Machine Learning

Supervised learning 1

S. Nõmm

[1]Department of Software Science, Tallinn University of Technology

20.02.2024

Supervised learning

Is a task of inferring function (training a model) on the basis of labeled training data. The goal is to construct a function (train a model) which would mimic (in a certain sense) behaviour of the underling process.

- Classification labels are discrete (categorical values).
  - k-nearest neighbours.
  - Decision trees.
  - Support Vector Machines.
  - Neural networks.
  - Ensemble (committee).
  - Boosted techniques.
- Regression: Dependent variable (continuous) plays a role of labels.
  - Linear
  - Nonlinear
  - Application of trees and SVM for regression.
  - Advanced methods like Neural Networks, etc.
- Markov models.

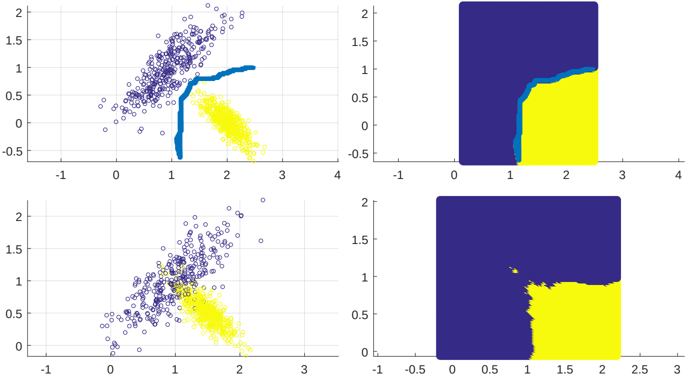
Classification

- Learning existing grouping on the basis of the labeled (training) set.
- The goal is to generate (choose the structure and train) a model which would mimic existing grouping.
- Based on the features of the element model should estimate which class element belong to or estimate value of dependent variable.
- Unlike the case of unsupervised learning miss classification may be precisely measured.
- What is the cost of miss classification or error in the case of regression?

k - nearest neighbours (k-NN)

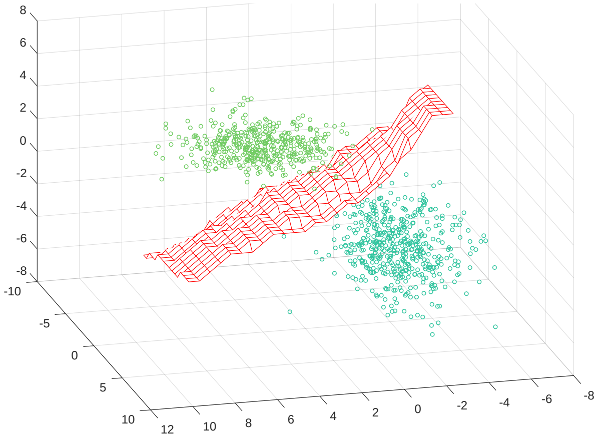
- Let D denote training (labeled) data set.
- For each unlabeled point (point to be classified)
  - Find k - nearest neighbours.
  - Assign mode (majority) label of k - nearest neighbours.

k - nearest neighbors, geometric interpretation, 2D



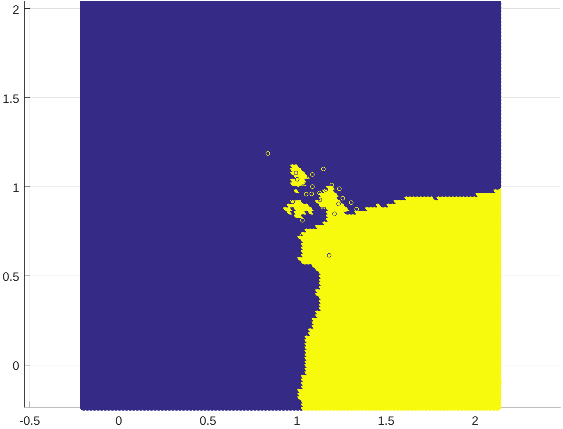
- Decision boundary (decision surface) (statistical classification with two classes) is a hypersurface that partitions the data set into two subsets, one for each class.
- Classifier tries to learn (construct) decision boundary that will lead minimal empirical error.

k - nearest neighbors, 3D



Accuracy

During the training (learning) process classifier tries to learn (construct) decision boundary that will lead minimal empirical error.



