Data Reduction

* Reduce number of data items
  + Random sampling
  + Stratified sampling
* Reduce the number of attributes (dimensions)
  + Dimension reduction by transformation
  + Dimension reduction by elimination

Utmost goal

* Keep the gist of the data
* Only throw away what is redundant or superfluous

Data Reduction

* Reason
  + Reduce size of data can be feasibly store
  + Mining algorithm can run in affordable time

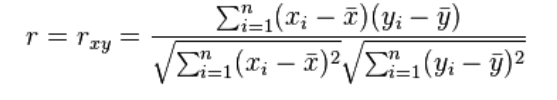
Sampling Principles

* Keep a representative number of samples:
  + Pick one of each
  + Or maybe a few more depending on importance
* How to pick
  + Measure of similarity in feature space
  + Data vector
    - Quantize each object into vector, each vector element is feature measurement compare the vectors in terms of similarity
  + Metric Distance: using in feature space to measure similarity
    - Manhattan Distance

* + - Euclidian Distance

Cosine Similarity vs Pearson’s Correlation Similarity

* Cosine Similarity
  + Text, whiteboard

    Description automatically generated
  + Stricter
* Pearson’s Correlation
  + 
  + Correlation distance is invariant to addition of a constant
* Both are invariant to scaling, multiplication with a constant

Variables vs Data items

* Distances can compare two attributes or two data items
  + Computing means, std based on correlation of two attributes
  + Or, computing stats based on two data items

Jaccard Distance

* measures the dissimilarity between data sets and is obtained by subtracting the Jaccard similarity coefficient from 1.

Clustering

* Definition
  + grouping of similar items based on metric distance
* Loss function
  + MSE
* Objective function
  + Minimize MSE
  + Diagram

    Description automatically generated
* In practice
  + Only one global minimum, many local minimum
  + Hard to find global minimum

K-means Clustering Algo

* Algorithm steps

1. Decide on a value for k
2. Initialize the k cluster centers (randomly, if necessary)
3. Decide the class memberships of the N objects by assigning them to the nearest cluster center
4. Re-estimate the k cluster centers, by assuming the memberships found above are correct
5. If none of the N objects changed membership in the last iteration, exit. Otherwise go to 3

* Strength
  + Relative efficient: O(tkn)
    - N: number of objects
    - K: number of clusters
    - T: number of iterations
  + Simple code
* Weakness
  + Need to specify k in advance
  + Find best k by trials with lowest MSE
  + Often end with local minimum

Redundancy Sampling

* Eliminate redundant attributes
  + Highly correlated
* Eliminate redundant data
  + Cluster the data with small range e
  + Only keep cluster centroids
  + Store size of clusters along to keep importance

Reservoir Sampling

* A good algorithm to use for streaming data when n is growing

CURE high-dimensional clustering algorithm

* Algorithm
  + initialize the point set S to empty
  + pick the point farthest from the mean as the first point for S
  + then iteratively pick points that are furthest from the points in S collected so far
* Complexity O(mn^2)
  + n is the total number of points, m is the number of desired points
  + can find arbitrarily shaped clusters and preserve outliers
  + need some good data structures to run efficiently: kd-tree, heap

PCA

* axis rotation
* find a coordinate system that can represent the variance in the data with as few axes as possible
* Algorithm
  + Find characterize the distribution by
    - Covariance matrix Cov
    - Correlation matrix Corr
  + Perform QR factorization or LU decomposition on the matrix to get
    - A: diagonal matrix with eigenvalues
  + Order the Eigenvectors in terms of their eigenvalues
* Explained variance rate = PC1/(PC1+PC2)

Covariance

* measure how much two random variables change together
* For N variable we have N^2 variable pairs -> covariance matrix
* Text

  Description automatically generated
* Correlation is defined for linear relationship

Covariance vs Correlation

* Covariance matrix
  + When the variable scales are similar
* Correlation matrix
  + When variables are on different scales
  + It standardizes the data
* In general they give different results, especially when the scales are different

How many Axes are needed

* A common “rule of thumb” when PCA is based on correlation is that axes with eigenvalues >1 are worth interpreting

Loading

* The amount of weight a data dimension has on a principal component

Significance of variables

* The sum of squared loadings on to the most significant eigenvectors