Mathematics For Data Science

Brian J. Mann

Feb 2, 2016

Who am I?

- PhD from University of Utah in 2014 working under Mladen
- Worked for Amazon Web Services doing supply chain optimization and forecasting
- Now at Galvanize teaching data science and consulting
- Slides

What is Data Science?

- Learning patterns or behavior from observed data, generally to predict behavior of new observations
- Uses statistics, computer science, machine learning

Examples

- Predict when to build new data centers accounting for a noisy demand signal (this is what I did at AWS)
- Given a satellite photo of a whale at the surface of the ocean, determine which particular whale it is (NOAA Right Whale Kaggle competition)
- Determine whether the effect of changing the UI on your company's phone app was significant
- Given a photo of an eye, determine if the individual has diabetic retinopathy

Intro to Classification Algorithms

- Observed data represented by points in $\mathbb{R}^N$
- Training observations labelled "positive" or "negative" (we'll use $\{+1, -1\}$)
- Goal: create a model function $g : \mathbb{R}^N \to \{+1, -1\}$ that predicts the class of new observations.

How?

- Use the training data!
- Find a function g that minimizes the error on the training set *without overfitting*
- Think about trying to model a trend on 20 data points with a 20 degree polynomial

Support Vector Machines (SVM)

- Idea: try to separate classes by an optimal hyperplane
- Here *optimal* means that the minimum distance from the hyperplane to any of the training points (the *margin*) is maximal.

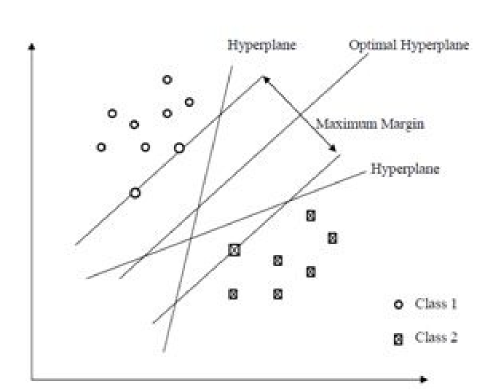

Figure 1:Maximal Margin

More SVM

- Quadratic optimization problem
- Can be solved via the method of Lagrange Multipliers
- The optimal hyperplane in SVM has the form

$$f(x) = \sum_i a_i \langle x_i, x \rangle + b = 0$$

  where $\{x_i\}$ are your training observations and $a_i \neq 0$ if and only if x_i is a *support vector* (the points on the edge of the margin)
- Let $g(x) = sgn(f(x))$

Ok, that sounds great. What's the problem?

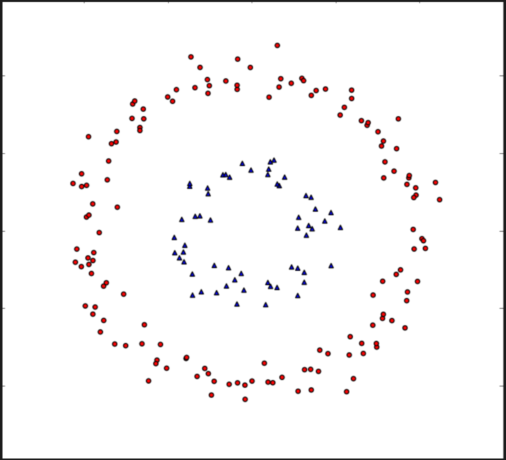

Figure 2:Well, shit.

The solution

- Map our data to a higher dimensional space where it's (almost) linearly separable

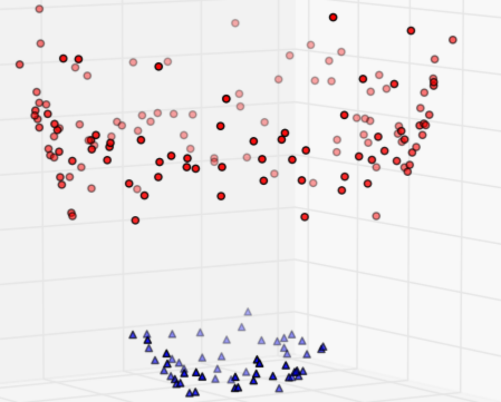

Figure 3:Yay!

Talk's over, right?

- Not quite, there's still some problems

Issue 1: Memory

- Even for polynomial transformations, the numbers of dimensions (features) in the target space can grow very quickly
- Consider the transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^5$ given by

$$(x_1, x_2) \mapsto (x_1^2, x_1 x_2, x_2^2, x_1, x_2, 1)$$

- More generally, a mapping $\phi_d : \mathbb{R}^N \to \mathbb{R}^{\binom{N+d}{d}}$ that maps a vector to the vector of all monomial terms in N variables of degree $\leq d$
- $\binom{N+d}{d}$ grows *very* quickly as $d >> 0$

Issue 2: Computation

- The optimal hyperplane for the transformed data is

$$f(x) = \sum_i a_i \cdot \langle \phi(x_i), \phi(x) \rangle + b = 0$$

- Need to compute the dot product of high-dimensional vectors (in fact, sometimes they might be infinite dimensional!)

