DATASCI 207

Nedelina Teneva, PhD
nteneva@berkeley.edu

School of Information, UC Berkeley

Announcements

- Finalize datasets by next week: enter dataset info in the Logistics Sheet
- No class on July 4th

HW Recap

- HW3
  - Q3: Data normalization: not **using the train mean/std dev to normalize test data**
  - Q4: Report **validation** loss for learning rates
  - Q5: RMSE is wrong due to incorrect normalization
- HW4
  - Q3: forgot to run the loss on train/test

Generative vs Discriminative Models

On Discriminative vs. Generative Classifiers: A comparison of Logistic Regression and Naive Bayes, Ng & Jordan

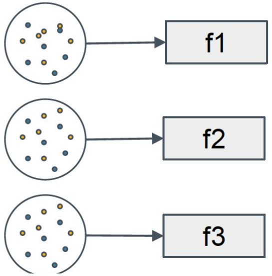
- **Generative classifiers**
  - learn a model of the joint probability p(x,y) and pick the most likely y
  - Make predictions using Bayes rule to compute the posterior p(y|x) and from it choose the most likely y.
  - Examples: Naive Bayes (earlier in the semester), Gaussian Mixture Models (later on)
- **Discriminative classifiers**:
  - Model the posterior p(y|x) directly and learn a map from the inputs (x) to the labels (y)
  - Examples: logistic regression, trees, Support Vector Machines (SVM)

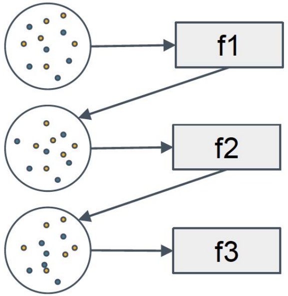
Generative models need to compute p(x|y) for Bayes theorem to work – this is an extra step, which discriminative classifiers avoid by modeling p(y|x) directly (Bayesian <> Frequentists debate)

Ensembles

Learning with Ensembles

- The idea is to combine different classifiers into a super-classifier whose performance is much better than using each individual classifier alone.
- Assume that you collect predictions from 10 expert classifiers (e.g., KNN, NB, ...). ensemble methods will allow us to combine their individual predictions to come up with a final prediction that is more accurate and robust.
- 3 most commonly used ensembles: majority vote, bagging, and boosting.
    - **Bagging**: train models in parallel via bootstrap sampling
    - **Boosting**: train additive models in series where each predicts the residual from the previous ones





Majority Vote

The majority vote principle

- selects the class label that has been predicted by the majority of classifiers (i.e., received more than 50 percent of votes).
- the term majority vote is used for a binary class setting.
- plurality vote is used for a multiclass setting.

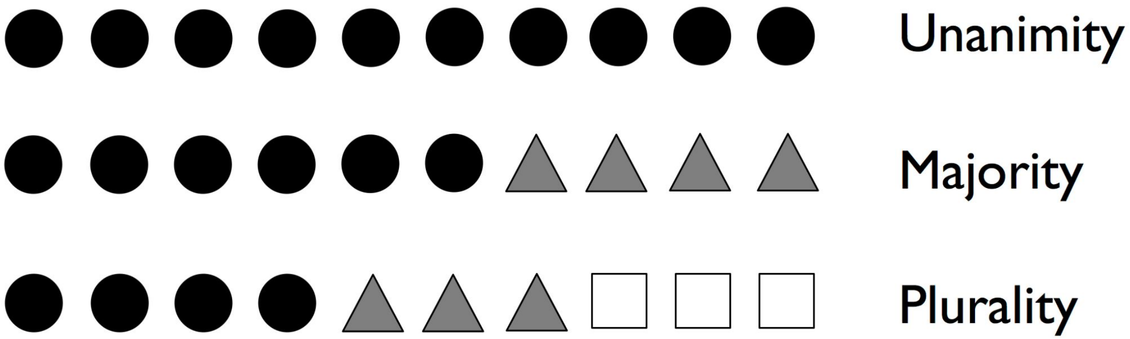


Fig: Raschka's book

Bagging

The bagging principle

- is closely related to the majority vote technique.
- the difference is that instead of using the same training data to fit the different classifiers in the ensemble, **we draw bootstrap (random) samples with replacement** from the initial training data.

