Lecture 26: Ensemble Methods

Machine Learning

Today

- Ensemble Methods.
  - Classifier Fusion
    - “late fusion” vs. “early fusion”
  - Cross-validation as Ensemble Evaluation
  - Boosting
    - AdaBoost
  - Random Forests
- Recap of the course.

Ensemble Methods

- People don’t make decisions by considering all evidence equally.

- Consider a “fake gun”.
  - By shape it makes sense to call it a **WEAPON**
  - By material and function it makes sense to call it a **TOY**

- People can hold both ideas simultaneously, and combine multiple perspectives, even trusting one more than another.

Ensemble Methods

- Ensemble Methods are based around the hypothesis that an aggregated decision from multiple experts can be superior to a decision from a single system.

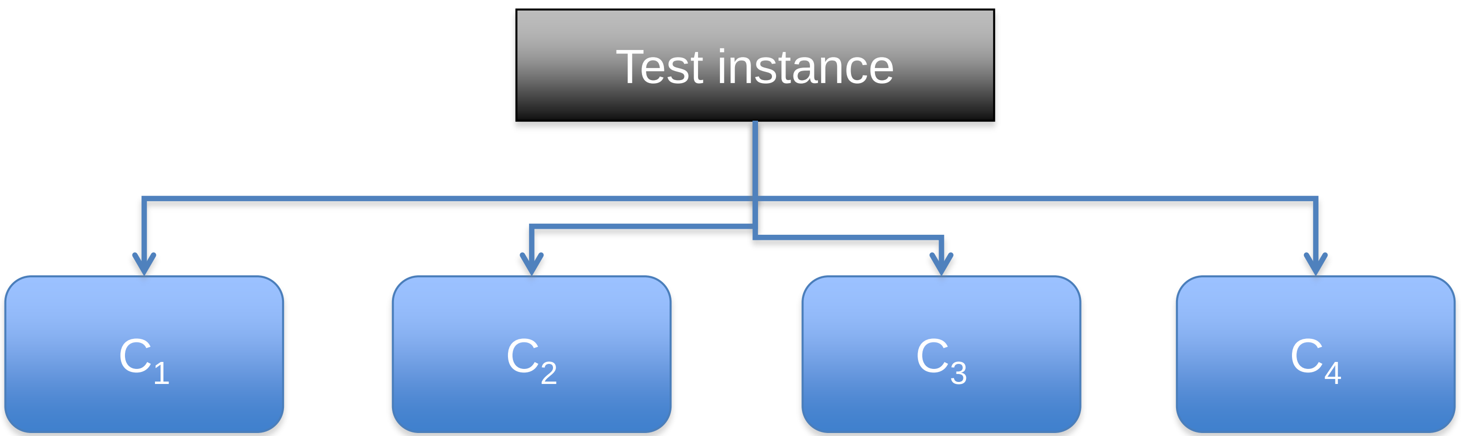
Classifier Fusion

- Train k classifiers to map x to t where t in C

- These classifiers should be trained on either
  - Distinct features, or
  - Distinct data points

