Machine Learning course - Chapter 9

Classification

(Second part)

04-04-16
Laura Hangard

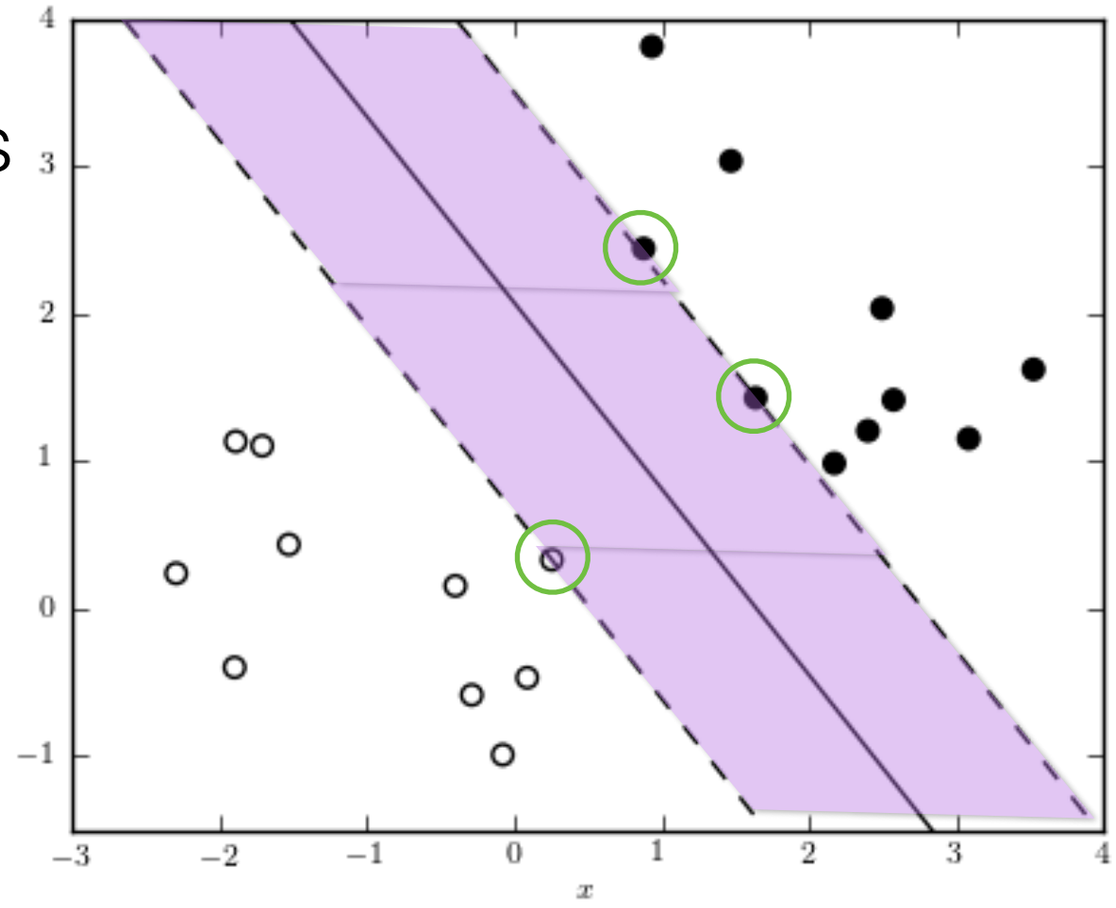
9.6. Support Vector Machines (SVM)

Another way of choosing a linear decision boundary.

9.6. Support Vector Machines (SVM)

"Margin" = hyperplane that maximizes the distance of the closest point from either class.

