**IML Summary**

**Machine Learning Concepts**:

three ingredients of ML: data, models, learning; ML problem: regression, classification, clustering, anomaly detection and association rule; what are instances, attributes, concepts; supervised learning vs unsupervised learning; training data vs test data; instance is represented as feature vectors; possible attribute types: nominal, ordinal, continuous.

**Probability Theory**:

Probability of an event: fraction of times the event is true in independent trials(range of 0 to 1);

Joint probability: probability of two events occurring concurrently.

Property: iff A and B are independent.

Conditional probability: probability of an event given another event occurring.

Property: ;

Disjoint event: if .

Bayes rule:

Terms: : prior probability; : likelihood; : evidence; : posterior probability.

Binomial distribution: probability of an event with probability of occurring out of times is:

Multinomial distribution: probability of an event with probabilities of different outcomes occurring exact times is:

Marginalization: probability of one event irrespective of the outcome of another event.

Maximum likelihood estimate: find parameter set that maximizes the likelihood of the data.

Maximum posteriori estimate: find parameter set that maximizes the posterior distribution.

Expectation: the weighted average of all possible outcomes.

discrete variable:

continuous variable:

**Optimization**:

What is learning: find a set of model parameters that optimize the performance of a model.

How to find maxima/minima: ① define the objective function ; ② compute the first derivative with respect to parameter ; ③ set the derivative to zero; ④ solve for .

Multiple parameters: If a model has multiple parameters to be optimized, we need to compute the partial derivative with respect to each parameter , which is , and the updating phrase is done for all parameters. The recipe of gradient descent for multiple parameters:

文本

描述已自动生成

Constraint optimization: the parameter we are about to optimize is constrained to certain range, such that:

By combining Lagrange multiplier , we have the constraint objective function:

**Naïve Bayes**:

Key idea: the objective is to find a class label that maximizes conditional probability and reformulate the equation by using Bayes rule:

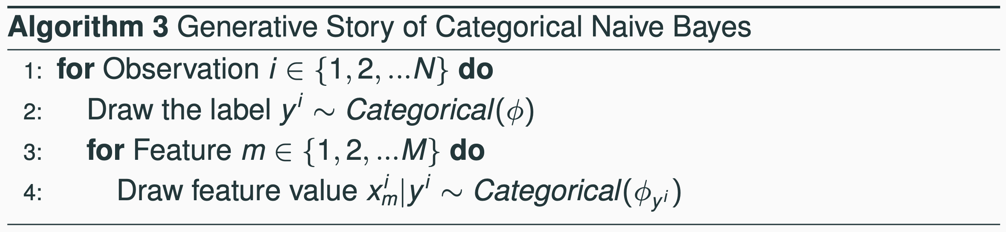
Thus, our objective function is:

Naïve Bayes assumption: ① the features are assumed to be independent given a class label ;

② instances are independent each other; ③ the distribution of training set and test set is the same.

For categorical feature value, we calculate the conditional probability by counting.

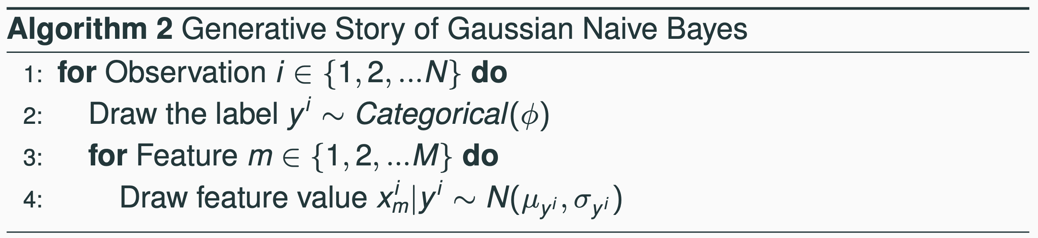
Categorical Naïve Bayes:



Maximum likelihood estimate to calculate and .

For continuous feature value, we calculate the conditional probability by using Gaussian distribution function:

Gaussian Naïve Bayes:



Handling zero probability: ① epsilon smoothing: replace 0 with a very small number , and we choose a class with the smallest number of . ② Laplace smoothing with probability:

图示

描述已自动生成

It will reduce variance but increase bias.

Handling missing value in a test instance: ignore those attributes with missing value.

