Ensemble Methods

Roman Kern
KDDM2

> **Motivation**: Consider Kaggle, routinely the winners employ ensembles to gain an advantage.
> **Goal**: In this lecture, the main approaches for ensembles will be presented and their main assumptions.

Ensemble Methods
Outline

> Ensembles can be utilised in a supervised, as well as unsupervised setting.
> **Ensembles play an important part in data science**.

Introduction

Motivation & Basics

Introduction
Ensemble Methods Intro

Quick facts

- Basic Idea: Have **multiple models** and a **method to combine** them into a single one.
- Predominately used in classification and regression
- Sometimes called: combined models, meta learning, committee machines, multiple classifier systems
- Ensemble methods do have a long history and used in statistics for more than 200 years

Ensemble Methods Intro

Types of ensembles

- ... different hypothesis
- ... different algorithms
- ... different parts of the data set

> ... or integrate different sources of evidence.
> One might not always aware of working with an ensemble.
> Page https://xgboost.readthedocs.io/en/latest/tutorials/model.html gives a nice example of an ensemble method.
> Goal: Predict if someone likes computer games.
> First tree is built upon the age, and the second one on the daily commute behaviour.
> The prediction is then based on their **combination**.
> In some ensemble the **hypothesis changes** during learning (e.g., boosting, learning to correct the errors of the other ensemble members)

Ensemble Methods Intro

Motivation

- ... as every model has its limitations
- Goal: **combine the strength of all models**
- e.g., improve the accuracy of using an ensemble
- e.g., be more robust in regard to noise

> Do you need **more data**? No (but it certainly helps).

Basic Approaches

- Averaging
- Voting
- Probabilistic methods

Ensemble Methods Intro

Combination of Models

- Need a function to combine the results from the models
- For real values output
  - Linear combination
  - Product rule
- For categorical output, e.g. class labels
  - Majority vote

Ensemble Methods Intro

Linear combination

- Simple form of combining the output of an ensemble
- Given T models, $f_t(y|x)$
- $g(y|x) = \sum_{t=1}^{T} w_t f_t(y|x)$
- Problem of estimating the optimal weights (w_t)
- e.g., simple solution: use the uniform distribution: $w_t = 1/T$

> Assuming a dataset comprising independent variables x, and dependent variables y,
> ... with the goal to predict y, given x (i.e., discriminative classifier)
> The simplest form such a function is a linear combination of the models' output f_t, i.e. a **weighted average**.
> ... and its combination g.

Product rule

- Alternative form of combining the output of an ensemble
- $g(y|x) = \frac{1}{Z} \prod_{t=1}^{T} f_t(y|x)^{w_t}$
- ... where Z is a normalisation factor
- Again, estimating the weights is non-trivial

> Like the other two previous cases, this is just one example.
> The exact way the models are combined is an essential part of the ensemble.

Majority Vote

- Combining the output, if categorical
- The models produce a label as output, e.g. $h_t(x) \in \{+1, -1\}$
- $H(x) = sign(\sum_{t=1}^{T} w_t h_t(x))$
- If the weights are non-uniform, it is a **weighted vote**

> Key insights, which will be later analysed more closely.
> ... we need diversity.
> **Simple explanation**: Just using the very same model multiple times will not improve our results.
> Most of the methods implicitly integrate diversity.

Selection of models

- The models should not be identical, i.e. produce identical results
- ... therefore an ensemble should represent a **degree of diversity**
- Two basic types of achieving this diversity
  - *Implicitly*, e.g. by integrating randomness (bagging)
  - *Explicitly*, e.g. integrate variance into the process (boosting)

Motivation for ensemble methods (1/2)

- Statistical
  - Large number of hypothesis (in relation to training data-set)
  - Not clear, which hypothesis is the best
  - Using an ensemble reduces the risk of picking a bad model

Motivation for ensemble methods (2/2)

- Computational
  - Avoid local minima
  - Partially addressed by heuristics
- Representational
  - A single model/hypothesis might not be able to represent the data

Dietterich, T. G. (2000). Ensemble methods in machine learning. In Multiple classifier systems (pp. 1-15).

Classification

Ensemble Methods for Classification

Underlying question

How much of the ensemble prediction is due to the accuracies of the **individual models** and how much due to **their combination**?

$\rightarrow$ express the ensemble error as two terms:

- Error of individual models
- Impact of interactions, the **diversity**

> It depends on the combination, whether one can separate the two terms.