How good is trained classifier?

Validation

- Overall accuracy and Confusion matrix (table),computed for the validation subset, are the goodness parameters of trained classifier.

	Predicted Class 1	Predicted class 2
Actual class 1	58	2
Actual class 2	6	134

- How reliable these parameters are ?

Learning: Underfitting and overfitting

- *Underfitting* the learned function is too simple In the context of human learning: underfitting similar to the case when one learns too little.
- *Overfitting* the learned function is too complex In the context of human learning: overfitting is more similar to memorizing than learning.

Feature selection for classification

- Case of categorical data: Gini Index or Entropy. Value specific:

  $$G(v_i) = 1 - \sum_{j=1}^{k} p_j^2; \qquad E(v_i) = -\sum_{j=1}^{k} p_j \log_2(p_j)$$

  where p_j is the fraction of data points containing attribute value v_i. Lower values of Gini index or Entropy imply greater discriminative power.

- Feature specific: Let n_i is the number of data points taking value v_i. Feature specific Gini index is defined as the weighted average value of value specific Gini indexes.

  $$G = \sum_{i=1}^{r} \frac{n_i G(v_i)}{n}$$

  where r is the number of different values v_i and $n = \sum n_i$.

- Feature specific values of Entropy are computed in the similar way.

Feature selection for classification II

- Case of numeric data: Fisher's score

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j \sigma_j^2}$$

  Greater values imply greater discriminative power of the variable.

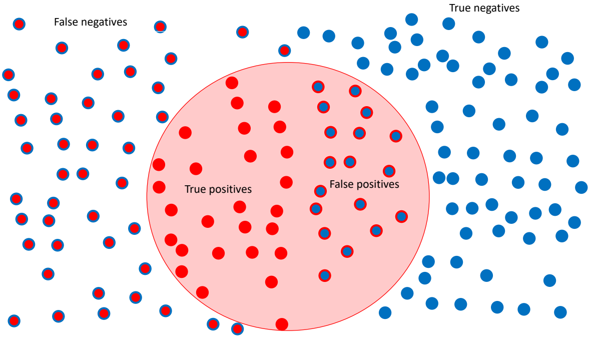
- Wrapper methods.

Classification model goodness!

- How good is the model?
- What is the goal of modeling?

Classification outcome

- Consider binary classifier.
- In the data set there are two classes: Positive (P) and negative (N)
- Outcomes of the classification: True positive, true negative, false positive (type I error), false negative(type II error).



Context of information retrieval

NB! Observe notions!

- Relevant elements of the data set. One is interested to find (retrieve elements of the certain class).
- Precision is defined as:

$$\text{precision} = \frac{|\text{relevant} \cap \text{retrieved}|}{|\text{retrieved}|}$$

- Recall (sensitivity, hit rate, True Positive Rate) is defined as:

$$\text{recall} = \frac{|\text{relevant} \cap \text{retrieved}|}{|\text{relevant}|}$$

Context of classification I

Denote: tp - true positive, tn - true negative, fp - false positive and fn - false negative.

- Precision (positive predictive value):

$$\text{Precision} = \frac{tp}{tp + fp}$$

- Recall (sensitivity, hit rate, TPR):

$$\text{Recall} = \frac{tp}{tp + fn}$$

- True negative rate (Specificity, selectivity):

$$\text{TNR} = \frac{tn}{tn + fp}$$

- Accuracy:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

- Predicted positive condition rate

$$\text{Predicted} \quad \text{positive} \quad \text{condition} \quad \text{rate} = \frac{tp + fp}{tp + tn + fp + fn}$$

F-measure *not to be confused with similarly named values!!!*

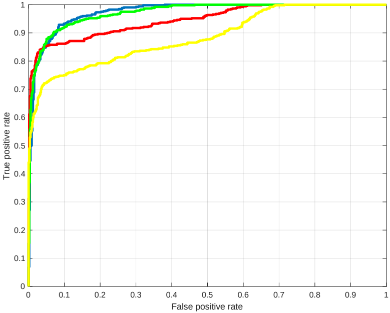
Frequently referred as F_1-score ... is harmonic average of precision and recall.

