Machine Learning

Xi Tan (xtan3.1415926@gmail.com)

January 13, 2019

Contents

Preface

This book project, which consists of four subjects: Finance, Mathematics, Statistics, and Computer Science, is tailored specifically to prepare someone for a quant career. It originated from my general belief of the hierarchy of solving a problem — problems are solved at strategic, tactical, and operational levels.

Microeconomics and *Macroeconomics* explain the driving forces of capital markets, from a legislator's perspective. *Accounting* and *Corporate Finance* take a closer and necessary look at these forces, from a different angle. *Stochastic Calculus* and *Asset Pricing* provide with a set of tools and ideas that enables us to **strategically** model one of the central problems in Quantitative Finance.

Generally speaking, there are two paths to solve a quantitative finance problem at the **tactical** level: the mathematical way and the statistical way. There are only two pieces of math we need to know: *Analysis*, in particular measure-theoretical probability and differential equations; and *Linear Algebra*, with functional analysis in mind. Statistics, on the other hand, should start with *Statistical Experiment Design*, from which we learn how to collect data for statistical models. Next, the study of *Random Variables* and *Stochastic Processes* introduce the building blocks of the statistical "pillbox", with *Mathematical Statistics* the "scaffold". Once the "pillbox" is ready, we are equipped to tackle our problems using *Machine Learning*, which is essentially a collection of statistical models and optimization algorithms.

Computer Architecture and *Operating System* are respectively about the "hardware" and "software" of a single computer. The interaction of multiple computers is understood in *Computer Network*. Once we are comfortable with these concepts, we will be able to use *Data Structure and Algorithms* to solve problems at the **operational** level, and use *C++* and/or *Java* to implement our ideas.

I am aware that it can take a while, and even multiple advanced degrees, to finish this curriculum, but let's remember the motto from the Leipzig Gewandhaus Orchestra: "***Res severa est verum gaudium***".

Xi Tan
West Lafayette, IN
October, 2013

Chapter 1

Introduction

There are two main types of Machine Learning problems: the **predictive** or **supervised learning**, and the **descriptive** or **unsupervised learning** [1].

For supervised learning, there are two subtypes: the **classification** problem, where the response variable is categorical (either ordinal or nomial); and the **regression** problem, where the response variable is numerical (either discrete or continuous).

For unsupervised learning, it is also called **density estimation** in Statistics literature. Essentially we want to build models of the form $p(\mathbf{x}_i|\theta)$, and supervised learning can be seen as to build models of the form $p(y_i|\mathbf{x}_i, \theta)$, which is a problem of conditional density estimation. [2]

1.1 Some Distinctions

1.1.1 Machine Learning v.s. Statistical Learning

Machine Learning focuses more on high dimension low noise situations, and performance and efficiency are the main concerns. **Statistical Learning** focuses more on low dimension high noise situations, interpretation and statistical inference are the main concerns.

1.1.2 Parametric v.s. Non-parametric Models

A parametric model has a fixed *finite* number of parameters, while the number of parameters in a non-parametric model grows with the amount of training data. A model with infinite number of parameters is usually non-parametric.

[1] It is also called **Knowledge Discovery** in the Data Mining literature

[2] We see from the formulation that, unsupervised learning usually involves multivariate probability models while supervised learning usually involves univariate probability models.

Chapter 2

A Brief History of Machine Learning

The perceptron model was invented in 1957, and it generated over optimistic view for AI during 1960s. After Marvin Minsky pointed out the limitation of this model in expressing complex functions, researchers stopped pursuing this model for the next decade.

In 1970s, the machine learning field was dormant, when expert systems became the mainstream approach in AI. The revival of machine learning came in mid-1980s, when the decision tree model was invented and distributed as software. It is also in mid 1980s multi-layer neural networks were invented, With enough hidden layers, a neural network can express any function, thus overcoming the limitation of perceptron. We see a revival of the neural network study.

Around 1995, SVM was proposed and have become quickly adopted.

After year 2000, Logistic regression was rediscovered and re-designed for large scale machine learning problems . In the ten years following 2003, logistic regression has attracted a lot of research work.

We discussed the development of 4 major machine learning methods. There are other method developed in parallel, but see declining use today in the machine field: Naive Bayes, Bayesian networks, and Maximum Entropy classifier (most used in natural language processing).

In addition to the individual methods, we have seen the invention of ensemble learning, where several classifiers are used together, and its wide adoption today.

http://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence

1950s-1960s: the Perceptron Model 1970s-1980s: the Expert Systems mid 1980s: Multi-layer Neural Networks 1995: SVM 2000s: Logistic Regression Rediscovered

Chapter 3

Regression v.s. Classification

Consider a supervised learning problem in which we wish to approximate an unknown target function $f : \mathcal{X} \to \mathcal{Y}$, or equivalently $P(Y|X)$. One way to learn $P(Y|X)$ is to use the training data to estimate $P(X|Y)$ and $P(Y)$, and then use Bayes rule to determine $P(Y|X)$.