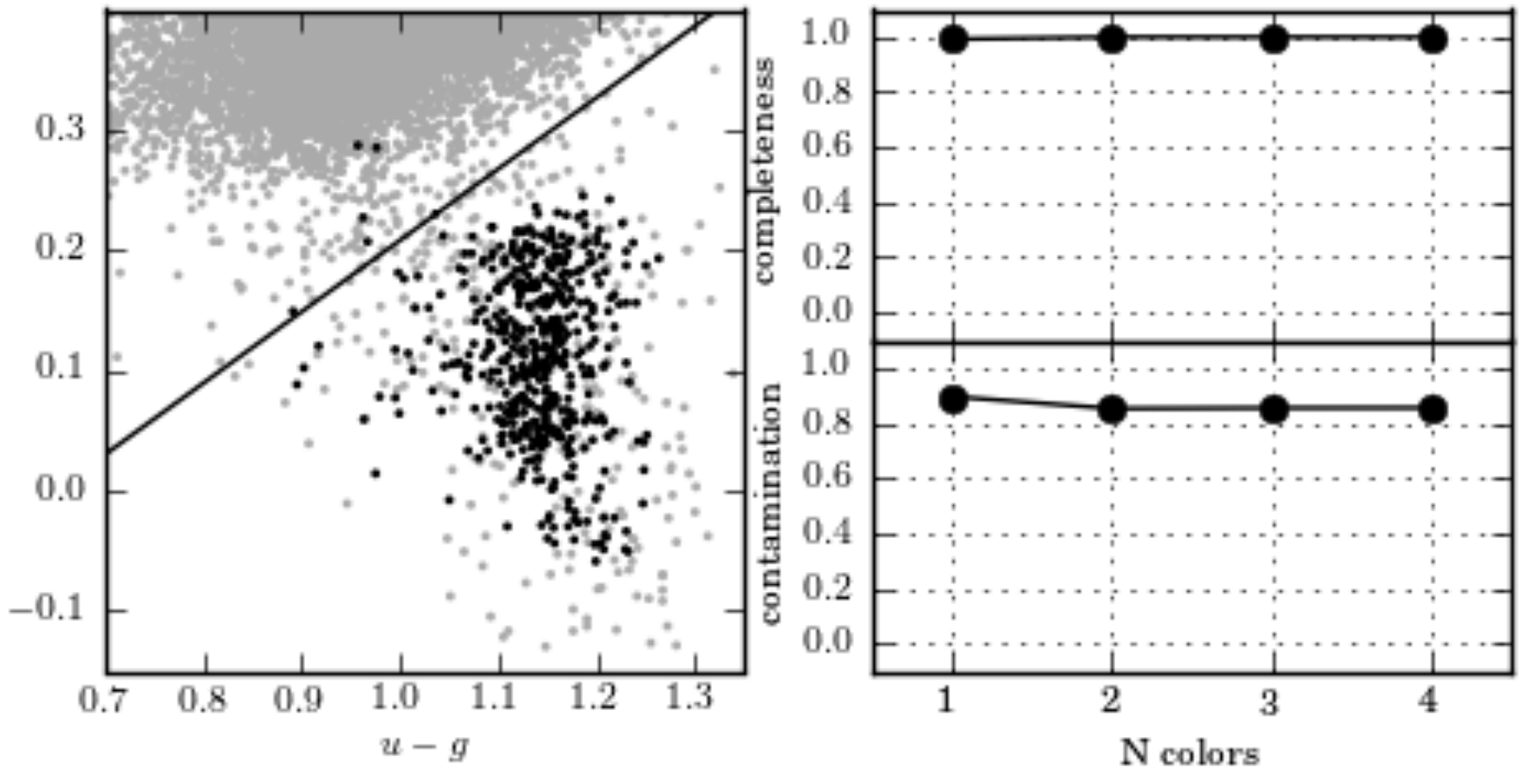
"Support vectors" = points on the margin.



The idea is to maximize the size of the margin.

9.6. Support Vector Machines (SVM)

SVM decision boundary for the RR Lyrae dataset :



Outliers don't affect result

→ *powerful tool for discriminative classification.*

Higher completeness than previous methods : unbalance in number of objects in each sample do not affect the choice of best boundary between classes.

