UNIT 9 →

3/18

kNN, SVM Random Forest, Ensemble

- **instance based learning methods**: store the training examples without doing any calculations during training practice, and classification/prediction is delayed until new examples are given
- **k-nearest neighbor**:
  - ↳ training process: read in all training examples
  - ↳ classification process: given a test example, compare the similarity between the test example and all training examples, choose the majority-voted category label in the k nearest training examples
  - ↳ choosing the value of k: if k is too small, sensitive to noise points, if k is too large, neighborhood may include points from other classes
- advantages of kNN
  - -no assumptions made. (remember the independence assumption in naive bayes), works well when the decision function to be learned is very complex,
  - -decision boundary has no pre-defined shape
- disadvantages
  - -sensitive to noisy training data, all attributes participate ~~of those that~~ in classification, high computational cost
- **linear decision boundaries**
  - -find a linear hyperplane that can separate the data, find one that maximizes the margin
- **support vectors**: training examples located on the margins
- **non support vectors**: training examples which are not support vectors do not participate in prediction
- **model complexity**: the number of support vectors is an indicator of the complexity of the trained SVM model, farther = better
- **regularization**: use manual ~~train~~ tuning or gradient desient search to find the best C
- **SVM strength**: high tolerance to noise, flexibility, probabilistic prediction result, scalability, successful with real-world data

- SVM weaknesses: require a number of parameters for each kernel type, easy to interpret for linear kernel, but not easy to interpret model generated by nonlinear kernels
- Ensemble methods: construct a set of classifiers from the training data, predict class label of previously unseen records by aggregating predictions made by multiple classifiers
- bagging: used when the goal is to reduce the variance of a classifier
- boosting: used to create a collection of predictors