Fouille de données

▷ Ensemble methods – Succeding together!

P. Lenca[λ] & S. Lallich[β]
[λ] Institut Telecom, Telecom Bretagne UMR CNRS 6285 Lab-STICC
[β] University of Lyon, Lyon 2 Laboratory ERIC

2015-2016

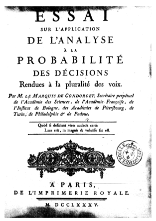
Outline

Outline

Condorcet's jury theorem [dC85]

Political science theorem

- about the relative probability of a given group of individuals arriving at a correct decision.
- consider a jury of k independent judges which has to choose between two outcomes (k is odd) by majority vote
- the error risk of each judge is p
- the number of bad decisions has a binomial distribution $B(k, p)$



ESSAI
SUR L'APPLICATION
DE L'ANALYSE
A LA
PROBABILITÉ
DES DÉCISIONS
Rendues à la pluralité des voix.

A PARIS,
DE L'IMPRIMERIE ROYALE.

$\hookrightarrow$ The error risk of the jury is equal to $Pr(B(k, p) > [k/2])$.

Committees of Classifiers

k independent classifiers (k is odd)

- the committee of classifiers reach a decision by majority vote.
- the error risk of each classifier is supposed to be p
- the error risk of the committee is equal to $Pr(B(k,p) > [k/2])$

$\hookrightarrow$

- if $p < 0.5$ (weak classifier), it is advantageous to have several classifiers
- if $p = 0.5$, it is useless to have multiple classifiers
- if $p \geq 0.5$ the committee is worse than each classifier! If the classifiers are bad, it is preferable to have one classifier

Evaluation & Credidibility Issues

Example

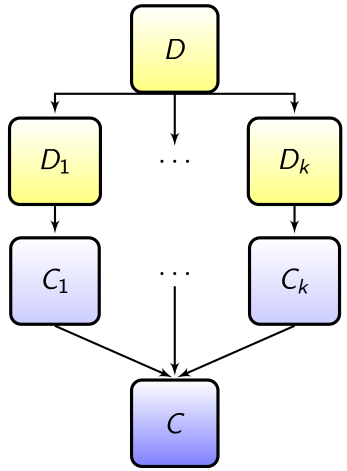
To obtain an error risk of the jury less than 0.05:

- p = 0.20: 7 classifieurs are enough
- p = 0.35: 27 classifiers are needed

Error rate of the committee according to p and k

k \ p	0,05	0,1	0,15	0,2	0,3	0,35	0,5	0,65	0,7	0,8	0,85	0,9	0,95
1	0,05	0,10	0,15	0,20	0,30	0,35	0,50	0,65	0,70	0,80	0,85	0,90	0,95
3	0,01	0,03	0,06	0,10	0,22	0,28	0,50	0,72	0,78	0,90	0,94	0,97	0,99
5	0,00	0,01	0,03	0,06	0,16	0,24	0,50	0,76	0,84	0,94	0,97	0,99	1,00
7	0,00	0,00	0,01	0,03	0,13	0,20	0,50	0,80	0,87	0,97	0,99	1,00	1,00
9	0,00	0,00	0,01	0,02	0,10	0,17	0,50	0,83	0,90	0,98	0,99	1,00	1,00
11	0,00	0,00	0,00	0,01	0,08	0,15	0,50	0,85	0,92	0,99	1,00	1,00	1,00
15	0,00	0,00	0,00	0,00	0,05	0,11	0,50	0,89	0,95	1,00	1,00	1,00	1,00
19	0,00	0,00	0,00	0,00	0,03	0,09	0,50	0,91	0,97	1,00	1,00	1,00	1,00
23	0,00	0,00	0,00	0,00	0,02	0,07	0,50	0,93	0,98	1,00	1,00	1,00	1,00
27	0,00	0,00	0,00	0,00	0,01	0,05	0,50	0,95	0,99	1,00	1,00	1,00	1,00
31	0,00	0,00	0,00	0,00	0,01	0,04	0,50	0,96	0,99	1,00	1,00	1,00	1,00
35	0,00	0,00	0,00	0,00	0,01	0,03	0,50	0,97	0,99	1,00	1,00	1,00	1,00
39	0,00	0,00	0,00	0,00	0,00	0,03	0,50	0,97	1,00	1,00	1,00	1,00	1,00
43	0,00	0,00	0,00	0,00	0,00	0,02	0,50	0,98	1,00	1,00	1,00	1,00	1,00
47	0,00	0,00	0,00	0,00	0,00	0,02	0,50	0,98	1,00	1,00	1,00	1,00	1,00
51	0,00	0,00	0,00	0,00	0,00	0,01	0,50	0,99	1,00	1,00	1,00	1,00	1,00