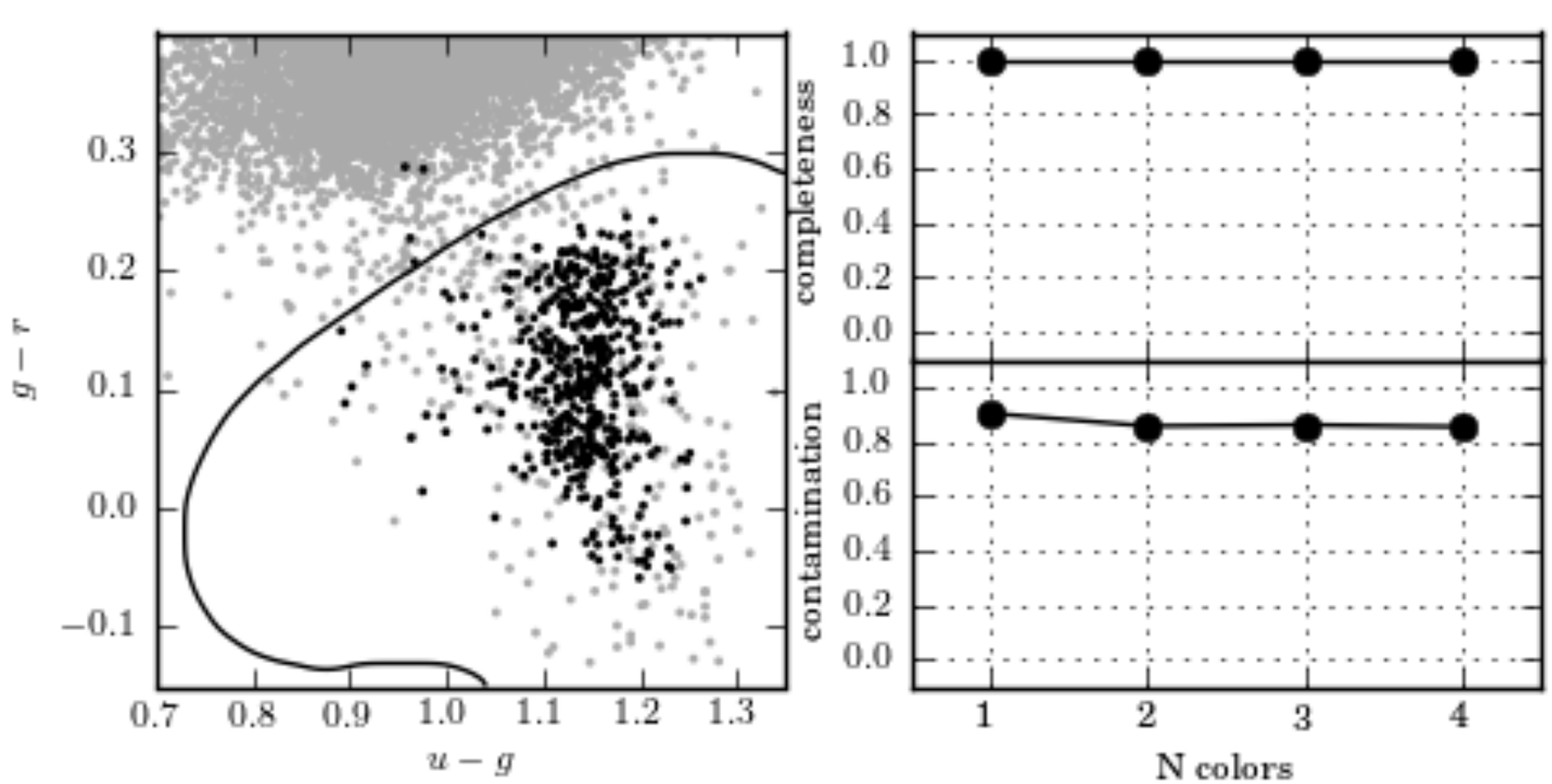
But **high level of contamination**.

9.6. Support Vector Machines (SVM)

Major limitation : only consider **linear** decision boundaries.

➔ **Kernelization** good way to make SVM non linear.

Kernel SVM applied to RR Lyrae data :



(Here not better because contamination not driven by non linear effects.)

9.7. Decision Trees

Powerful methodology for classification.