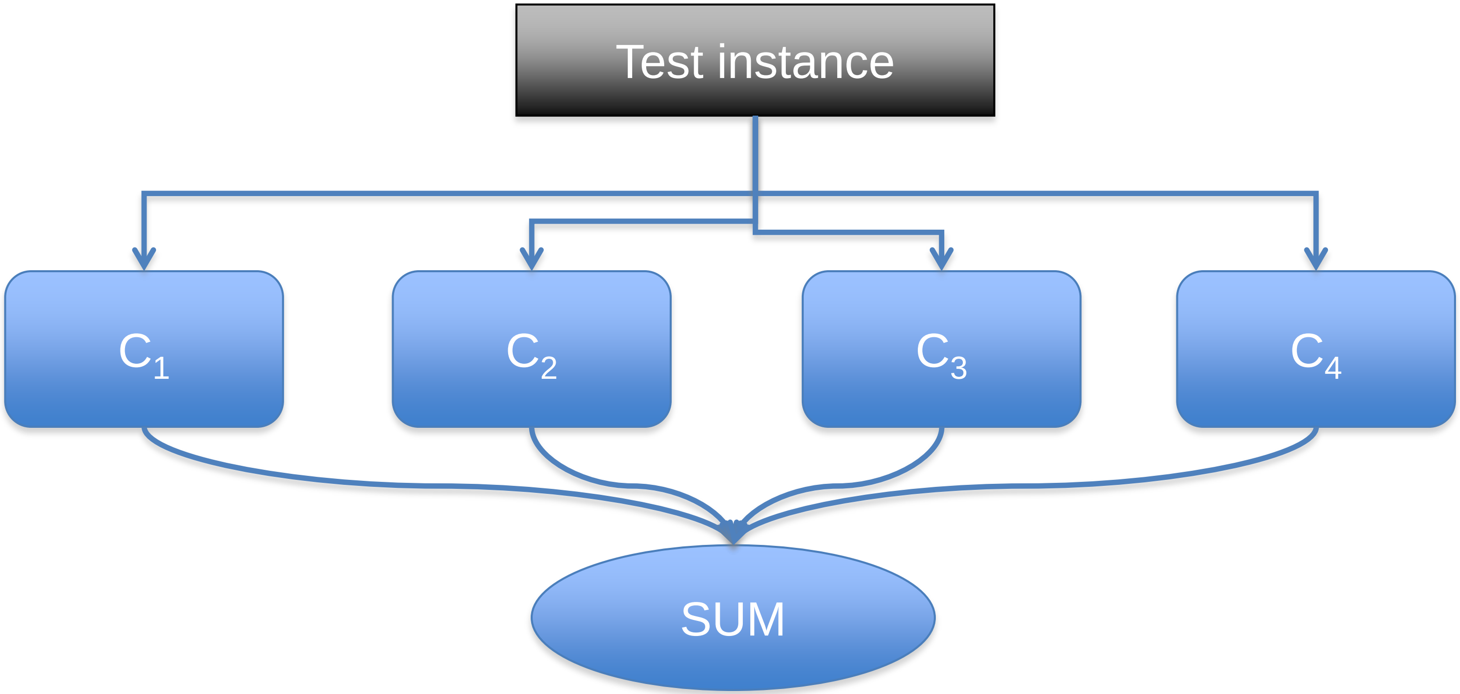
Classifier Fusion

- How do you construct an answer from k predictions?