- $$F = 2 * \frac{\text{precisionrecall}}{\text{precision} + \text{recall}}$$
- More general definition:

$$F_\beta = (1 + \beta^2)\frac{\text{precisionrecall}}{\beta^2\text{precision} + \text{recall}}$$

Receiver Operating Characteristic or ROC curve

- Let $\mathcal{D} = \{x_i, y_i\}$ is the labeled data set.
- Assume also that $\delta(x) = \mathbb{I}(f(x) > \tau)$ - decision rule. $f(x)$ is the confidence function and τ threshold parameter
- Each particular value of τ corresponds to a certain decision rule.
- For each decision rule one may compute recall and false positive rate.
- Associate recall values with the axis Y and false positive rate values with axis X.



Cross validation

- Non-exhaustive do not use all possible ways of splitting into training and validation sets
  - k - fold.
  - Holdout.
  - Repeated random sub-sampling.
- Exhaustive: use all possible ways to divide the data set into training and validation sets
  - Leave p-out cross validation.
  - Leave one out cross validation.

Cross validation: k- fold validation

- Divide the training data (after removing test data) randomly into k - folds.
- Perform following k experiments:
    - Compose the training data by concatenating k-1 folds leaving one fold out.
    - Train the model on those k-1 folds
    - Test it on the left-out fold
    - Record the result
- Report the average of the k experiments.
- Nested vs non-nested cross validation.

Decision trees

- Non-parametric supervised learning technique.
- Tree-like graph is used to represent the model of decision making and possible consequences of such decisions.
- Internal nodes are conditions (questions). terminal nodes represent labels of classes.
- Questions or conditions play a role of features. Answers to the questions are referred as feature values.
- Training a tree model is referred as *tree growing*.

Growing a tree 1

Greedy heuristic is the most popular technique. Let F be the possible set of features and S is the subset of data. The idea is to find most useful feature (among remaining) at each node.

$$j(S) = \arg\min_{j \in F} \mathrm{cost}(\{x_i, y_i : x_i \in S, x_{i,j} = c_k\}) \\ + \mathrm{cost}(\{x_i, y_i\, x_i \in S, x_{i,j} \neq c_k\})$$

Classification cost:

$$\hat{\pi}_c = \frac{1}{|S|} \sum_{x_i \in S} \mathbb{1}\{y_i = c\}$$

Misclassification rate:

$$\frac{1}{|S|} \sum_{x_j} \in S \mathbb{1}(y_i \neq \hat{y}) = 1 - \hat{\pi y}$$

Cost functions

- Entropy:

$$\mathbb{H}(\hat{\pi}) = -\sum_{c=1}^{C} \hat{\pi}_c \log_2 \hat{\pi}_c$$

  Minimizing entropy is equivalent to maximizing information gain which is $\mathbb{H}(Y) - \mathbb{H}(Y|X_j)$.

- Gini index:

$$G = \sum_{c=1}^{C} \hat{\pi}_c (1 - \hat{\pi}_c)$$

Growing a tree 3

- Repeat:
    - For each feature divide data into corresponding subsets. Evaluate accuracy of such split with respect to response variable.
    - "Most accurate" feature wins. It will become condition at a given node.
    - Exclude chosen feature from the feature set.
- Until no more features left.

Example: When to play tennis

Outlook	Temperature	Humidity	Wind	Play
sunny	warm	high	weak	no
sunny	warm	high	strong	no
rain	warm	high	weak	yes
rain	cool	normal	weak	yes
rain	cool	normal	strong	no
sunny	cool	normal	strong	yes
sunny	warm	high	weak	no
sunny	cool	normal	weak	yes
rain	warm	normal	weak	yes
sunny	warm	normal	strong	yes
rain	warm	high	strong	yes
sunny	warm	normal	weak	yes
rain	warm	high	strong	no

Information gain

Definition

Information gain G_I of an action is the decrease of the ambiguity achieved as the result of the action.

- In the context of decision tree growing the action is splitting the node.
- If entropy is chosen as the cost function then information gain is defined as follows:

$$G_I = E - (E_l \cdot p_l + E_r \cdot p_r)$$

where E is the entropy before splitting E_l is the entropy of left child and E_r is the entropy of the right child. Indexes r and l have the same meaning for the proportions p.