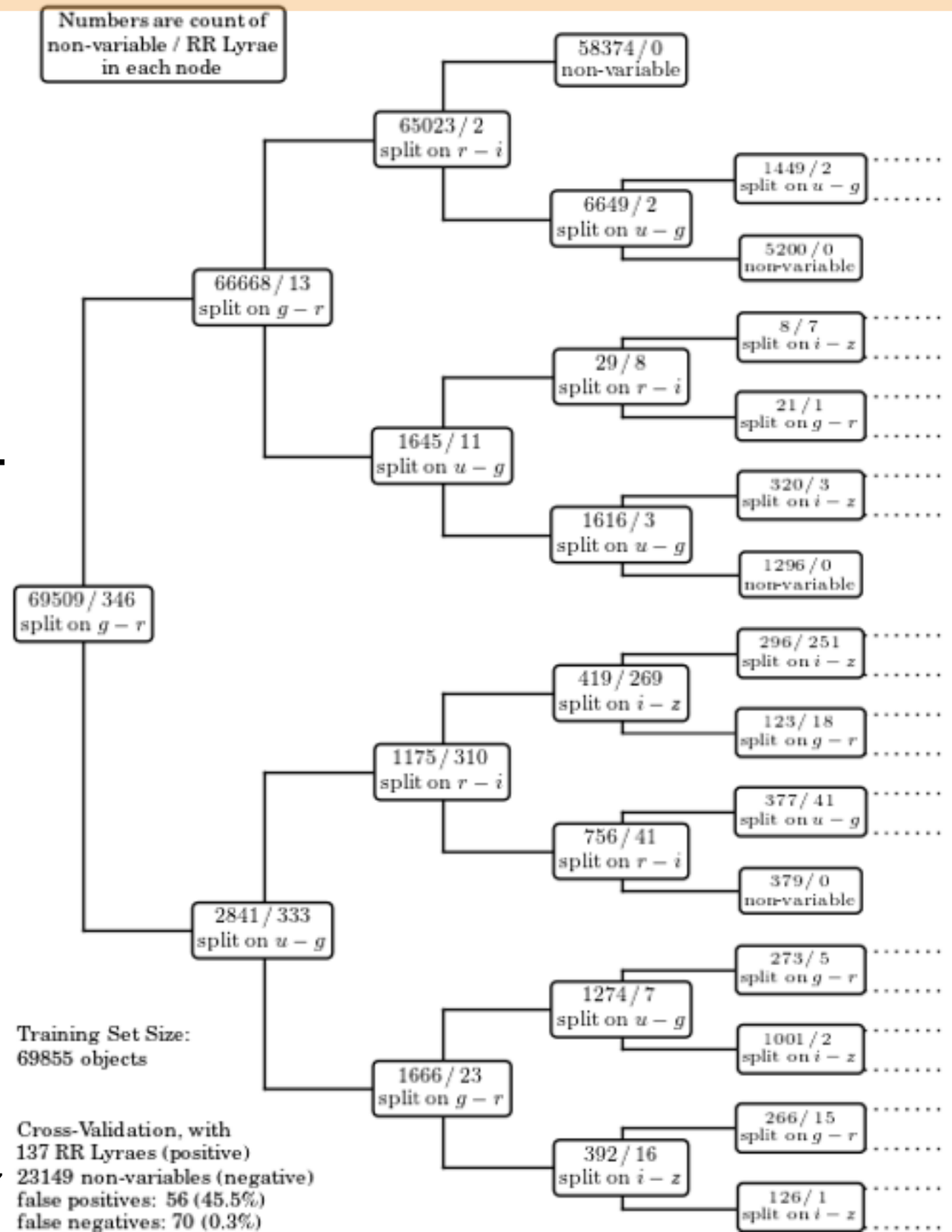
9.7. Decision Trees

- First node : entire dataset

- **Split into 2 subsets at each node** (based on predefined decision boundary)
- **Boundaries are axis aligned** (ie. data split along features)
- **Repeat** until predefined stopping criteria