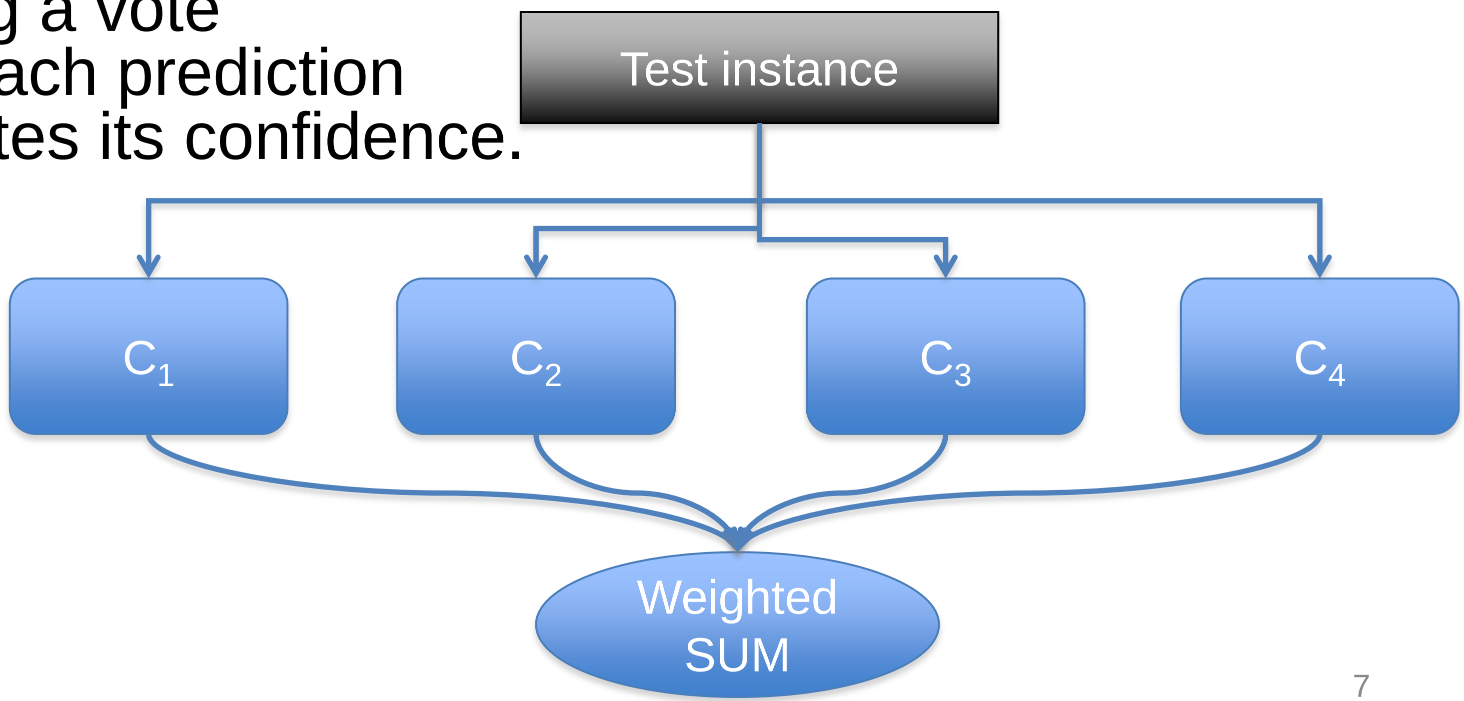
Majority Voting

- Each Classifier generates a prediction and confidence score.
- Chose the prediction that receives the most "votes" predictions from the ensemble



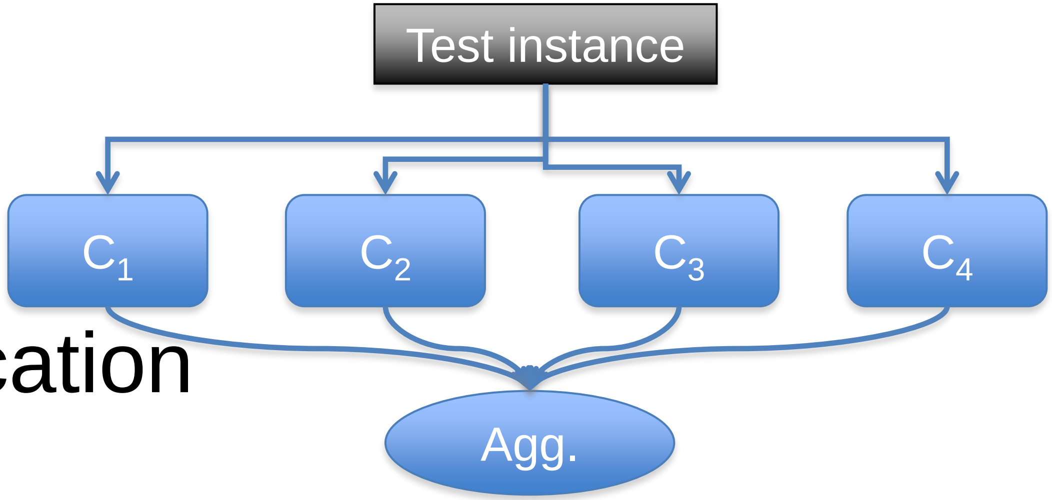
Weighted Majority Voting

- Most classifiers can be interpreted as delivering a distribution over predictions.
- Rather than sum the number of votes, generate an average distribution from the sum.
- This is the same as taking a vote where each prediction contributes its confidence.

