INTRODUCTION TO MACHINE LEARNING COMPSCI 4ML3

LECTURE 5

HASSAN ASHTIANI

THE TRADE-OFF

- A powerful/flexible curve-fitting method
  - Small training error
  - Requires more training data to generalize
  - Otherwise large test error
- A less flexible curve-fitting Method
  - Larger training error
  - Requires less training data
  - Smaller difference between training and test error
- The so-called "bias-variance" trade-off

THE CASE OF MULTIVARIATE POLYNOMIALS

- Assume $M \gg d$
- Number of terms (monomials): $\approx (\frac{M}{d})^d$
- #training samples $\approx$ #parameters $\approx (\frac{M}{d})^d$
  - #training samples should increase exponentially with d
  - Susceptible to over-fitting…
  - An example of **curse of dimensionality**!
- We can say **sample complexity** of learning multivariate polynomials is exponential in d
  - Orthogonal to computational complexity

Model selection: how to avoid overfitting?

- Selecting M (the complexity of the model)
  - based on d (dimension) and n (number of samples)
- More practically, try several options for M
  - Use a **holdout (evaluation) sample**
  - Never use test data to tune parameters!

AVOID OVERFITTING WITH REGULARIZED LEAST SQUARES

$$\min_{W \in \mathcal{R}^d} \|XW - Y\|_2^2 + \lambda \|W\|_2^2$$

- Encourage a solution with a smaller norm
- $W^{RLS} = (X^T X + \lambda I)^{-1} X^T Y$
  - Exercise: prove that this is the optimal solution
- Does the inverse always exist?
  - Yes! (Exercise: Prove)
- How to choose λ?

POLYNOMIAL CURVE-FITTING REVISITED

- MAP THE INPUTS x^i TO A HIGHER DIMENSIONAL SPACE
  - A KIND OF "PRE-PROCESSING" THE DATA
- DO LINEAR REGRESSION ON THE HIGH-DIMENSIONAL SPACE
  - EQUIVALENT TO PERFORMING NON-LINEAR REGRESSION IN THE ORIGINAL SPACE
- MAP $\phi(x): \mathrm{R}^{d_1} \longmapsto \mathrm{R}^{d_2}$ WHERE $d_2 \gg d_1$
- $\phi(x) = \begin{pmatrix} \phi_1(x) \\ \dots \\ \phi_{d_2}(x) \end{pmatrix}$ IS NONLINEAR, E.G., $x \in R$ AND $\phi(x) = \begin{pmatrix} x \\ x^2 \\ x^3 \\ \dots \\ x^{d_2} \end{pmatrix}$
- WHAT IF d_2 IS MUCH LARGER THAN THE NUMBER OF SAMPLES?

CURVE-FITTING WITH BASIS FUNCTIONS

- Feature map: $\phi(x): \mathrm{R}^{d_1} \longmapsto \mathrm{R}^{d_2} \quad d_2 \gg d_1$
- $\Phi_{n \times d_2} = [\phi(x^1) \quad \dots \quad \phi(x^n)]^T$
- Training
  - $\boldsymbol{W^* = \min_W \|\Phi W - Y\|_2^2 + \lambda \|W\|_2^2}$
  - $\boldsymbol{W^* = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T Y}$
- Prediction
  - $\boldsymbol{\hat{y} = < W^*, \phi(x) > = W^{*T} \phi(x)}$

OTHER CHOICES OF $\phi(x)$

- Pick a fixed (nonlinear) $\Phi(x)$
  - Encodes your prior knowledge about the data
  - Feature engineering!
- Polynomial basis functions
- Gaussian basis functions:
  - $\phi_i(x) = e^{-\frac{\|x-\mu_i\|_2^2}{2\sigma^2}}$
- DFT (FFT), wavelet for time series
- Is it possible to learn the mapping $\phi_i(x)$ itself?
  - Later, e.g., neural networks

COMPUTATIONAL COMPLEXITY OF NAÏVE RLS

- TRAINING: CALCULATE $w^{RLS} = (\phi^T\phi + \lambda I)^{-1}\phi^T Y$
- BOTTLENECK: MATRIX INVERSION
  - HOW MANY OPERATIONS?

- PREDICTION: $\hat{y} =< \phi(x), w^{RLS} >$
  - HOW MANY OPERATIONS?

- REGULARIZATION ALLOWS US TO GO INTO HIGH-DIMENSIONAL SPACE WITHOUT OVERFITTING, BUT IT DOES NOT SOLVE THE COMPUTATIONAL PROBLEM

COMPUTATIONAL COMPLEXITY

- Matrix multiplication (n-by-n matrices)
  - Native method: $O(N^3)$
  - Strassen's Algorithm: $O(N^{2.8074})$
  - Coppersmith–Winograd-like algorithms [current best $O(N^{2.3728639})$]
- Matrix inversion
  - Gaussian Elimination: $O(N^3)$
  - Possible to reduce it to multiplication

THE COMPUTATIONAL PROBLEM

- CAN WE SOLVE THE REGULARIZED LEAST SQUARES IN R^{d_2} **WITHOUT EXPLICITLY MAPPING THE DATA** INTO R^{d_2}?
  - $W^* = \underset{W \in R^{d_2}}{min} \|\Phi W - Y\|_2^2 + \lambda \|W\|_2^2$
- SOMETHING LIKE MULTIPLICATION USING FFT
- IF SO, WE COULD EVEN MAP THE DATA TO AN INFINITE DIMENSIONAL SPACE!!

FFT AND MULTIPLICATION