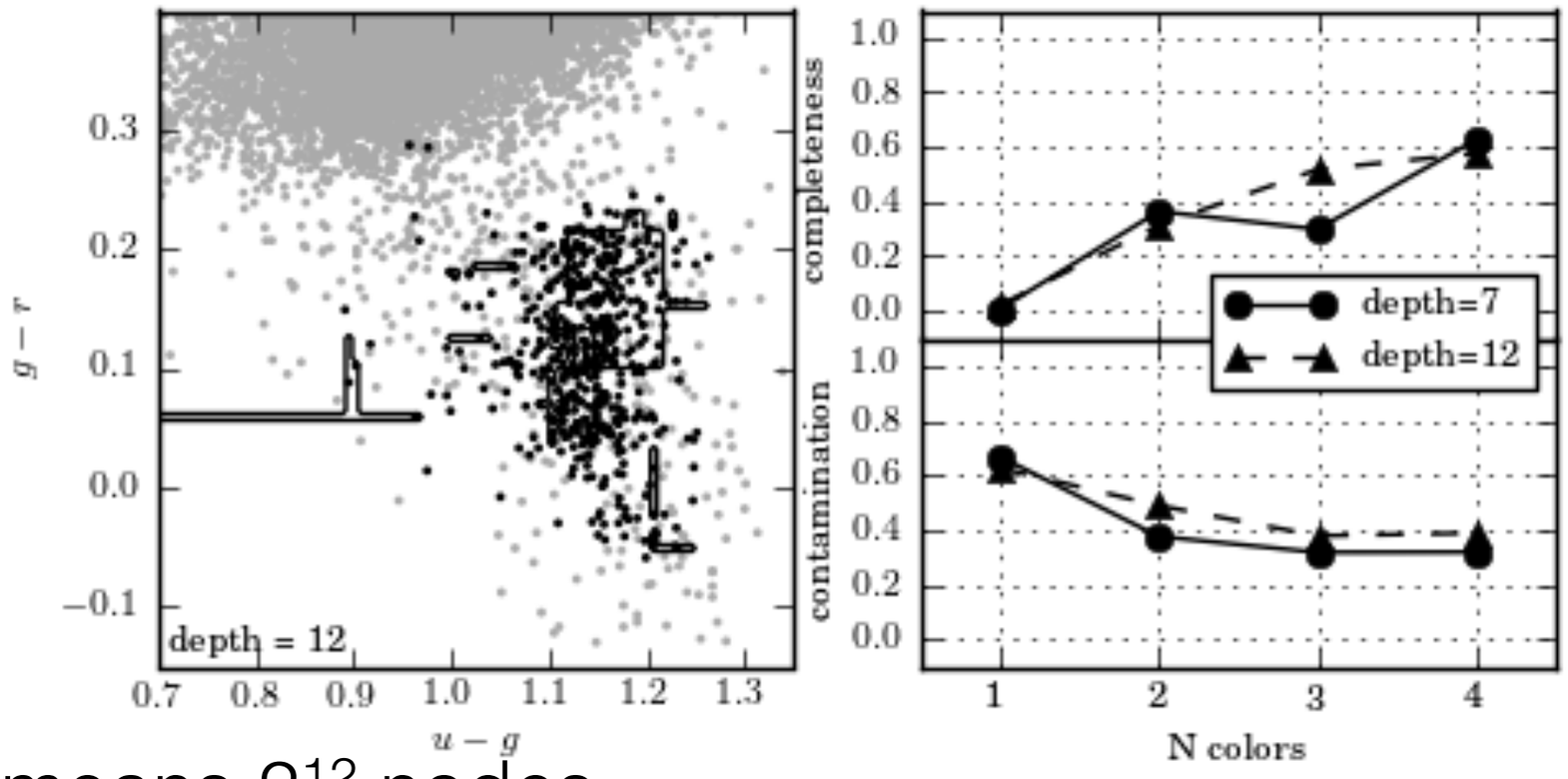
- End of tree : “leaf nodes” with fraction of each class.

Relative fraction of points classified as one class or the other defines the class associated with each leaf node.



9.7. Decision Trees

Depth of tree has an effect on precision and accuracy.
Decision Tree applied to RR Lyrae dataset :



Depth = 12 means 2^{12} nodes.
Risk of overfitting the data if you divide into regions that are too small.
Use fewer nodes for a better classifier.
Resume :
Series of binary questions asked to the data that become more and more refined.

9.7. Decision Trees

Defining the Split Criteria:

Choice of feature and value on which to split the data at each node.