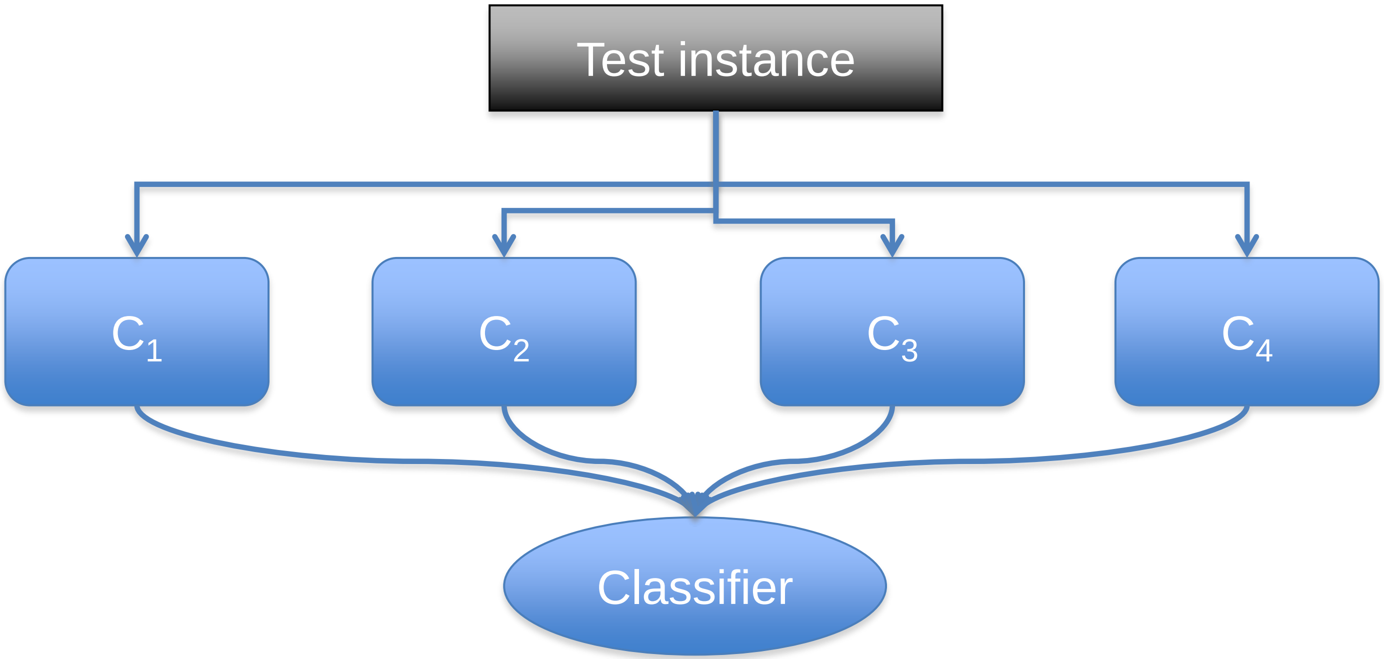


Sum, Min, Max

- Majority Voting can be viewed as summing the scores from each ensemble member.
- Other aggregation function can be used including:
  - – maximum
  - – minimum.
- What is the implication of these?