Log transformation: to avoid numerical underflow issue, it will typically transform original probability to log-probability, such that:

**Evaluation Metrics**:

Accuracy = Precision = Recall = F1-score =

Train-test split technique: holdout: randomly partition instances into training and test instances with the fixed portion(e.g. 70-30); repeated random subsampling: perform holdout multiple times and average the performance of these models; cross validation: partition data into splits, and use splits for training and the rest as test set iteratively, and finally average the performance; stratification: the train-test set partition must satisfy that the data has the same distribution.

Multi-class evaluation:

① macro-averaging:

图片包含 文本

描述已自动生成

② micro-averaging:

图片包含 文本

描述已自动生成

③ weighted averaging:

图片包含 图示, 示意图

描述已自动生成

What is baseline and benchmark; Zero-R baseline: classify all instances to the most common class in the training data; One-R baseline: select one attribute with the smallest error rate.

What is underfitting(too simple) and overfitting(too complex); What is learning curve;

What is bias, variance and noise, bias-variance tradeoff.

Learning curve:

图表, 图示, 折线图

描述已自动生成

High bias remedy: ① use more complex model; ② add features; ③ boosting.

High variance remedy: ① add more training data; ② reduce features; ③ reduce model complexity; ④ bagging.

**KNN**:

Eager learning vs lazy learning: KNN is a lazy learning algorithm that we do not need to train a model, but compare test instances with training instances.

4 steps of KNN classifier: ① store training instances; ② measure distance between test and training instances; ③ find k-nearest neighbors; ④ return the class of the testing instances by majority voting.

Measurement of distance for different types of attributes:

1. Nominal:

where and is the number of features and the number of matched features.

1. Ordinal: we need to normalize these features, since the order matters

where is the number of features, and is the corresponding rank of a feature value.

1. Numerical:

* Euclidean distance:
* Manhattan distance:
* Cosine similarity:

Weighted KNN: classify a test instance according to the weighted accumulative class of KNN training instances.

Weighted strategies:

* Inversed linear distance:

where is the maximum distance between neighbors and the test instance, is the minimum distance between neighbors and the test instance, and is the distance between -th neighbor and the test instance.

* Inversed distance:

where is the small constant that is to avoid zero denominator.

Breaking tie techniques:

* Random tie breaking
* Choose one with highest prior probability
* See if the th instance breaks the tie

Choice of k: smaller k would have lower performance; higher k would have higher performance but high computational overhead.

**Logistic Regression**:

Discriminative model: a model that optimizes directly.

Naïve model: use linear combination of parameters and features to make a prediction:

However, the linear combination is not a promising model to predict the label accurately.

Odds: the fraction of success over the fraction of failure.

Sigmoid function:

图表, 直方图

描述已自动生成

Objective function:

Gradient of objective function:

Softmax:

**Perceptron**:

Motivation: biological imitation of neurons

Structure:

图示

描述已自动生成

Step function:

Perceptron algorithm:

文本, 信件

描述已自动生成

The convergence of perceptron algorithm depends on the ① parameter initialization; ② learning rate; ③ data to be split (non-linearly separatable data is not guaranteed to convergence).

Online learning vs batch learning.

**Neural Network**:

Three types of layers: input layer, hidden layer, output layer

Structure of neural network: each layer is fully connected with the neighboring layer, each neuron has an activation function that enables model performing non-linear tasks.

Activation function: ReLU, tanh, sigmoid(binary classification), softmax(multi-class classification)

Loss function:

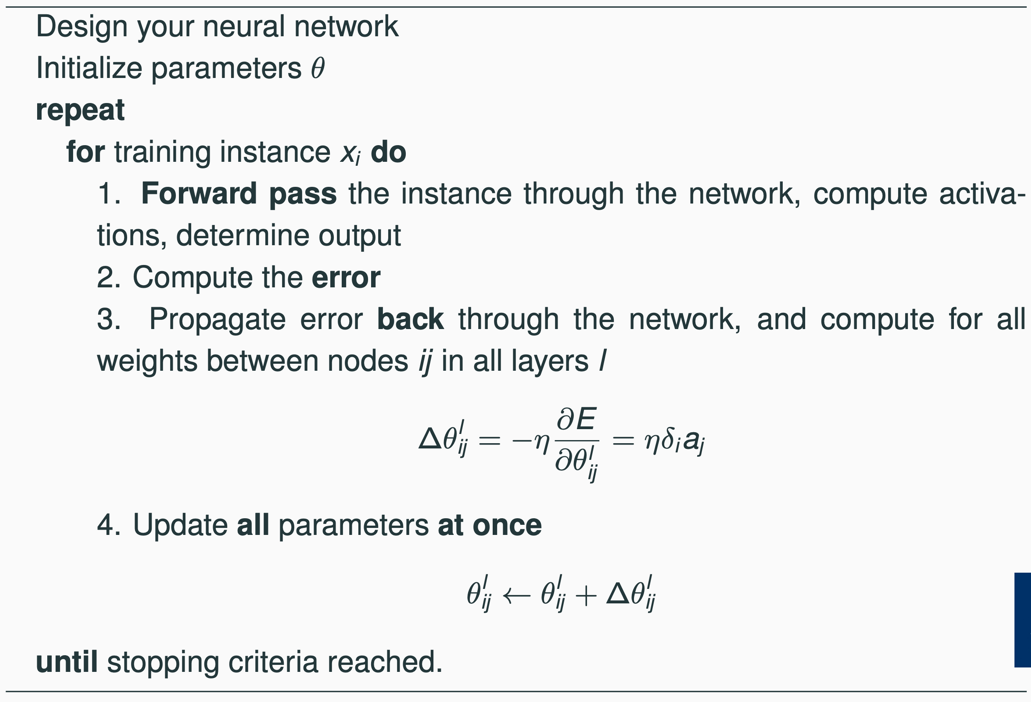
* Regression: mean of square error.
* Binary classification:
* Multi-class classification:

Forward propagation:

Backpropagation: suppose the activation function is sigmoid function for hidden layers and mean square error for output layer.

* Partial derivative w.r.t. the parameter of the output layer:
* Partial derivative w.r.t. the parameter of the layers ahead of output layer:

Recipe of backpropagation algorithm:



**Decision Tree**:

Structure: tree-like structure that classifies an instance based on rules.

图示

描述已自动生成

ID3 algorithm:

* Overview:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

* Entropy: the expected value of self-information that measure the level of surprise of a random variable.
* Mean information: the weighted entropy of sub-nodes.
* Information gain: the difference between entropy of root node and mean information of child nodes.
* Split ratio: to penalize highly branching attributes, we add a normalization term.
* Gain ratio:

Stopping criteria: ① instances of a single node are of the same class; ② the improvement of information gain or gain ratio is less than a predefined threshold; ③ run out of all possible attributes.

**Feature Selection**:

Wrapper method:

* Enumerate all possible subsets, train them and find the best attribute subset.
* Starting from an empty set, choose the best attribute iteratively, until the performance does not improve.
* Starting with all attributes, remove one attribute that causes least performance degradation, until the performance does not degrade.

Feature filtering:

* Pointwise mutual information: measure the correlation between one attribute and class label.
* Mutual information: combine each , , , PMI.
* Chi-square:

where and is observed value and expected value respectively.

**Ensembling Method**:

Approaches to ensemble learning: instance manipulation, feature manipulation, class label manipulation, algorithm manipulation.

Stacking: majority voting(each instance’s prediction is made by multiple base classifier voting), meta classification(train a classifier over the output of base classifier).

Bagging: construct novel datasets through a combination of random sampling and replacement

图形用户界面, 文本, 应用程序, 聊天或短信, 电子邮件

描述已自动生成

Random tree: for each node, only some of attributes are selected for building decision tree; it is helpful to control unhelpful attributes; it is faster but add more variance.

Random forest: multiple random trees ensemble together to build a strong decision tree; each tree is built using different dataset(bagging); it is helpful to minimize model variance without introducing model bias.

Boosting: iteratively change the distribution and weights of training instances based on whether it is classified correctly; base classifier is combined via weighted voting; it is helpful to reduce bias but introduce variance

Suppose we have base classifiers: , initial instance weight , the classifier in iteration is , we compute the error rate such that:

The importance of :

Weight for instance in iteration :

where is normalization factor that let the sum of for all is equal to 1.

Final prediction:

**Unsupervised Learning**:

What is clustering; types of clustering: exclusive vs overlapping, deterministic vs probabilistic, hierarchical vs partitioning, partial vs complete, heterogenous vs homogenous, incremental vs batch.