- Define "**information gain**" as reduction in entropy due to partitioning of the data.

- Consider each feature one at a time, and the one providing **largest** information gain is split.

(+ Gini coefficient based on proba of misclassification, see book)

9.7. Decision Trees

Building the tree:

Where to stop splitting?

- a node contains only one class of object;
- a split does not improve information gain;
- nb of points/node reaches a predefined value.

Increasing depth => decreasing error on the training set.

But careful ! if the depth of the tree increases too much, you won't fit correlations in the data but noise.

→ Use **cross-validation** technique to optimize depth of tree.

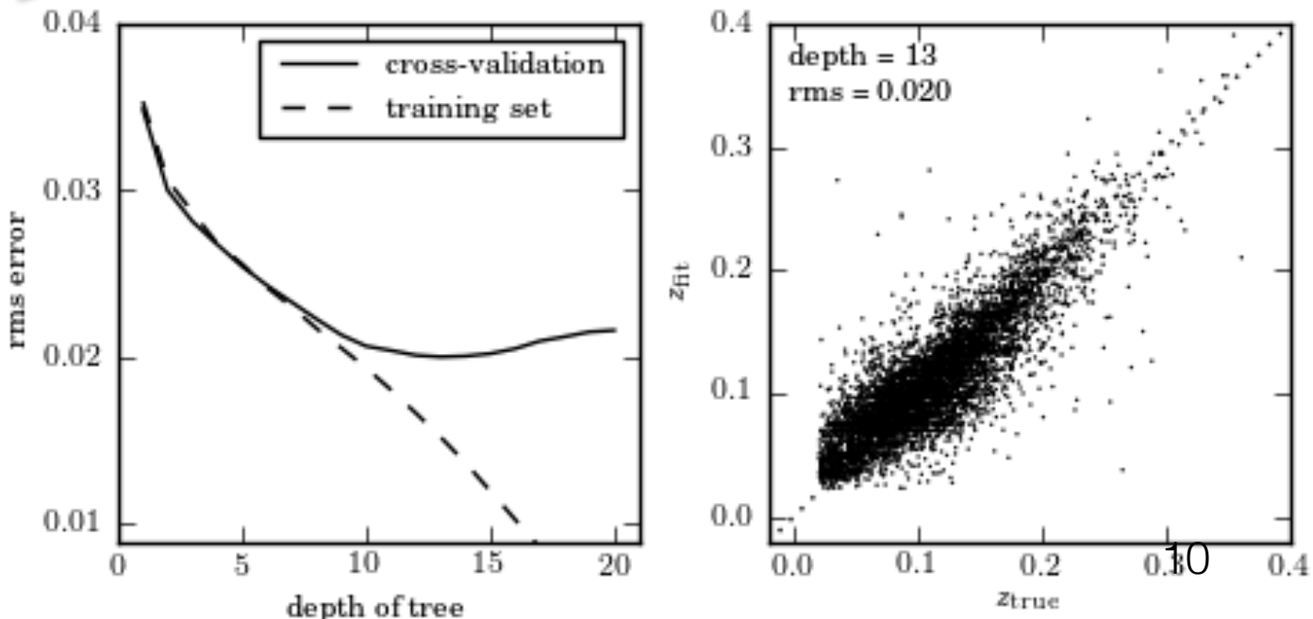


Figure 9.14
Photometric redshift estimation using decision-tree regression. The training set consists of u, g , r, i, z magnitudes of 60,000 galaxies from the SDSS spectroscopic sample. Cross-validation is performed on an additional 6000 galaxies. The left panel shows training error and cross-validation error as a function of the maximum depth of the tree. For a number of nodes N > 13, overfitting is evident.

9.7. Decision Trees

Bagging and Random Forests (RF): “ensemble learning”

Bagging : average the predictive results of a series of equally sized bootstrap samples from training dataset.

Random forests : you generate a set of decision trees from the bootstrap samples.
Final classification from the RF is based on average of the classifications of all decision trees.

- Reduce risk of overfitting (averaging)
- Reduce problem of correlations between decision boundaries.