Ensemble approach



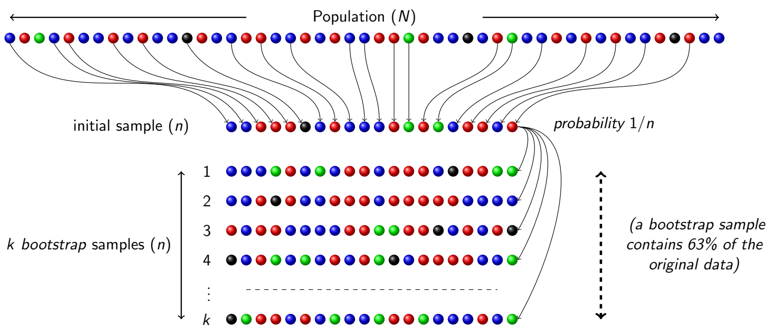
Ensemble approach

Two types of methods

- Homogeneous: train the same classifier under different contexts
  (modify the original dataset by resampling (step 2), changing some parameters of the algorithm)
- Heterogeneous: train different type of classifiers on the same dataset
  (use various classifiers such as neural networks, decision trees, regression trees, linear regression)

Outline

Bootstrap sample [Efr79]



- the probability of an example not to be selected after n drawings is $(1-1/n)^n \sim e^{-1}$ (if n is large enough); the probability of an example to be selected after n drawings is $1-e^{-1} = 0.632$

$\hookrightarrow$ It is very easy to construct automatically many bootstrap samples from the same learning set.

Bootstrap sample

The population is to the original sample as the original sample is to each bootstrap sample.

- bootstrap has the advantage to conserve the covariance structure of the variables
- bootstrap is very useful to simulate sampling fluctuations
- training a classifier on multiple bootstrap samples of the original sample allows to simulate training on multiple samples of the population

Outline

Bagging [Bre94]

Multiple classifiers from an original dataset.

- training: construct each classifier from a bootstrap sample of the data
- prediction:
  - classification (categorical class variable): assign the class according to majority vote of the set of classifiers
  - regression (numerical class variable): make the final prediction by averaging the prediction of the different classifiers
- training a classifier on multiple bootstrap samples of the original sample allows to simulate training on multiple samples of the population

Bagging

- works well because it reduces variance by voting/averaging
- usually, the more classifiers the better
- is very useful if data are noisy
- works well if the classifier is unstable (neural networks, decision trees, regression trees, linear regression, unlike k-nearest neighbours which is stable); An algorithm is unstable if perturbing the learning set can induce significant changes in the classifier constructed