Regression error for the linear combination

- Squared error of the ensemble regression
- $(g(x) - d)^2 = \frac{1}{T}\sum_{t=1}^{T}(g_t(x) - d)^2 - \frac{1}{T}\sum_{t=1}^{T}(g_t(x) - g(x))^2$
- First term: error of the individual models
- Second term: interactions between the predictions
  - ... the ambiguity, ≥ 0
- $\rightarrow$ Therefore it is preferable to increase the ambiguity (diversity)

> The lhs represents the difference b/w the prediction of the (ensemble) method $g()$ and the ground truth d.
> Actually there is a tradeoff of bias, variance and covariance, known as accuracy-diversity dilemma.

Classification error for the linear combination

- For a simple averaging ensemble (and some assumptions)
- $e_{ave} = e_{add}(\frac{1+\delta(T-1)}{T})$
  - ... where e_{add} is the error of the individual model
  - ... and δ being the correlation between the models

Tumer, K., & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. Connection Science 8(3-4), 385–403.

> The bigger the correlation is b/w the models (i.e., the more similar they are), the higher the error.
> So, independent models should be preferred (as long their individual, respective error is sufficiently small).
> ... later we see that sufficiently small is just better than random guessing.

Basic Approaches

- **Bagging** - combines strong learners → reduce variance
- **Boosting** - combines weak learners → reduce bias
- Many more: mixture of experts, cascades, ...

> Weak learner might be just better than random guessing.

Bootstrap Sampling

- Create a distribution of data-sets from a single dataset
- If used within ensemble methods, it is typically called **bagging**
- Simple approach, but has shown to increase performance

Davison, A. C., & Hinkley, D. (2006). Bootstrap methods and their applications (8th ed.). Cambridge: Cambridge Series in Statistical and Probabilistic Mathematics

> Sample from the dataset will create subsets that should be independent.
> Of course the dataset needs to be sufficiently large.

Bagging

- Each member of the ensemble is generated by a different dataset
- Good for **unstable models**
  - ... where small differences in the input dataset yield big differences in output
  - Also known as *high variance* models

Note: Bagging is an abbreviation for bootstrap aggregating
Breiman, L. (1998). Arcing classifiers. Annals of Statistics, 26(3), 801–845.

> → not so good for simple models.

Bagging Algorithm (train)

1. Input: Ensemble size T, training set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$
2. For each model M_t
   a. For n' times, where $n' \leq n$
      i. Sampling (random) from D with replacement
   b. Train model M_t with subset

> Subset may contain duplications, i.e. if $n' = n$

Bagging Algorithm (classify)

- For classification typically majority vote
- For regression typically linear combination

Boosting

- Family of ensemble learners
- Boost weak learners to a strong learner
- **Adaboost** is the most prominent one
- Weak learners need to be better than random guessing

Adaboost

- Basic idea: Weight the individual instances of the data-set
- Iteratively learn models and record their errors
- Distribute the effort of the next round on the mis-classified examples

Adaboost (train)

1. Input: Ensemble size T, training set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$
2. Define a uniform distribution W_t over elements of D
3. For each model M_i
   a. Train model M_i using distribution W_t
   b. Calculate the error of model ϵ_t and weight $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
   c. ... if $\epsilon_t > 0.5$ break (and discard model)
   d. ... else update the distribution W_t according to ϵ_t

Adaboost (classify)

- Linear combination, $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Stacked generalisation

- Idea: Have the output of a layer of classifiers as input to another layer
- For 2 layers:
  1. Split the training data-set into two parts
  2. Learn the first layer using the first part
  3. Classify the second part and
  4. ... take the decision as input for the second part

Wolpert, D. H. (1992). Stacked generalization. Neural Networks 5(2), 241–259

Mixture of Experts

- Idea: some models should specialise on parts of the input space
- Ingredients
  - Base models (e.g. specialised models - so called experts)
  - Component to estimate probabilities, often called a gating network
- The gating networks learns to select the appropriate expert for parts of the input space

Mixture of Experts - Example #1

- Ensemble of base learners being combined using weighted linear combination
- The weight is found via a neural network
  - The neural network is learnt via the same input data-set

Mixture of Experts - Example #2

- Mixture of expert models are called mixture models
- e.g. the Expectations-maximisation algorithm

Cascade of classifiers

- Setting
  - Have a sequence of models, each with high hitrate ($\geq h$) and low false alarm rate ($< f$)
  - ... with increasing complexity
  - In the data-set the negative examples are more common
- The cascade is learnt via boosting

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001.

Cascade of classifiers

- For example:
  - For $h = 0.99$ and $f = 0.3$ and a cascade of size 10
  - ... one gets the hitrate of about 0.9 and a false alarm rate of about 0.000006

> One example for a cascade of classifiers is the face detection in cameras.
> Here a series of identification algorithms work:
> First one with a high false positive rate, but very quick.
> Succeedingly the candidates will be filtered out by increasingly lower false positive rates, at the expense of runtime.
> i.e., the last one is the "slowest" but most precise.