9.7. Decision Trees

Need to define :

n = number of trees.
Will be chosen by increasing it until cross-validation error reaches a plateau

m = number of features considered for splitting at each node of the trees.
(keep m small to have a simpler model and avoid overfitting)
Typically m ~ sqrt(features in the sample)

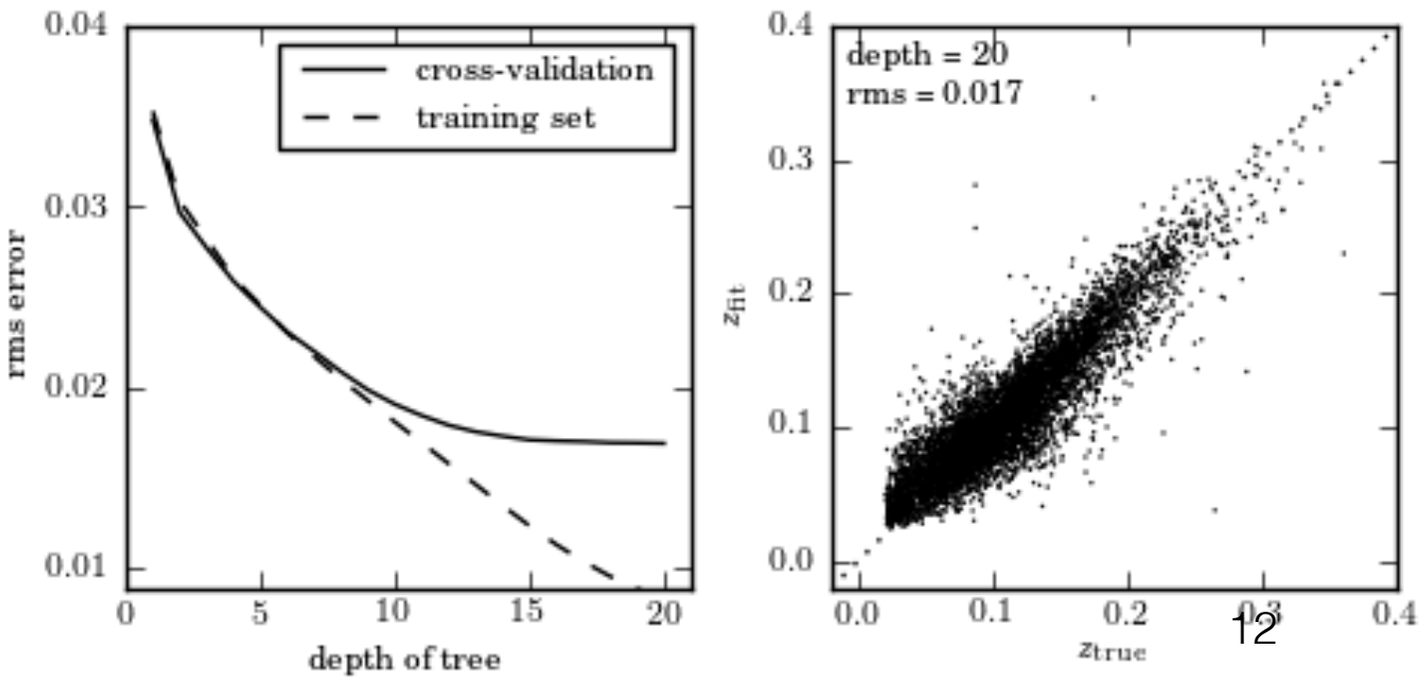


Figure 9.15
Photometric redshift estimation using random forest regression, with ten random trees. Comparison to figure 9.14 shows that random forests correct for the overfitting evident in very deep decision trees. Here the optimal depth is 20 or above, and a much better cross-validation error is achieved.

9.7. Decision Trees

Boosting Classification :

Ensemble approach but this time using a **chain of classifiers**.

→ **Reweight the data based on the performance of the previous classifier.**

Limitation : computation time for large datasets (chain of classifiers vs. in parallel for RF).

9.8. Evaluating Classifiers

Which one to chose?

9.8. Evaluating classifiers : ROC Curves

Chose your classifier

You want to achieve:

- High completeness?
- Minimum contamination?

Visualizations:

- ROC curves
- Efficiency vs. completeness