Chapter 4

Parametric v.s. Nonparametric Models

Chapter 5

Decision Trees

Chapter 6

Clustering

Chapter 7

Dimension Reduction

Chapter 8

Graphical Models

The motivation of graphical models is, in the previous settings, variables are assume to be (conditionally) independent of each other: such as observed variables $\mathbf{x}$ and/or $\mathbf{y}$; or latent variables $\mathbf{z}$ (excluding parameters $\boldsymbol{\theta}$). Graphical models consider the case when variables are correlated.

Chapter 9

Neural Networks

Chapter 10

Kernel Methods

10.1 Support Vector Machines (SVM)

10.2 Gaussian Processes (GP)

Chapter 11

Statistical Learning Theory

Statistical learning theory studies the properties, in particular error-bounds, of learning algorithms in a statistical framework.

The No Free Lunch theorem says, if no assumptions about how training data are related to the testing data, prediction is impossible; furthermore, if no assumptions about the data to be expected, generalization is impossible.

Simply means we need to make the assumption that there is a stationary distribution of data.

Definition 11.0.1 Suppose $f : \mathbb{R} \to \mathbb{R}_+$ and $g : \mathbb{R} \to \mathbb{R}_+$, we write

$$f = O(g) \tag{11.1}$$

if there exists $x_0, \alpha \in \mathbb{R}_+$ such that for all $x > x_0$ we have $f(x) \leq \alpha g(x)$. Replacing $\leq$ with $\geq$, we write $f = \Omega(g)$.

Definition 11.0.2 Suppose $f : \mathbb{R} \to \mathbb{R}_+$ and $g : \mathbb{R} \to \mathbb{R}_+$, we write

$$f = o(g) \tag{11.2}$$

if for every $\alpha > 0$ there exists x_0 such that for all $x > x_0$ we have $f(x) \leq \alpha g(x)$. Replacing $\leq$ with $\geq$, we write $f = \omega(g)$.

Definition 11.0.3 If $f = O(g)$ and $f = \Omega(g)$, then we write $f = \Theta(g)$.

Definition 11.0.4 We write $f = \tilde{O}(g)$ if there exists $k \in \mathbb{N}$ such that $f(x) = O(g(x) \log^k(g(x)))$.

Chapter 12

Latent Dirichlet allocation (LDA)

Chapter 13

Principle Component Analysis (PCA)

Chapter 14

Linear Discriminant Analysis (LDA)

Chapter 15

Expectation Maximization (EM)

Chapter 16

Expectation Propagation (EP)

Chapter 17

Markov Chain Monte Carlo Methods

17.1 Introduction

This note is based on Peter Orbanz's BNP notes:

http://people.stat.sc.edu/hansont/stat740/MCMC.pdf

17.2 Notation

Bold upper case letters represent matrices, e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{\Theta}$. Bold lower case letters represent vector-valued random variables and their realizations (we do not distinguish between the two), e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\theta}$. Curly upper case letters represent spaces (i.e., possible values) of random variables, e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \Theta$.

17.3 Introduction

Markov chain Monte Carlo (MCMC) methods can be used to draw random samples from a target distribution p. It is particularly useful in Bayesian data analysis, due to the difficulties of evaluating the denominator in the Bayes' formula, a.k.a. the partition function.

1. A discrete-time, discrete-space Markov chain is $X^{(0)}, X^{(1)}, \ldots$ where $X^{(t)}$ obeys the Markov property that

$$P\left[X^{(t)}\middle|x^{(0)}, \ldots, x^{(t-1)}\right] = P\left[X^{(t)}\middle|x^{(t-1)}\right] \tag{17.1}$$

2. A Markov chain is *irreducible* if any state j can be reached from any state i in a finite number of steps for all i and j.

3. A Markov chain is *periodic* if it can visit certain portions of the state space only at regularly spaced intervals.

The MCMC sampling strategy is to construct an irreducible, aperiodic Markov chain for which the stationary distribution equals the target distribution p.

Suppose we want to draw samples from $p(x)$. The M-H algorithm proceeds as follows: Draw a candidate state, x^*, according to the proposal distribution $g(x^*|x)$, by computing the acceptance probability

$$\alpha(x^*, x) = \min\left[1, a(x^*, x) = \frac{p(x^*)g(x|x^*)}{p(x)g(x^*|x)}\right]. \tag{17.2}$$

where $a(x^*, x)$ is called the M-H ratio, and $\alpha(x^*, x)$ the probability of move. With *probability of move* $\alpha(x^*, x)$, set the new state, x' to x^*. Otherwise, let x' be the same as x. The intuition behind the probability of move is that, if the detailed balance condition is satisfied: $p(x)g(x^*|x) = p(x^*)g(x|x^*)$, then we are done, otherwise, the denominator $g(x^*|x)p(x)$ is proportional to the probability of moving from x to x^*, if it is large then the numerator, which is proportional to the probability of moving from x^* to x, then we should penalize it.

