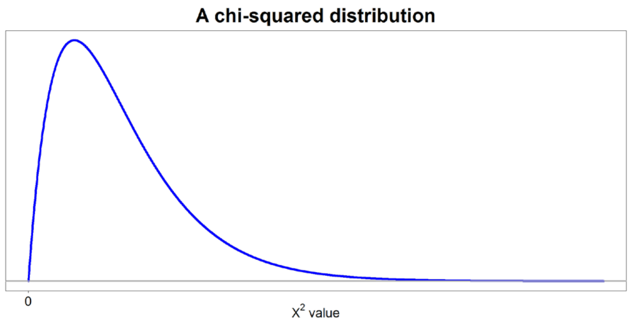
Feature or Attribute Selection

- Another way to reduce dimensionality of data – which cases could be candidates to be removed?
- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
  - E.g. purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
  - E.g. students' ID is often irrelevant to the task of predicting students' GPA

- Two types of methods – Filter and Wrapper

Feature Selection using Correlation

- For categorical data, given two attributes A and B with values $a_1, \dots, a_c$ and $b_1, \dots, b_r$ the correlation can be calculated using the χ^2 test:



$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{\left(o_{ij} - e_{ij}\right)^2}{e_{ij}}$$

  - With o_{ij} being the actual frequency of the event (a_i, b_j)
  - And e_{ij} the expected frequency (n is the number of instances)

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}$$

- The larger χ^2, the less likely the two variables are independent

Feature Selection using Correlation

- Numerical data can be compared using Pearson's correlation coefficient

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$$

- With means $\bar{A}$ and $\bar{B}$, number of instances n, and standard deviations σ_A and σ_B
- So what does the correlation measure?
- How can it be used to remove redundant or unimportant features?

Heuristic Search in Attribute Selection

- There are 2^d possible subsets of d attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption
  - Best step-wise feature selection:
    - The best single-attribute is picked first
    - Then next best attribute condition on the first, ...
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute
  - Best combined attribute selection and elimination
  - Optimal branch and bound:
    - Use attribute elimination and backtracking

Relief

- The step-wise feature selection has a big drawback – which one?
- Relief is a feature selection algorithm that addresses this:

Input: Data set with d attributes and n instances that belong to one of two classes, and parameter $N_r < n$

First normalize the data, Create a weight vector W with one weight $w_i \in W$ for each attribute

Initialize the weights to 0

for $j \in 1 \dots N_r$ **do**

    Randomly select instance $x = [x_1, \dots, x_d]$

    Choose instance $h = [h_1, \dots, h_d]$ as the closest neighbour of x in the same class (*nearHit*)

    Choose instance $m = [m_1, \dots, m_d]$ as the closest neighbour of x in the other class (*nearMiss*)

    for $i \in 1 \dots d$ **do**

        $w_i = w_i - (x_i - h_i)^2 + (x_i - m_i)^2$

    end

End

for $i \in 1 \dots d$ **do**

    $w_i = \frac{w_i}{N_r}$

end

Relief

- Relief takes into account **all** attributes
- Result is a weight vector that represents the importance of each feature
- Features are then selected based on a threshold or ranked
- The algorithm above is the basic version of Relief, there are various extensions (ReliefF, RReliefF,. . .)

Wrappers

- The correlation method and Relief are filters
- Wrappers: generate a subset of the features and evaluate the performance of the classifier on the subset
- Add or remove attributes from the subset and see if the performance of the classifier improves
- Risk of overfitting, especially if choosing the same classifier as for the main learning task