

Fiche Machine Learning

Pierre Colson

Contents

General Stuff	1
Nearest Neighbour	1
Decision Trees	2
Challenge in machine learning	2
Regression	3
Learning as Inference	4
Other	5

Markdown version on [github](#)

Compiled using [pandoc](#) and [gpdf script](#)

More fiches [here](#)

General Stuff

- **Supervised learning**, learning from humain supervision
- **Unsupervised Learning**, learn without humain supervision
- Step for Classification :
 - **Trainig phase**: to give the concept of classes to a machine using labeled data
 - **Testing phase**: to determine the calss of new unseen (unlabeled) data

Nearest Neighbour

- Binary classification
- Find the nearest neighbour, and classify x to the same class
- k -nearest neighbour:
 - Algo:
 - * Compute distances to all samples from new data x
 - * Pick k -neighbours that are nearest to x
 - * Majoruty vote to classify x
 - Other
 - * The boundary becomes smoother as k increases
 - * Lower computational cost for lower k

- * k -NN better generalizes given many samples
- Pros
 - * Simple, only with a single parameter k
 - * Applicable to multi-class problems
 - * Good performance, effective in low dimension data
- Cons
 - * Costly to compute distances to search for the nearest
 - * Memory requirement: must store all the training set

Decision Trees

- Test the attributes (features) sequentially
- Each leaf node bears a category label, and the test pattern is assigned the category of the leaf node reached.
- Tree Construction:
 - Choose the best question (according to the information gain), and split the input data into subsets
 - Terminate: Call branches with a unique class labels leaves
 - Grow: Recursively extend other branches
- Entropy - measure of uncertainty - number of bit of information

$$ENTROPY = \sum_i -p_i \log_2 p_i$$

- Information gain:

Ask about attribute A for a data set S that has Entropy $ENT(S)$ and get subsets S_v according to the value of A

$$GAIN = ENT(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} ENT(S_v)$$

- Gini impurity: Another definition of predictability (impurity)

$$\sum_i p_i(1 - p_i) = 1 - \sum_i p_i^2$$

- Overfitting, when the learned models are overly specialized for the training samples (Good result on training data, but generalizes poorly)
- The simplest explanation compatible with data tends to be the right one
- To avoid overfitting we can use validation set and pruning. Pruning means simplifying/compressing and optimizing a decision tree by removing sections of the tree that are uncritical and redundant to classify instances.

Challenge in machine learning

- The misclassification of a model f rate on a training data D :

$$err(f, D) = \frac{1}{N} \sum_{i=1}^N Ind(f(x_i) \neq y_i)$$

- Overfitting occurs due to:
 - Non-representative sample
 - Noisy examples
- K -fold cross validation (training set and validation set)
- Intuitions in low-dimensions do not apply in high-dimensions. Techniques for dimensionality reduction / feature selection exist.
- Error due to **Bias**: the difference between the average (expected) prediction of our model and the correct value
 - It is the discrepancy between its averaged estimated and true function
 - Low model complexity \implies High-bias
 - High model complexity \implies Low-bias
- Error due to **Variance**: The variability of a model prediction for a given data point between different realizations of the model
 - It is the expected divergence of the estimated prediction from its average value.
 - Low model complexity \implies Low-variance
 - High model complexity \implies High-variance
- $MSE = Variance + Bias^2$

Regression

- Regression = Real valued output
- Linear Regression tries to estimate the function $f(x)$ and predict the output by

$$\hat{f}(x) = \sum_{i=0}^d w_i x_i = w^T x$$

- To measure the error for N samples we use the mean square error

$$E_{in}(\hat{f}) = \frac{1}{N} \sum_{n=1}^N (\hat{f}(x_n) - y_n)^2$$

- The weight vector that sets the gradient to zero minimizes the errors. RSS is the sum of squared errors

$$w = (X^T X)^{-1} X^T Y$$

- RANSAC: RANdom SAMpling Consensus
 - Randomly select a (minimum number of) sample of s data points from S and instantiate the model from this subset
 - Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of samples and defines the inliers of S
 - If the subset is S_i is greater than some threshold T , re-estimate the model using all the points S_i and terminate
 - If the size of S_i is less than T select a new subset and repeat the above
 - After N trials the largest consensus set S_i is selected and the model is re-estimated using all the points in the subset S_i