The measures of unsupervised clustering include cluster cohesion and cluster separation:

The measures of supervised clustering include entropy and purity:

K-means clustering:

图形用户界面, 文本, 应用程序

描述已自动生成

complexity: , where is the number of instances, is the number of attributes, is the number of cluster centroids, is the number of iterations.

Elbow method to select :

图表, 折线图

描述已自动生成

Agglomerative clustering:

图形用户界面, 文本, 应用程序

描述已自动生成

Single link: minimum distance between two points in two clusters

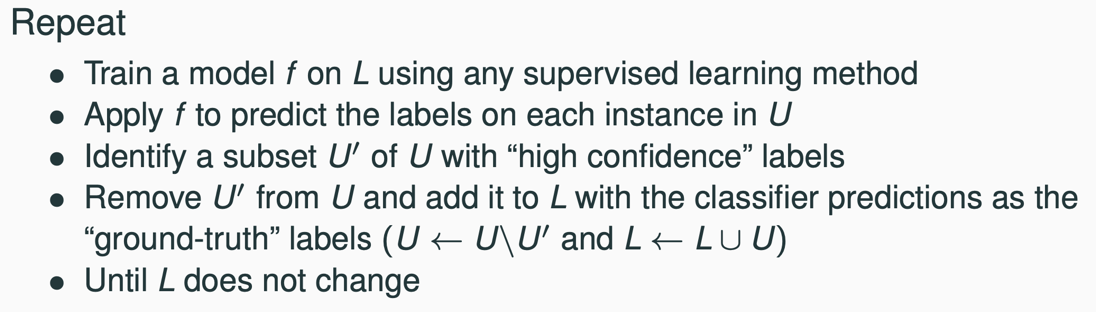
Complete link: maximum distance between two points in two clusters

Group average: average distance between all points in two clusters

**Semi-supervised Learning**:

Algorithm: Let be the set of labelled training instances , be the set of unlabeled training instances .

Self-training:



If the labelled instances’ confidence is below the threshold, it will move back to unlabeled pool.

Active learning: an active classifier that can pose queries for labelling by an oracle. Typically, the most uncertain instance will be sent to oracle for labelling.

Query strategies:

① : choose instance with the smallest confidence

② , where and are the first and second most probable labels for .

③ : choose instance with the largest entropy.

④ Query by committee: train multiple classifiers, and query instances with the highest disagreement measured by entropy.

What is data augmentation.

**Anomaly Detection**:

What is anomaly; what is anomaly detection.

Structure of anomalies: global anomaly, contextual anomaly, collective anomaly.

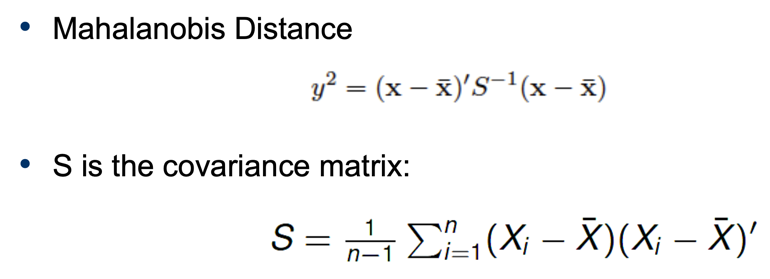
Supervised anomaly detection (has labels for both normal and anomaly data) and its challenges.

Semi-supervised anomaly detection (labels only for normal data) and its challenges.

Unsupervised anomaly detection (cluster normal objects) and its challenges.

Statistical anomaly detection: learn a model fitting the given dataset and identify those objects that lay in the low probability region. Typically, we have univariant, multivariant Gaussian distribution, where data not in the range of is treated as anomalies.

Mahalanobis distance:

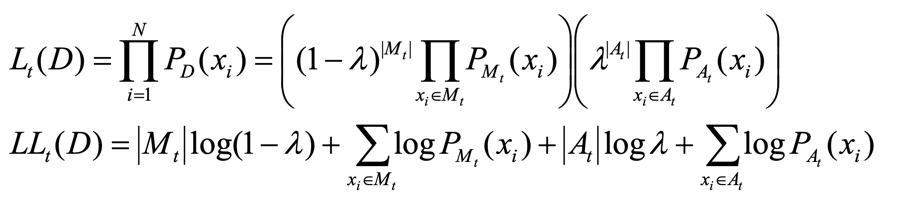


Likelihood approach: suppose we have two distributions: one is majority distribution, another is anomalous distribution.