Growing the tree: case of continues features

Denote X the matrix where columns correspond to different features and rows correspond to the different observation points.

- If all the data points are of the same class return the leaf node that predicts this class.
- Among all splitting points for each column find the one giving largest information gain.
- Then chose the column with the maximum gain.
- Perform splitting.
- If stopping criteria is satisfied return the tree.
- If stopping criteria is not satisfied apply tree growing procedure to each child.

Pruning

- In order prevent overfitting stop growing the tree when the decrease is not sufficient to justify adding extra subtree.
- Grow a full tree and then prune the branches giving less decrease in error.

Bayes theorem

- Let us suppose that there k classes are given.
- The *posterior probability* of a class C_k for an input x is:

$$p(C_k \mid x) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(x)}$$

- $p(\boldsymbol{x} \mid C_k)$ is the likelihood, $p(C_k)$ is the *prior probability*, $p(x)$ is the *marginal data likelihood*.
- $p(C_k)$ is the probability of a class k *a priori*, before getting any knowledge about the data.
- $p(C_k \mid \boldsymbol{x})$ is the class probability *a posteriori*, after getting knowledge about the data.
- Bayes theorem updates prior distribution into posterior on the basis of empiric information.

Conditional and unconditional independence

- If X and Y are *unconditionally independent* then their joint distribution is the product of the marginal distributions:

$$X \perp Y \Leftrightarrow p(X,Y) = p(X)p(Y)$$

- If the influence is mediated through a third variable Z, then X and Y are said to be *conditionally independent*

$$X \perp Y \mid Z \Leftrightarrow p(X,Y \mid Z) = p(X \mid Z)p(Y \mid Z)$$

- Conditional independence does not imply unconditional independence and vice versa:

$$X \perp Y \mid Z \not\Leftrightarrow X \perp Y$$

Feature representation

- Amount of the training data may pose a problem in computing likelihood $p(\boldsymbol{x} \mid y)$. (Low amout of training data may prevent reliable computation of the likelihood).
- Consider the document as the set of words
- for the given vocabulary V present each document as a binary vector.
- If word belong to the vocabulary corresponding element take the value 1 and 0 otherwise.
- This approach will lead to the following likelihood function

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{|V|} p(x_j \mid y)$$

Naïve Bayes assumption

- Likelihood is computed as:

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{n} p(x_j \mid y)$$

- *Naïve Bayes assumption:* the features are conditionally independent given the class label.
- the word *naïve* reveres to the fact that actually features are not expected to be independent or conditionally independent.
- Model has relatively few parameters and therefore immune to overfilling.

Naïve Bayes model

- Parameters of the model

$$\begin{aligned}\theta_{j|y=1} &= p(x_1 = 1 \mid y = 1)\\ \theta_{j|y=0} &= p(x_1 = 1 \mid y = 0)\\ \theta_y &= p(y = 1)\end{aligned}$$

- The MLE estiamtes of the parameters are:

$$\begin{aligned}\theta_{j|y=1} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 1)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}\\ \theta_{j|y=0} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 0)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 0)}\\ \theta_y &= \frac{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}{m}\end{aligned}$$

Prediction with naïve Bayes model

- the goal is to find wether a new element is of class 1 or 0 (in the example of spam filtering wether given e-mail message is spam or not).
- According to Bayes theorem.

$$p(y=1\mid \boldsymbol{x},\boldsymbol{\theta}) \propto p(\boldsymbol{x}\mid y,\boldsymbol{\theta})p(y\mid\boldsymbol{\theta}) = p(y=1\mid\theta)\prod_{j=1}^{n}p(x_{i,j}\mid y=1,\boldsymbol{\theta})$$

$$p(y=0\mid \boldsymbol{x},\boldsymbol{\theta}) \propto p(\boldsymbol{x}\mid y,\boldsymbol{\theta})p(y\mid\boldsymbol{\theta}) = p(y=0\mid\theta)\prod_{j=1}^{n}p(x_{i,j}\mid y=0,\boldsymbol{\theta})$$

- Predict the class with highest posterior probability:

$$y^{*} = \arg\max_{y\in\{0,1\}} p(y\mid\boldsymbol{x},\boldsymbol{\theta})$$