Machine Learning and Data Mining – Midterm Review

Philip Warton

October 27, 2021

Structure

Concept questions

- Machine Learning Terms
- kNN
- Probability / MLE / MAP
- Naive Bayes
- Lin. Reg.
- Log. Reg. and Perceptron
- SVMs

In-depth qestions

- kNN and Cross Validation
- Linear Classifiers and Regularization
- Naive Bayes
- SVMs

General Concepts

We are interested in mappings $f : X \to Y$ from an input to an output.

We have 2^{2^d} mappings even with only binary data, so there is no way to figure out the true function without some assumptions. For example, kNN assumes that labels change smoothly as features change in local regions. That is, locality matters. For logistic regression, we assume that

- the relationship between input/output can be derived from a linear function
- label changes smoothly
- independent variables

Hypothesis space, $\mathcal{H}$. Find a function $g \in \mathcal{H}$ such that $g \approxeq f$ where $f : X \to Y$ is the "true function". Modeling error is the difference between the best $g \in \mathcal{H}$ and f. That is, f lies outside $\mathcal{H}$. Estimation error is the difference between g_D and g given some dataset D. Optimization error is the difference beween a chosen function g and the best function given that dataset g_D. Linear regression has an analytic answer and therefore no optimization error. Overfitting: good on train bad on test. Underfitting: bad on train bad on test.

kNN

Nearby points determine label (majority vote). $k = 1$ tends to be overfit. As k becomes closer to n we start to simply take the majority class. The algorithm is given by

$$f_k(x) = \frac{1}{k} \sum_{i \in \mathbb{N}} ||y_i||$$

Some properties:

- computationally expensive
- $O(nd)$ for every test points
- Lots of work to speed this up with with smart data structures
- For massive datasets, it requires lots of memory, remove "unimportant examples"

Scale of values is important, so features should be normalized. Irrelevant features contribute to distance. Hyperparameters:

- Metric d
- k
- Input function
- Output function

Cross validation: training on different "chunks"/"folds"