文本

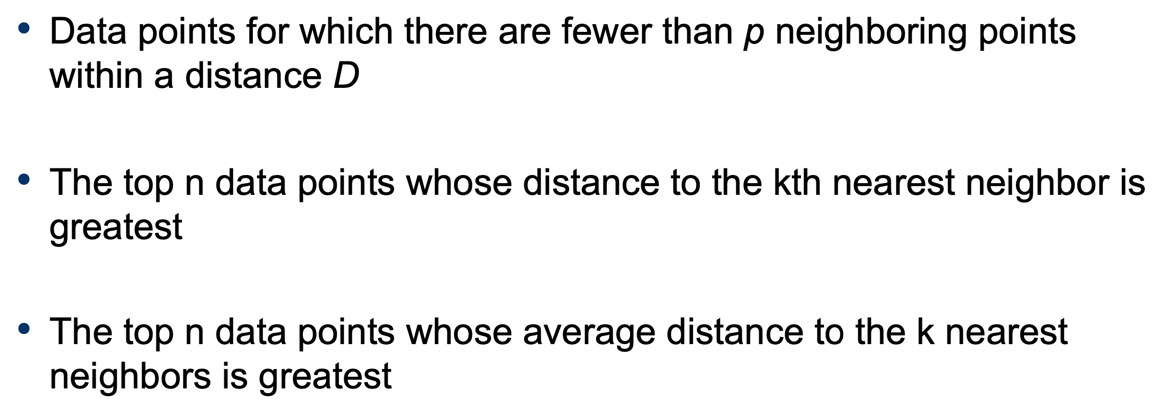
描述已自动生成

where overall data distribution is . Data likelihood at time :



Proximity-based anomaly detection: An object is an anomaly if the nearest neighbor(typically -th nearest neighbor) is far away.

Three ways to define outliers in terms of proximity:



Density-based anomaly detection: anomaly object is in low-density region.

Density is the average distance to nearest neighbors:

Cluster-based anomaly detection: anomalous points are points that do not belong to any clusters.

Assess degree to which object belongs to any cluster:

文本

描述已自动生成

Eliminate objects: remove objects which most improve objective function.

Discard small clusters far from other clusters.

**Association Rule**:

Itemset: a collection of one or more items; support count(): frequency of occurrence of an item set; support: fraction of transaction that contains an itemset(); confidence: fraction of items in one itemset in transaction in another itemset(); frequent itemset: itemset whose support value is greater than .

Find association rules:

* Brute-force approach: enumerate all possible itemset prune those whose support and confidence less than a threshold. Complexity: .
* Frequent itemset generation: generate itemset with support value greater than , then generate association rule whose confidence value is greater than .

Complexity:

Apriori principle(anti-monotone property):