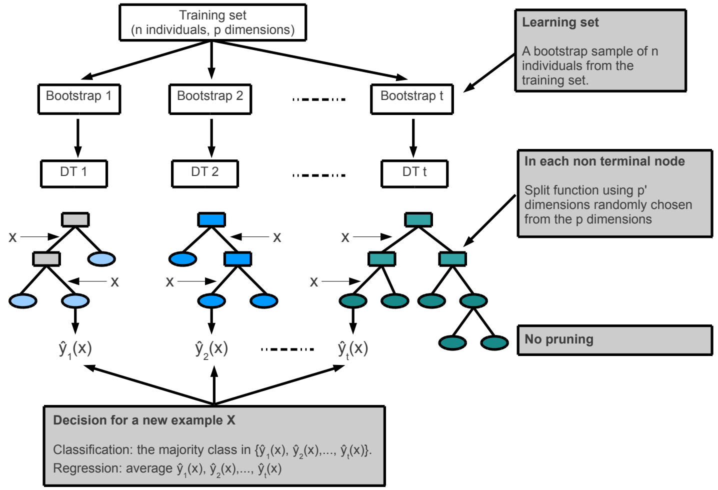
Outline

Random Forests [Bre01]

Multiple decision trees from an original dataset.

- training: construct each tree from a bootstrap sample of the data, split each node of the tree from random subset of attributes [Ho95, AG97, Ho98], without pruning
- prediction:
  - classification (categorical class variable): assign the class according to majority vote of the set of trees
  - regression (numerical class variable): make the final prediction by averaging the prediction of the different trees

Random Forests



Random Forests

- The performance of RF is even higher than:
  - each tree classifier is accurate
  - the different tree classifiers are diverse, i.e. they missclassify different examples
- RF performance is as good as the one of boosting, thanks to the random selection of the features at each split
- RF performs sometimes better than boosting because it is relatively robust to outliers and noise
- RF is faster than bagging or boosting. It is simple and can be easily parallelized

Random Forests

Out Of the Bag.

In random forests, there is no need for holdout or cross-validation to get an unbiased estimate of the error. This one is estimated internally, during the run, as follows:

- for each tree, about one-third of the cases are left out of the Bootstrap sample and are not used to construct the tree, so, a test set classification is obtained for each case in about one-third of the trees
- the predicted class of a case is the class that got most of the votes every time the case was OOB
- the proportion of times that the predicted class is not equal to the true class averaged over all cases is the OOB error estimate; OOB estimate is generally unbiased, except if

Outline

Boosting [Sch89, Fre90]

Combine many weak classifiers to produce a strong ensemble of classifiers (committee).

- iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier
- weak classifiers are typically weighted in some way that is usually related to the weak learners' accuracy
- after a weak learner is added, the data is reweighted: misclassified examples gain weight and examples classified correctly lose weight; Thus, future weak learners focus more on the examples that previous weak learners misclassified

$\hookrightarrow$ The main difference between boosting algorithms is their method of weighting training data points and hypotheses.

Boosting

- compared with bagging, boosting tends to achieve greater accuracy
- but it is sensitive to noisy data and outliers
- it can be less susceptible to the overfitting problem

Outline

References I

[AG97] Y. Amit and D. Geman.
Shape quantization and recognition with randomized trees.
Neural Computation, 9(7):1545–1588, 1997.

[Bre94] Leo Breiman.
Bagging predictors.
Technical Report 421, Department of Statistic, University of California, 1994.

[Bre01] L. Breiman.
Random forests.
Machine Learning, 45(1):5–32, 2001.

[dC85] Marquis de Condorcet.
Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.
1785.

[Efr79] Bradley Efron.
Bootstrap methods: Another look at the jackknife.
The Annals of Statistics, 7:1–26, 1979.

[Fre90] Yoav Freund.
Boosting a weak learning algorithm by majority.
In Mark A. Fulk and John Case, editors, *COLT*, pages 202–216. Morgan Kaufmann, 1990.

TELECOM Bretagne

References II

[Ho95] T. K. Ho.
Random decision forest.
In *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[Ho98] T. K. Ho.
The random subspace method for constructing decision forests.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):832–844, 1998.

[Sch89] Robert E. Schapire.
The strength of weak learnability (extended abstract).
In *FOCS*, pages 28–33. IEEE Computer Society, 1989.

TELECOM Bretagne