Cost:

- RANSAC can be vulnerable to the correct choice of the threshold
- k -NN Regression (non parametric)
 - Similar to the k -NN classifier
 - To regress Y for a given value of X , consider k closest point to X in training data and take the average of the responses

$$f(x) = \frac{1}{k} \sum_{x_i \in N_i} y_i$$

- Larger values of k provide a smoother and less variable fit (lower variance)
- Parametric vs non parametric
 - If the parametric form is close to the true form of f , the parametric approach will outperform the non parametric
 - As a general rule parametric methods will tend to outperform nonparametric when there is a small number of observations per predictor
 - Interpretability standpoint: Linear regression preferred to KNN if the test MSEs are similar or slightly lower
- Ridge regression
 - Similar to least squares but minimizes different quantity instead of sum of squared

$$RSS + \lambda \sum_{i=1}^d w_i^2$$

- The Lasso
 - Similar to ridge regression but with slightly different term

$$RSS + \lambda \sum_{i=1}^d |w_i|$$

Learning as Inference

- Classification: Y is discrete
- Regression: Y is continuous
- $Pr(Y = y) \equiv Pr(y) \leftarrow$ Prior
- $Pr(x | Y = y) \equiv Pr(x | y) \leftarrow$ Likelihood
- $Pr(X = x) \equiv Pr(x) \leftarrow$ Evidence
- $Pr(y | X = x) = \frac{Pr(X|Y=y)Pr(Y=y)}{Pr(X=x)} \leftarrow$ Posterior
- Parametric Inference:
 - Estimate θ using D
 - Compute $Pr(y | x, \hat{\theta})$ to make inference
 - Learning corresponds to estimate θ

Approaches:

- Maximum Likelihood (ML) Estimation

- Maximum A Posteriori (MAP) Estimation
- Non-parametric Inference
 - Estimate $Pr(\theta | D)$
 - Compute $Pr(y | x, D)$ from $Pr(y | x, \theta, D)Pr(\theta | D)$ by marginalizing out θ

Approaches:

- Bayesian methods
- Assumption: Observations are independent and identically distributed
- Maximum Likelihood Estimate

Other

- Classification is discrete
- Regression is continuous
- Probabilistic learning involves estimating $P(x, y)$ from a Dataset
- Probabilistic learning cannot be used to create generative model
- Naive Bayes classifier is a generative model (generative model learns joint probability distribution)
- Logistic regression is a discriminative model (discriminative model learns conditional probability distribution)
- For MLE we assume all observations are independently and identically distributed given y
- An artificial neuron generates an output signal based on the integrated weighted input
- Structural Risk Minimization is selecting a separating hyperplane such that future data is most likely classified correctly
- Boosting is : Each training example has a weight which is re-weight through iterations
- Covariance matrix of given vectors is a useful tool to apply the maximum variance in PCA
- Expectation Maximisation : Algorithm to learn with latent variable
- Posterior probability : Conditional probability taking into account the evidence
- RANSAC : Robust method to fit a model to data with outliers
- Dropout : A method for preventing artificial neural networks from overfitting
- The Lasso : An approach to regression that results in feature selection
- Bagging : Bootstrap aggregating
- Error back propagation : Algorithm to train artificial neural networks
- k-fold cross validation : A technique for assessing a model while exploiting available data for training and testing
- In Decision Forests, randomness is used in feature selection at each node and in generating bootstrap replicas
- In PCA and subspace method, x should belong to the class where the projection length to the corresponding subspaces is maximised
- Maximum likelihood maximizes the probability of the observation conditioned on the class
- Naive Bayes assumes all D dimensions of an observation are conditionally independent given Y

- perceptron learning stop modifying weights when all training data is correctly classified
- The kernel function correspond to the scalar product between two data points transformed into a higher dimensional space
- Adaboost algorithm : A weight is given to each training sample, and it is iteratively updated
- The main purpose of PCA is to reduce the effective number of variable