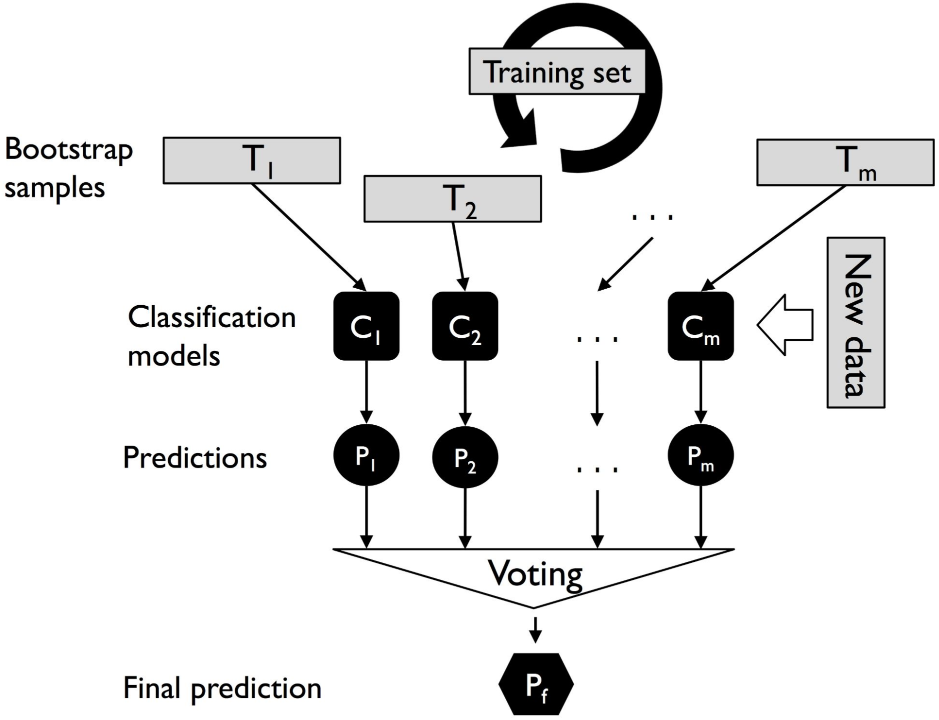


Fig: Raschka's book

Adaboost

The adaptive boosting principle - AdaBoost classifiers

- the ensemble consists of very simple base classifiers (**weak** learners; think decision tree stump).
- focuses on training examples that are hard to classify, i.e. let the weak learns learn from their mistakes (misclassified training examples) to improve the performance of the ensemble.
- in contrast to bagging, the boosting algorithm uses random subsets of training examples drawn from the training dataset **without** replacement.

example decision stump: tree of height 1

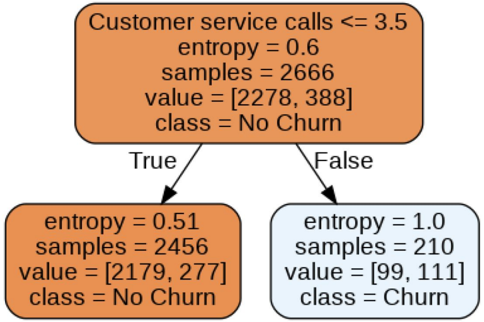


Fig: from Async

Ensemble Exercises

https://github.com/MIDS-W207/nteneva/tree/main/live_sessions_current/week7

KNN

KNN

- How does it work?

KNN

- How does it work?

Algorithm Pseudocode:

- Step 1: choose the number of k (neighbors) and a distance metric.
- Step 2: Find the k-nearest neighbors of the record we want to classify.
- Step 3: Assign the class label by majority vote.

KNN

- It's a very **different and simple** algorithm compared to the ones we will see in this course.
- Lazy learner: it doesn't learn any discriminative function from the training data, but memorizes the data instead.
- Advantages:
  - adapts easily to new data.
- Disadvantages:
  - we cannot discard training examples because no training is involved.
  - storage space can become a challenge with large datasets.
  - computationally complexity grows linearly with the size of the data.

Implement KNN

https://github.com/MIDS-W207/nteneva/blob/main/live_sessions_current/week7/exercise_knn_classifier.ipynb