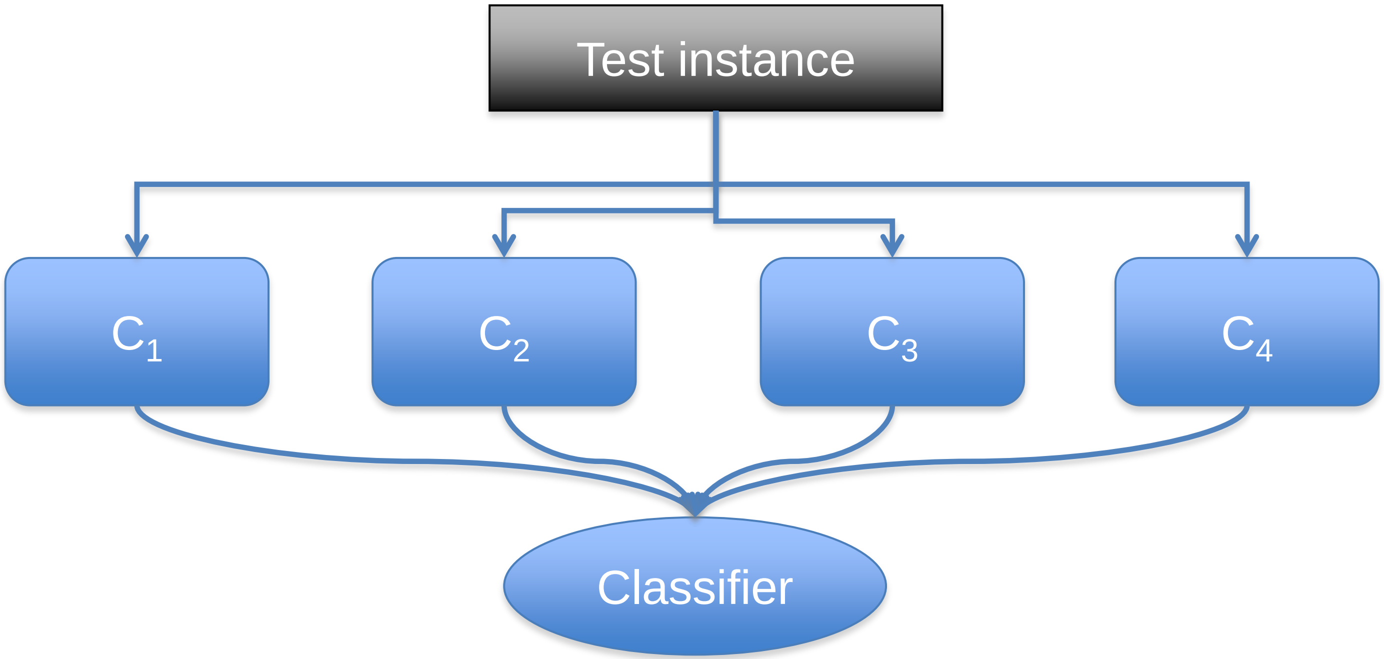


Second Tier classifier



- Classifier predictions are used as input features for a second classifier.
- How should the second tier classifier be trained?

Second Tier classifier



- The second tier classifier must be trained on the training data.
- Want reliable predictions from C_k.
- Use 50% data to train C_k and the other 50% to train the fusion classifier.

Classifier Fusion

- Each of these approaches are called “late fusion”.
  - The combination of features or data points happens after the initial classifier training.
- “Early fusion” is when the initial training data or feature set is augmented.

Classifier Fusion

- Advantages
  - Experts to be trained separately on specialized data
  - Can be trained quicker, due to smaller data sets and feature space dimensionality.
- Disadvantages
  - Interactions across feature sets may be missed
  - Explanation of how and why it works can be limited.

Cross-Validation

- Cross validation trains k classifiers, one for each fold.
- The evaluation measure is constructed from an average of the k evaluations

- No ensemble is used in the classification scheme.
- The ensemble is used only for evaluation

AdaBoost

- Adaptive Boosting is an approach that constructs an ensemble of simple “weak” classifiers.
- Each classifier is trained on a single feature.
  - Often single split decision trees.
- The task of the classification training is to identify an ensemble of classifiers and their weights for combination

http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf

AdaBoost classification

$$C(x) = \alpha_1 C_1(x) + \alpha_2 C_2(x) + \ldots + \alpha_k C_k(x)$$

- AdaBoost generates a prediction from a weighted sum of predictions of each classifier.
- The AdaBoost algorithm determines the weights.
- Similar to systems that use a second tier classifier to learn a combination function.

AdaBoost Algorithm

- Repeat
  - Identify the best unused classifier Ci.
  - Assign it a weight based on its performance
  - Update the weights of each **data point** based on whether or not it is classified correctly
- Until performance converges or all classifiers are included.