A solution

- What if there was a way to compute $\langle \phi(x), \phi(y) \rangle$ directly without ever computing $\phi(x)$ or $\phi(y)$?
- There is!
- This is what *kernel functions* do for us

Kernels

- Make ϕ implicit
- This implicit ϕ might have an infinite dimensional target vector space

What is a kernel function?

A *kernel function* is a continuous function

$$K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$$

which satisfies

1. $K(x, y) = K(y, x)$ (symmetric)
2. K is positive-semidefinite i.e.

$$\sum_i \sum_j K(x_i, x_j) c_i c_j \geq 0$$

for all finite sequences $x_1, \ldots, x_n$ and all $c_i, c_j \in \mathbb{R}$

Mercer's Theorem

Mercer's Theorem says that if $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a kernel function, then there exists a vector space with an inner product (a *Hilbert space*) V and a mapping $\phi : \mathbb{R}^N \to V$ so that

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

- In English, if K is a kernel function, it consists of a transformation followed by an inner product in some higher dimensional space V.
- Kernels allow us to compute high-dimensional inner products in V in terms of our original inputs in $\mathbb{R}^N$.

Example: Polynomial kernel

- $K(x, y) = (\langle x, y \rangle + c)^d$
- c and d are chosen *a priori* by the user, not trained
- Comes from the polynomial transformation ϕ_d

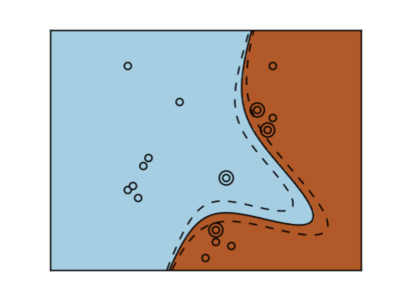

Figure 4:Polynomial Kernel

Example: RBF (Gaussian) kernel

- $K(x, y) = \exp(-\gamma||x - y||^2)$
- γ is chosen *a priori* by the user

More RBF kernel

What are ϕ and the dimension of V in this case?

- Let $\gamma = 1/2$ for ease of computation, then

$$K(x,y) = \sum_{j=0}^{\infty} \frac{\langle x, y\rangle^j}{j!} \exp(\frac{-||x||^2}{2}) \exp(\frac{-||y||^2}{2})$$

- With a little algebra one gets

$$\phi(x) = \left(\frac{e^{-\frac{||x||^2}{2j}}}{\sqrt{j!}^{1/j}} \binom{j}{n_1, \ldots, n_k}^{1/2} \right)_{j=0,\ldots,\infty, \sum_{i=1}^k n_i = j}$$

- V is infinite dimensional ($V = l^2$ the space of square-summable sequences)

More RBF kernel

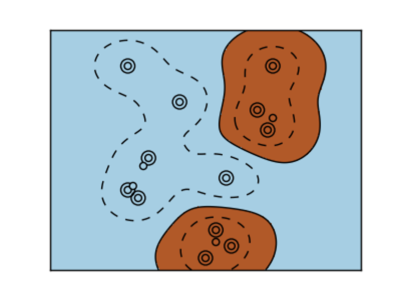

Figure 5:RBF Kernel

Questions?

Tips for transitioning to industry

- I'll focus on data science, but a lot of this applies elsewhere

Learn a programming language

- Python
  - Popular, big active community
  - Scikit-Learn is one of the best machine learning libraries available
  - General purpose programming language - not just for math and statistics
- R
  - Popular, but fewer contributors
  - Designed with data and statistics in mind
  - Not so great as a general purpose language, but great for *ad hoc* data analysis

More programming languages

- Scala
  - Higher barrier to entry than Python/R
  - Compiles to Java bytecode, so it can use any Java package
  - Functional
  - Will be able to use it at company that uses Java
- Java
  - Immensely popular in the software development industry
  - Not so great with data analysis and statistics
  - Standard in CS cirriculum.

Get connected

- Find colleagues or friends in industry to refer you
  - Much much higher success rate than just submitting your resume online
- Use LinkedIn and Twitter
- Write a technical blog
- Start writing some code and use github

Speaking of github

- Learn to use version control (git)

Focus on getting good at just a few things

- Stick to one programming language to start (Python)
- Pick a goal job and focus on the skills needed to get it
    - Data scientist: stats, machine learning, data cleaning, basic programming and CS skills
    - Software developer: Java, Scala, or Python. CS fundamentals (data structures, algorithms)

Machine Learning

- Andrew Ng Machine Learning course on Coursera
- *Learning from Data*, Abu-Mostafa, Magdon-Ismail, Lin
- *Introduction to Statistical Learning*, James, Witten, Hastie, Tibshirani
- Kaggle competitions

Try before you buy

- If you think we might want to go into industry, get a summer internship
- Looks great on your resume
- Builds your professional network