

Lecture 3: Supervised learning introduction

CSI 5v93: Introduction to machine learning

Baylor University
Computer Science Department

Dr. Greg Hamerly
<http://cs.baylor.edu/~hamerly/>

CSI 5v93: Introduction to machine learning, Lecture 3 – p. 1/16

Questions?

CSI 5v93: Introduction to machine learning, Lecture 3 – p. 2/16

Chapter 2: Overview of supervised learning

- 2.1 – Introduction
- 2.2 – Variable types and terminology
- 2.3 – Two simple approaches to prediction: Least squares and nearest neighbors
- 2.4 – Statistical decision theory
- 2.5 – Local methods in high dimensions
- 2.6 – Statistical models, supervised learning, and function approximation
- 2.7 – Structured regression models
- 2.8 – Classes of restricted estimators
- 2.9 – Model selection and the bias-variance tradeoff

Dimensionality problems (2.5)

Even though the k -nearest neighbor classifier seems like it will do everything we need, it has problems when the input dimension is high.

This is known as the “curse of dimensionality”, and it affects every learning algorithm (though some worse than others).

The curse: an example

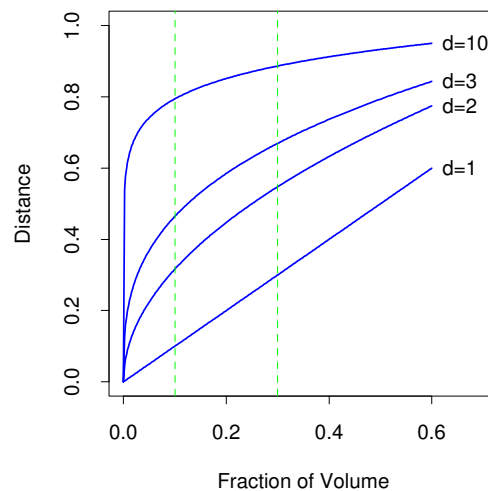
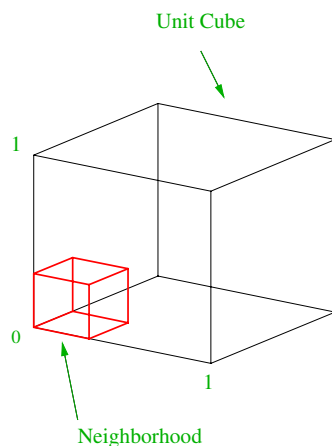
Basic idea: in higher dimensions, everything looks far away.

Example explanation: the Euclidean distance formula is

$$(1) \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

As d increases, there are more added terms. Distances get large.

The curse: another example



Right graph shows the required sidelength to capture the given fraction of volume in a unit hypercube for different dimensions.

For $d = 10$, to capture just 10% of the volume requires a box with sidelength 80%!

Practical implications for the curse

Imagine the interval $[0, 1]$, that you have n points drawn from the interval.

They have a density of n .

To have the same density in d dimensions ($[0, 1]^d$), you need roughly exponentially more points!

$$n^d$$

Higher dimension often means we need exponentially more data to learn effectively.

Statistical models (2.6)

Our model for quantitative learning problems is often an additive-error model:

$$Y = f(X) + \varepsilon$$

where $E[\varepsilon] = 0$ and ε is i.i.d. and independent of X

We can model other assumptions (such as ε_i depends on x_i), but this independent model is simplest and most often used.

Learning

Our model:

$$Y = f(X) + \varepsilon$$

Learning can be viewed as:

- start with a model \hat{f}
- given (x_i, y_i)
- modify \hat{f} so that $\hat{f}(x_i) - y_i$ gets small

Learning as function approximation

Our model:

$$Y = f(X) + \varepsilon$$

Learning can also be viewed from a *function approximation* point of view, where $\{x_i, y_i\} \in \mathbb{R}^{d+1}$

This approach is less romantic but allows us to use distances, Euclidean spaces, probabilistic inference, and other tools.

Parameter terminology

θ is used to represent the model parameters we will learn, in general.

$\hat{\theta}$ represents our estimate of the parameters.

With linear regression, for example, $\theta \equiv \beta$.

If you see $f_{\theta}(X)$, that is the model f for the specific value of parameters θ .

Linear basis expansions

Besides linear regression and nearest-neighbors, another class of useful approximators is *linear basis expansions*:

$$f_{\theta}(x) = \sum_{k=1}^K h_k(x)\theta_k$$

Here, the $h_k(x)$ can be any function of x , and θ_k are the parameters to learn.

We can learn this using least-squares, and it will have a unique answer if all the $h_k(x)$ s do not have any free parameters:

$$RSS(\theta) = \sum_{i=1}^n (y_i - f_{\theta}(x_i))^2$$

Maximum likelihood estimation

Rather than using *least-squares* for estimating the parameters, we can use a method called *maximum likelihood*.

Central idea:

- a probabilistic model gives a probability to a datapoint
- change the probabilistic model to give maximum probability to the training set (all points)

The parameters of the model that give the max probability are called the *maximum likelihood estimate* or *MLE*.

For the additive-error model with Gaussian noise, least-squares gives the same solution as maximum likelihood. In general this is not the case.

Bayes' rule

Bayes' rule is used a lot in machine learning for manipulating probabilities.

$$\Pr(A|B) = \frac{\Pr(B|A) \Pr(A)}{\Pr(B)}$$

Usually we use this rule to manipulate probabilities for models:

$$\Pr(\theta|Y) = \frac{\Pr(Y|\theta) \Pr(\theta)}{\Pr(Y)}$$

So we have:

- $\Pr(Y|\theta)$ is our model
- $\Pr(\theta)$ is our *prior* (could be uniform)
- $\Pr(Y)$ is not usually important

Maximum likelihood estimation

Given that we can compute the probability of some value $\Pr_{\theta}(y_i)$, define the probability of the whole dataset as either:

$$\Pr_{\theta}(Y) = \prod_{i=1}^n \Pr_{\theta}(y_i) \qquad \Pr(Y|\theta) = \prod_{i=1}^n \Pr(y_i|\theta)$$

We can define this probability in terms of θ , assuming that Y is fixed, giving the *log likelihood*:

$$\mathcal{L}(\theta) = \log \Pr(\theta|Y) = \sum_{i=1}^n \log \Pr(y_i|\theta)$$

This is the term to maximize, we want

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta)$$

2-minute journal

Please write a response to the following on a piece of paper and hand it in immediately. Please make it anonymous (no names). Write about:

- major points you learned today
- areas not understood or requiring clarification