Identify the best classifier

- Evaluate the performance of each unused classifier.
- Calculate weighted accuracy using the current data point weights.

$$W_e = \sum_{y_i \neq k_m(x_i)} w_i^{(m)}$$

Assign the weight for the current classifier

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-e_m}{e_m}\right)$$

$$e_m = \frac{W_m}{W}$$

- The larger the reduction in error, the larger the classifier weight

Update the data point weights for the next iteration

- If i is a miss:

$$w_i^{(m+1)} = w_i^{(m)} e^{-\alpha_m} = w_i^{(m)} \sqrt{\frac{e_m}{1 - e_m}}$$

- If i is a hit:

$$w_i^{(m+1)} = w_i^{(m)} e^{\alpha_m} = w_i^{(m)} \sqrt{\frac{1 - e_m}{e_m}}$$

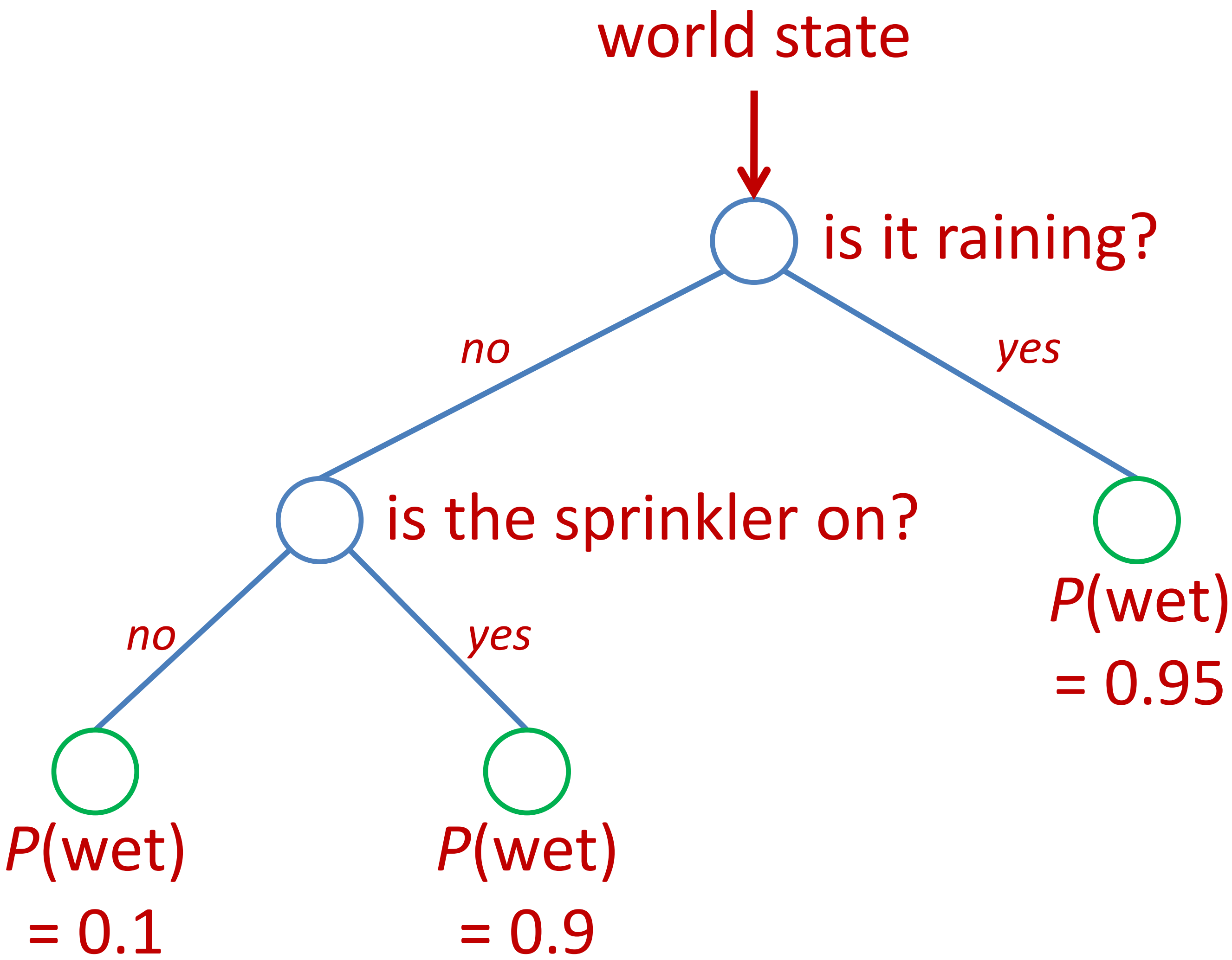
AdaBoost Algorithm

- Repeat
  - Identify the best unused classifier Ci.
  - Assign it a weight based on its performance
  - Update the weights of each **data point** based on whether or not it is classified correctly
- Until performance converges or all classifiers are included.