图形用户界面, 文本

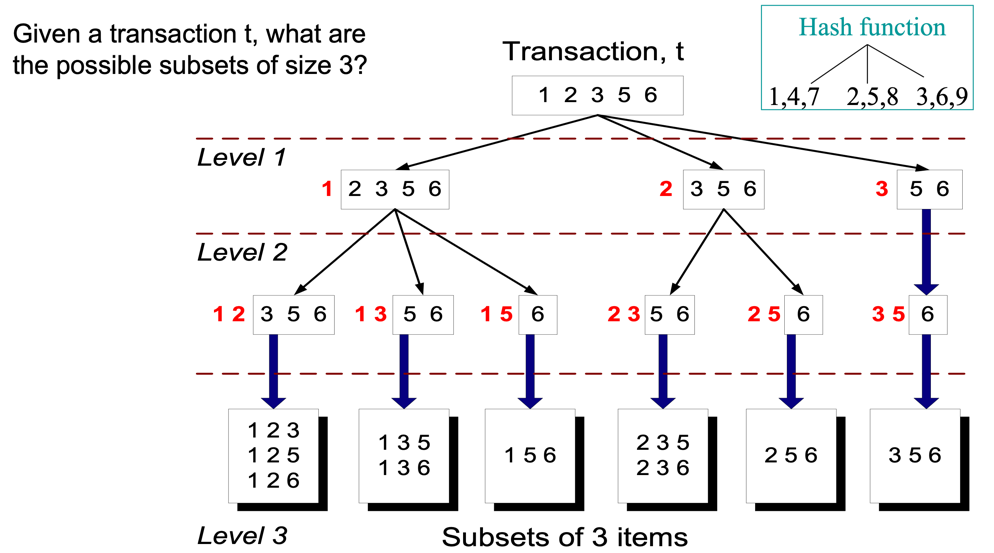
描述已自动生成

To reduce the number of comparisons, we store the candidates in a hash structure, where each transaction compares with the candidate in the hash structure.

图示

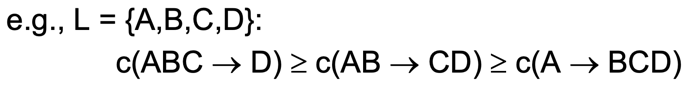
描述已自动生成

Subset stored by hash function:



If the length of frequent itemset is , the number of candidate association rules is .

Anti-monotone property of association rule:



**Recommender System**:

Goals: relevance, novelty, serendipity, diversity.

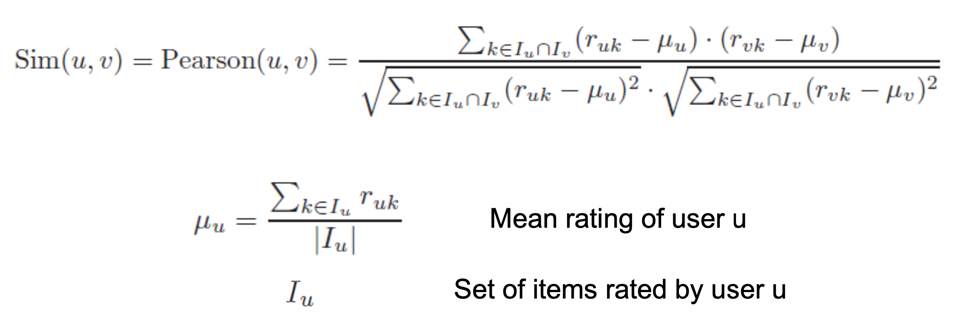
Content-based recommendation: items are described as features , users are also described as features . We use cosine similarity to measure whether we recommend item to user :

Collaborating filtering: predict user preferences by collecting taste information from many other users.

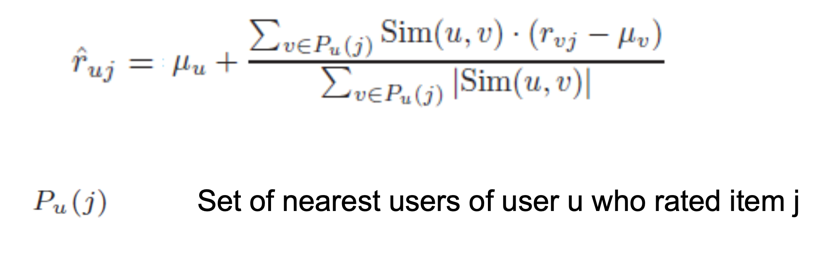
User-based model: similar users have similar rating on similar items.

Suppose we have a rating matrix : represents rating by user for item .

Pearson correlation:

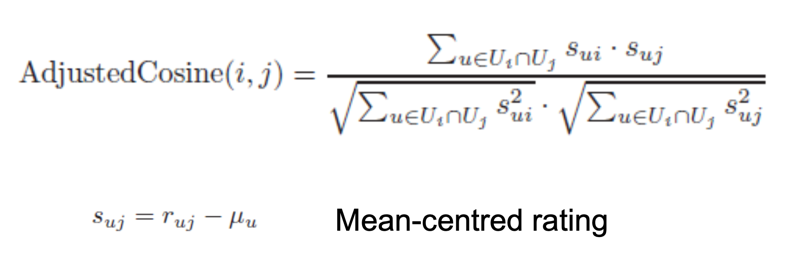


Prediction of rating of user for item :



Item-based model: similar items are rated in similar way by the same user.

Adjusted cosine similarity:



Prediction:

图片包含 文本

描述已自动生成