ECE M146 Introduction to Machine Learning

Prof. Lara Dolecek

ECE Department, UCLA

Today's Lecture

Recap:

- Decision trees and linear models

New topics:

- K nearest neighbors (K-NN)
- Multiclass classification
- Model assessment and selection

Today's Lecture

Recap:

- Decision trees and linear models

New topics:

- K nearest neighbors (K-NN)
- Multiclass classification
- Model assessment and selection

Recap: Linear models for classification and regression

- We previously studied parametric methods, such as perceptron, linear regression, and logistic regression.
- These methods are governed by the vector w and specifically how the value $w^T x$ maps to the output.
- Once the vector w is derived, we essentially no longer need the training data.

- Last time we introduced non-parametric methods, and the decision trees in particular.

Recap: Decision Trees

- Decision Tree is a greedy modeling method for classification (and regression).
- At each step, split on the most informative attribute.
  - Information Gain.

- Terminate under given rules.
- At test time, go down the decision tree with the given attributes and declare as the output the label we arrive at.

Today's Lecture

Recap:

- Decision trees and linear models

New topics:

- K nearest neighbors (K-NN)
- Multiclass classification
- Model assessment and selection

K Nearest Neighbors Algorithm

- Like the decision trees we studied last time, K-NN is a **non-parametric** method

- Learning part in K-NN is “easy”: store all the training examples.

- But the testing part is “hard” (computationally more intensive): potentially need to compare the test point against all training points.

Let’s consider K-NN for binary classification

- Example with K = 1:

- Would you classify point ‘o’ as being in Class 1 or in Class 2?
- It depends on what rule we use.

Distance measure

- In K-NN with K=1, we find the closest neighbor in the training set and assign the label of that point as the label of the test point.
- Closest in what sense ?
- It is natural to consider Euclidean distance:

At test time

- Suppose that the data is d-dimensional.
- Then, at test time we have:

- We can avoid taking √ so we just compare the squared norms:

Decision boundaries

- Decision boundaries are computed explicitly, i.e., there is no mathematical formula for it.
- For K=1, each decision boundary is equidistant to two points from the opposite classes.
- Example:

- These are also called Voronoi regions

Another example

- What about this case ?

- Would you really expect class 1 in the middle of class 2 ? No.
- These situations are typically due to label noise, mislabeled data.
- But, cause overfitting!

Solution for overfitting

- Increase K, so we look at more than 1 nearest neighbor.
- Why not try K = 2 ?
- Ok, so let’s look at 3-NN.

Procedure:

1. For a given test point, compute the distance from it to all training points (N distances).
2. Find 3 closest points.
3. Take the majority vote.

Back to the previous example

- Let’s revisit the last example:

- So if K = 3 is better than K = 1, does that mean K = 103 is better than K = 3 ?

Choice of K

- For small K, decision boundary is very fine grained; decision regions are squiggly patches (overfitting is a possible issue).
- For large K, decision boundary is smooth; decision regions are large uninterrupted segments (bias is a possible issue).
- As it turns out, 1-NN has misclassification rate at most twice what is achieved by an optimal classifier (mathematical proof) it does make sense to choose smaller K.

Choice of the distance measure

- There are other choices of the distance, that can be more appropriate, depending on the application.

- Hamming distance, e.g., for categorical data:

- Manhattan distance:

More on the practical issues

- When implementing K-NN with Euclidean distance, ensure that all dimensions are normalized, so that one dimension does not artificially skew the distance and dominate.

- Check for the presence of “duplicate” attributes, as these too can overly represent a given dimension.

- It is sometimes beneficial to weigh the vote by the distance:

K-NN for regression

- The previous procedure can be easily extended to the regression problem:
- The output value is the (weighted) average of the values of K nearest neighbors.

Today's Lecture

Recap:

- Decision trees and linear models

New topics:

- K nearest neighbors (K-NN)
- Multiclass classification
- Model assessment and selection

Multi-class classification

- The methods we saw thus far in the course, we primarily described in the context of the binary classification problems.
- We can extend these approaches to the **multi-class classification** problem as well.

- General techniques for going from binary to multi-class classification:

1. One vs. all (OVA)
2. All vs. all (AVA)

One vs. All in K classes

- For all labeled training points,

  For each class k, $1 \leq k \leq K$,

  Create a binary classifier : class k vs. class not-k

- Example: K=3: There are 3 binary classifiers.
  - Class 1 vs. class not-1 (which is the union of classes 2 and 3)
  - Class 2 vs. class not-2 (which is the union of classes 1 and 3)
  - Class 3 vs. class not-3 (which is the union of classes 1 and 2)

One vs. All in K classes

- Advantage: we train K binary classifiers, which we already know how to do.
- Unfortunately, these classifiers are imbalanced in their training data (consequence of the design).

- At test time, pick the class that chose the test point.
- Issue: more than one class chose it. Then the ties are often broken randomly.
- Issue: no class chose this point (no vote).

All vs. All in K classes

- Interpret the problem as the tournament with $\binom{K}{2}$ matches. Each match is a binary classifier.
- Example: K = 3. There are 3 matches: Class 1 vs. class 2; class 2 vs. class 3, class 1 vs. class 3.

- We now partition the data set so that in each match there are only points that are labeled with the match participants.
- Issue: training individual classifiers on a much smaller data set

All vs. All in K classes

- At test time, run all $\binom{K}{2}$ classifiers.
- The winner gets +1 point.
- Label of the test point is determined by the class that earned the most points.

- Issue: can also lead to ambiguity, e.g., in K = 3 case each class gets +1 point in each of the three matches.

Multiclass logistic regression

- Recall the binary case:

Multiclass logistic regression

- Recall the binary case, continued:

Let’s now generalize this result to K classes

- Conditional probability:

- This is known as a soft max function.
- As before, find w to maximize the likelihood:

Derivations – ctd.

- Recall the trick in the binary case:

- We employ a similar trick here:

Derivations – ctd.

- As in the binary case, take the log to bring the exponents down.

- Using differentiation, get the gradient:
  - Here y_{ij} is the indicator function that the training point x_i is in the Class j.

- Relate to the binary case:

Today's Lecture

Recap:

- Decision trees and linear models

New topics:

- K nearest neighbors (K-NN)
- Multiclass classification
- Model assessment and selection

How to assess the quality of the learned model ?

1. Use **validation**.
    - Recall that we already discussed this approach in the context of decision trees.
- How ? Split the data. Train on $\mathcal{D}_1$ and validate on $\mathcal{D}_2$.

- When we test on $\mathcal{D}_2$ we get validation error which can be used to estimate generalization error.

How to assess the quality of the learned model ?

2. Use **cross-validation**.

- How ? Train separate models as follows:

- If the size of the validation set is 1, this approach is called leave one out (LOO). Note that there are N models in that case. LOO can be done efficiently in the case of linear regression.

How to assess the quality of the learned model ?

2. Use **cross-validation**.

- More generally, we can use K-fold validation.
- There are K models.
- Data is partitioned so that each time different 1/K fraction of data is in the validation set.
- Train each model on its own (K-1)/K fraction of data and validate it on its own validation set (1/K fraction of data).

- Average these K resulting validation errors. This is the overall val. Error.

How to assess the quality of the learned model ?

- In both cases, retrain on all available data and discard the smaller models.

Model selection

- Perform model assessment for each choice of the hyperparameter.
- For example, K in K-NN is a hyperparameter.
- Choose the value with the lowest validation error and trainedon all data.