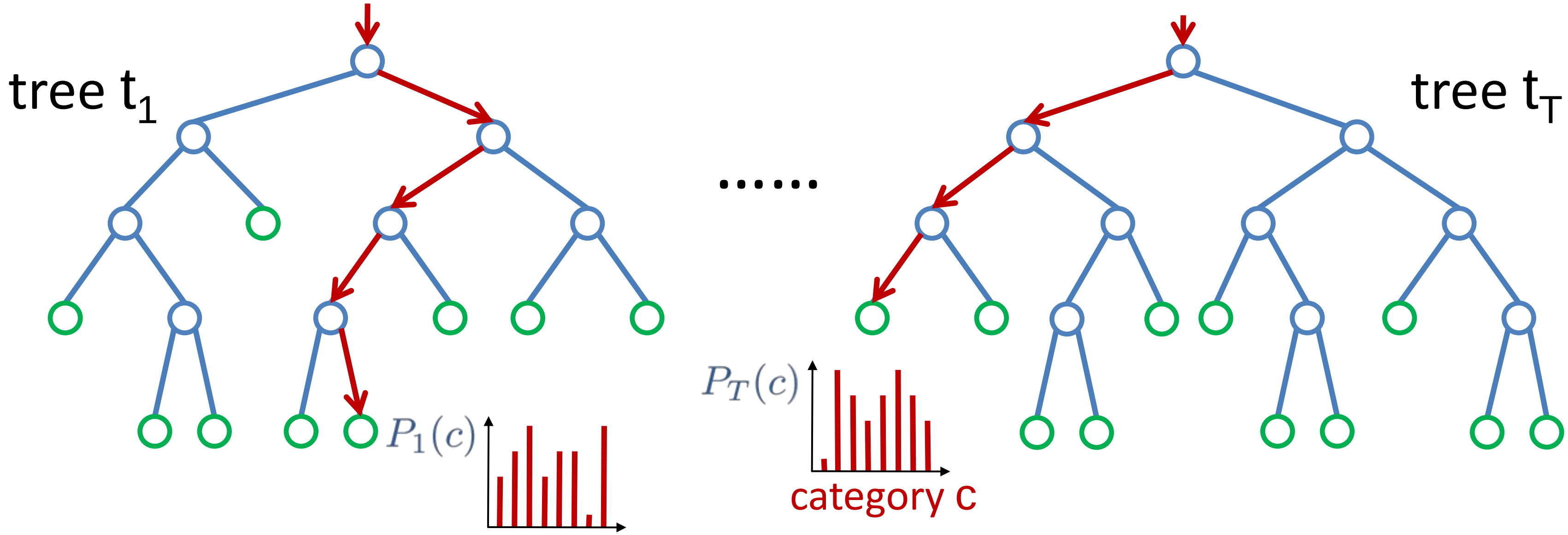
Random Forests

- Random Forests are similar to AdaBoost decision trees.
- An ensemble of classifiers is trained each on a different random subset of features.
  - Random subspace projection

Decision Tree



Construct a forest of trees



$$P(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{v})$$

Learning the Forest

- Divide training data into K subsets.
  - Improved Generalization
  - Reduced Memory requirements
- Train a unique decision tree on each K set
- Simple multi threading

- These divisions can also operate across features

Course Recap

- Statistical Estimation
  - Bayes Rule
- Maximum Likelihood Estimation
  - MAP
- Evaluation
  - **NEVER TEST ON TRAINING DATA**
- Classifiers
  - Linear Regression
    - Regularization
  - Logistic Regression
  - Neural Networks
  - Support Vector Machines
- Clustering
  - K-means
  - GMM
- Expectation Maximization
- Graphical Models
  - HMMs
- Sampling

Next Time

- Your Presentations