Machine Learning Practice and Theory

Day 2 - Mathematical Background

Govind Gopakumar

IIT Kanpur

Announcements

- Pre-Course survey
- Programming assignments
- Project ideas and partners
- Installation of Jupyter / IPython notebook
- Webpage : govg.github.io/acass

Recap

Machine Learning

- Trends in data
- Using the right model, and reasonable loss functions
- Transforming the problem according to simplicity

Divisions in Machine Learning

- Unsupervised learning : goal is to discover patterns in data
- Supervised learning : goal is to predict some aspect using data

Overview

Notations

Dealing with data :

- X : Data matrix (NxD)
- Y : Label matrix (Nx1)
- w : Model parameters
- L(X,Y,w) : Loss of model w on X,Y

Dealing with model:

- λ : Hyper parameters of a model
- w^* : Optimal model (may or may not be unique)

Mathematics in Machine Learning

- How do we describe and manipulate data?
  Use a matrix!
- How do we “model” something?
  Use a vector, or a function!
- How do we analytically solve models?
  Use Linear Algebra!
- How do we mathematically “learn”?
  Use Calculus, Linear Algebra!

Probability

Basics

Definitions

- Event : Some occurence that is desirable
- Sample space : All possible events
- $P(a) = \frac{\|a\|}{\|a\| + \|a'\|}$

Terms

- $\prod p(a_i)$ - probability of multiple events
- Can also model likelihood of event
- Naturally leads to MLE (general technique, to be covered later)

Random Variables

What are they?

- Map between events and some value
- Represented as a probability distribution function
- Discrete, continuous, categorical etc

How do we use them?

- Describe p(a) for a random variable
- Examples include normal, beta, poisson
- Integrate to 1

Distributions - I

Continuous

- Gaussian : Model any real number distribution
- Beta : Model number between [0,1]
- Dirichlet : Model a vector that sums to 1

Discrete

- Bernoulli : Model number of heads in a coin toss
- Poisson : Model counts of a variable

These can be combined together (joint, marginal)

Distributions - II

Gaussian distribution :

- $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(x-\mu)^2}$
- μ : Mean of the distribution
- σ^2 : Variance of the distribution

Multivariate Gaussian :

- $p(x) = \frac{1}{\sqrt{2\pi^k |\Sigma|}} e^{\frac{-1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$
- μ : Mean vector
- Σ : Covariance matrix

Distributions - III

Multiple variables :

- Define a "joint" distribution
- Denote by p(v,u)
- Is this the same as p(u)*p(v)? When is it not?

Examples in terms of Gaussians :

- Consider two variables, $v \sim \mathcal{N}(\mu_v, \sigma_v)$, $u \sim \mathcal{N}(\mu_u, \sigma_u)$
- How does the joint distribution look?
- What if they were drawn from a 2D Gaussian?
- When does the second case reduce to the first?

Bayes theorem - I

Invert the event!

- Reverse the probability of events
- $P(a|b) = \frac{P(b|a)P(a)}{P(b)}$

Terms in this expression

- $P(a|b)$ - called the posterior
- $P(b|a)$ - called the likelihood
- $P(a)$ - called the prior

Bayes theorem - II

Setting

- B : Color of the ball
- A : Selection of box
- $B_1(1,1,1), B_2(2,0,0), B_3(0,0,1)$
- All boxes are equally likely

Inverting the event

- P(b | a) : Probability that color was b given box is a.
- P(a | b) : Probability that box was a given color is b.
- How do we use Bayes theorem here?

Statistics

Statistics of a sample - I

Mean of sample

- $\mathbb{E}[X]$ - "average" of the distribution
- When can it be useless?
- When can it work as a representation?

Variances and covariances

- σ^2 - "spread" of the distribution
- Can be used to "normalize" data
- Can be used to see where data is useless

Generally, we do not come across other "moments" of the data in Machine Learning (skew, kurtosis etc).

Statistics of a sample - II

Of standard distributions

- Gaussian : Σ
- Bernoulli : $p(1-p)$

Of a sample

- Defined as "empirical" quantities
- Mean : μ
- Variance / Covariance
- Used in "moment matching" techniques

Linear Algebra

Spaces

Constituents :

- Vectors (v,u,w)
- Dot products
- Norms

Utility :

- Our data “lives” in some space
- Our model describes “shapes” in that space
- Must deal with math of this space!