Decision Stump

- **Decision stumps** are a popular choice for (some) ensemble learning
- ... as they are fast
- ... as they are less prone to overfitting
- A decision stump is a decision tree that only uses a single feature (attribute)

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11, 63–91.

Random Subset Method

- Basic idea: Instead of taking a subset of the data-set, use a subset of the feature set
- ... will work best, if there are many features
- ... and will not work as well if most of the features are just noise

> Also interesting, if many features are correlated with each other.
> A phenomenon, also known as multi-collinearity, where, e.g., simple linear regression struggles with.
> Often a result of confounders, which lead to partial correlation between (otherwise independent) variables.

Random Forest

Random Forest

- Combines two randomization strategies
  - Select random subset of the dataset to learn decision tree (bagging), e.g., select $n = 100$ random trees
  - Select random subset of features, e.g., select $\sqrt{m}$ features
- Random forests are used to estimate the importance of features (by comparing the error using a feature vs. not using a feature)

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

> Typically good performance, therefore often the goto-method for data science.
> Not a big problem for multi-collinearity, but the feature importance may suffer in such cases.

Boosted Trees

Boosted Trees

- Idea: Sequence of trees, which results are added
  - Could be seen as increasingly correcting the errors of the predecessor trees
- Gradient boosting
  - Take the gradient of an (differentiable) objective function into account, while building the trees

> The objective function is typically a loss (e.g., RMSE), plus a regularisation term.
> Each new tree learns on the residuals of the previous ensemble.
> The residuals can be seen as (negative) gradients (delta b/w true and predicted).
> Gradient boosting is flexible: change tree types, loss functions, even integrate bagging (Stochastic Gradient Boosting), ...
> Popular choices for implementation of the idea are: LightGBM, XGBoost.

Multiclass Classification

- Basic idea: split a multi-class problem into a set binary classification problems
- e.g., Error correcting output codes

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In International conference on machine learning.

> Some classifiers can deal with multiple classes (e.g., k-NN), which others don't (e.g., logistic regression).
> There are multiple ways to achieve multi-class classification with just binary classifiers.
> e.g., one-vs-one, one-vs-rest.

Ensemble classification for multi-class problems

- Have different base classifiers for different parts of the feature set
  - Train all base classifiers using the training data-set
  - Record their performance with cross-evaluation for each class
- ... have two thresholds, $min_{precision}$ and min_{recall}
  - If the precision for a certain class and model is $\geq min_{precision} \rightarrow$ allowed to vote
  - If the recall for a certain class and model is $\geq min_{recall} \rightarrow$ allowed to vote against (veto)

Ensemble classification for multi-class problems

- In the classification use a weighted vote
  - where veto is a negative vote
  - ... and the weight is according to the respective measure (precision or recall)

Kern, R., Seifert, C., Zechner, M., & Granitzer, M. (2011, September). Vote/Veto Meta-Classifier for Authorship Identification Notebook for PAN at CLEF 2011.

- **Active learning** is a form of semi-supervised learning
  - The basic idea is to give the human instances to label
  - ... which carry the most information (to update the model)
- **Query by Committee**
  - ... use an ensemble, i.e. the disagreement of multiple classifiers to pick instances

Clustering

Other Tasks and Conclusions

Cluster Ensembles

- Idea: Have **multiple clustering algorithms** group a data-set
- ... combine all results into a single clustering results
- Motivation: More reliable result than individual cluster solutions

Cluster Ensembles

Consensus Clustering

- Have a set of clusterings: $\{C_1, ..., C_m\}$
- Find an overall clustering solution C
- Minimise the disagreement using a metric: $D(C) = \sum_{C_i} d(C, C_i)$
- Also known as clustering aggregation

Cluster Ensembles

Mirkin Metric

- The metric reflects the numbers of pairs of instances ...
- ... being together in the overall clustering, but separate in C_i
- ... and vice versa

Other Ensembles

- Ensemble methods are not limited to machine learning tasks alone
- For example, in the field of recommender systems they are known as **hybrid recommender system**
  - e.g. combine a content based recommender with a collaborative filtering one

Ensembles in Data Science

Pros

- Typically good results, especially if dataset is not well understood
- Cope well with noisy datasets
- Gives insights
  - … what features are important
  - … what hypothesis might be the most suitable

Ensembles in Data Science

Cons

- Computationally complex
- Motivate a try-run-repeat approach

The End

Thank you for your attention!