Matrix Algebra

Basics

- Matrix (NxD) : Can denote a set of points
- Vector (1xD) : Denotes a single point
- Usually denotes our data

Properties

- Invertibility : $AA^{-1} = I$
- Definiteness : PD / PSD

Other terms

Eigenvalues

- $Av = \lambda v$: λ is an "eigenvalue"
- Denotes a direction in the space of the matrix

Measures of vectors

- $\|x\|_p$ - denotes the p-norm
- Different norms have different interpretations
- Similarities (cos, distance)

Functions and Optimization

Function shapes

Convexity

- Convex (and concave) functions have single optima
- Easy to optimize over
- Follow the slope method
- Closed under summation (this is very very nice and important!)

Smoothness and differentiability

- If a function is "smooth", it will be easy to find the slope.
- If it has kinks, slightly harder to find actual gradients!
- If it is discontinuous, no real way to find gradients!

Optimization theory

Basics :

- Gradient descent : how to follow the slope
- Simple gradients for simple loss functions
- Combine gradients for sum of functions

Examples of gradients :

- $(w - x)^2$: $2(w - x)$
- e^{-w} : $-e^{-w}$

Example of gradient descent

- For simple functions, easy to compute gradients
- General form of GD : $x^{t+1} = x^t - \eta g^t$
- Consider : $f(x) = (x + c)^2$
- Gradient : $g(x) = 2(x + c)$

Let's do gradient descent on this!

Modelling

Coin tossing : model

- What do we wish to model? : bias of coin (k)
- What data do we have? : H heads, T tails observed

MLE modelling

- p(H heads, T tails)?
- What can we do with this now?
- "Likelihood" can be our loss!
- What is the optimal choice here?
- Why could this fail?

Conclusion

Takeaways

- How to write down probability of events
- What the mean and variance tell us about a random quantity
- Why matrices are used in Machine Learning, how we manipulate them
- What sort of loss functions should we consider? How do we actually use them?

References

- Review lecture in CS771, IIT Kanpur
- Linear Algebra Overview
- Probability Overview
- Matrix Algebra Overview

Next Lecture overview

Our first classifier

Naive method of doing classification?

- Choose points which are nearby?
- Choose cluster which is nearby?

Formal "names"

- K-nearest Neighbors
- Distance from means

Distance from means - I

Overview of model

- Compute center of each class / label
- Assign the new point to closest mean
- What does “training” mean now?
- What does “testing” mean now?

Drawbacks and strengths?

- Storage?
- Time taken?
- When can this be a bad method?
- When can this be good?

Distance from means - II

Coming up with our "decision function"

- μ_+ : positive mean
- μ_- : negative mean
- $f(x^{new}) = d(x^{new}, \mu_-) - d(x^{new}, \mu_+)$

Geometry of the decision function

- What does the boundary look like for this?
- What can it learn? What can't it learn?

As similarity to training data

- $\|x^{new} - \mu_-\|^2 - \|x^{new} - \mu_+\|^2$
- $\langle \mu_+ - \mu_-, x^{new} \rangle + C$
- Can be simplified into : $f(x^{new}) = \sum \alpha_i \langle x_i, x^{new} \rangle + B$

What does this mean?

KNN - I

Overview of model

- Assign each point the class / value of its neighbor
- "K" - how many neighbors you account for
- What does "training" mean here?
- What would "testing" mean?

Drawbacks and strenghts?

- Storage?
- Time taken
- When can this be good or bad?

KNN - II

Geometry of the decision function

- What sort of boundary does this generate?
- How powerful can this be?
- The "distance" can always be measured in other forms!

Things to consider for this model

- What happens if we have outliers?
- Where could this be an issue?

KNN - III

What is the optimal K?

- What happens if we increase K?
- Consider limit of K -> N?
- What's the best choice then?

Extensions to KNN

- Can this be extended in the regression / labelling setting?
- Transformation of coordinates - How does that affect KNN?