The sampled sequence may contain duplicated copies of data points, the frequency of which is used to correct the difference between the proposal distribution and the target one. A well chosen proposal distribution produces candidate values that efficiently cover the support of the target distribution.

17.4 Independence Chains

If we choose the proposal distribution to be

$$g(x^*|x) = g(x^*) \tag{17.3}$$

then the M-H ratio is

$$a(x^*, x) = \frac{p(x^*)g(x)}{p(x)g(x^*)}. \tag{17.4}$$

For example, in a Bayesian framework, if the target distribution is the posterior $p(\theta|\mathbf{y})$, where $\mathbf{y}$ is the data. Then, if we choose the proposal distribution to be the prior $p(\theta)$

$$g(\theta^*|\theta) = p(\theta^*) \tag{17.5}$$

then the M-H ratio is

$$a(\theta^*, \theta|\mathbf{y}) = \frac{p(\theta^*|\mathbf{y})p(\theta)}{p(\theta|\mathbf{y})p(\theta^*)} = \frac{p(\mathbf{y}|\theta^*)p(\theta^*)/p(\mathbf{y})}{p(\mathbf{y}|\theta)p(\theta)/p(\mathbf{y})}\frac{p(\theta)}{p(\theta^*)} = \frac{p(\mathbf{y}|\theta^*)}{p(\mathbf{y}|\theta)} \tag{17.6}$$

So if the proposal distribution is the prior, the M-H ratio is the likelihood ratio.

17.5 Random walk chains

Let x^* be generated by setting

$$x^* = x + \epsilon, \quad \epsilon \sim h(\epsilon) \tag{17.7}$$

or equivalently,

$$g(x^*|x) = h(x^* - x) \tag{17.8}$$

For example, h can be the uniform, or the standard normal, or the Student's t distribution.

17.6 Gibbs sampler

Suppose it is easy to sample from the univariate conditional distributions:

$$x_i|\mathbf{x}_{-i} \sim f(x_i|\mathbf{x}_{-i}) \tag{17.9}$$

then the basic Gibbs sampler can be described as follows:

1. Select starting values $x^{(0)}$ and set $t = 0$.
2. Generate, in turn for $i = 1, \ldots, n$:

$$x_i{}^{(t+1)}|\mathbf{x}_{-i}^{(t)} \sim f\left(x_1|\mathbf{x}_{-i}^{(t)}\right). \tag{17.10}$$

3. Increment t and go to step 2.

A hybrid MCMC may contain different types of samplers. For example, The M-H within Gibbs algorithm is typically useful when the univariate conditional density for one or more elements is not available in closed form.

17.7 Test for Convergence

1. Burn-in.
2. Run multiple chains, and if the within- and between-chain behaviors are similar, suggests that the chains are stationary. Gelman-Rubin statistic.
3. Plot samples against time, or log-likelihood against time.
4. Autocorrelation function (ACF) plot: lag versus correlation. Slow decay suggests poor mixing.
5. Re-parameterize the model may help.
6. Burn-in should be about 5000 iterations, chain lengths should be about 100 times the burn-in.
7. Standard error should be less than 5% of the standard deviation.

17.8 How it is used

Marginalization: just ignore others. Mean and variance: use samples. Probability estimates: estimated by the frequencies. Standard error of estimates: batch runs to obtain estimates and compute mean and standard error (divided by the square root of batch size). Density: kernel density or simply histogram.

17.9 Advanced MCMC methods

Slice sampling and other auxiliary variable methods, reversible jump MCMC, perfect sampling, Hit-and-run (choose a direction and then a distance to run), multi-try (choose from a set of candidates), Langevin M-H (random walk with drift) and etc.

17.9.1 Slice sampling

Introduce an auxilary varialbe u, and if we can sample from $f(x,u) = f(x)f(u|x)$ then dropping u and retain x as desired. The slice sampling works as follows:

$$u^{(t+1)}|x^{(t)} \sim \text{Unif}\left(0, f\left(x^{(t)}\right)\right) \tag{17.11}$$

$$x^{(t+1)}|u^{(t+1)} \sim \text{Unif}\left(x : f(x) \geq u^{(t+1)}\right) \tag{17.12}$$

It is particularly useful for multi-modal problems (but not for high dimensional ones).

17.9.2 Reversible Jump MCMC

RJMCMC is suitable for nonparametric models where model dimensions change. The key is to use auxiliary variables to match the dimensions.

17.10 Introduction

17.10.1 Sampling Methods in General

Inverse Transform Sampling

Importance Sampling

17.10.2 Rejection Sampling

Suppose we want to sample from $P(X)$ by utilizing a *proposal distribution* $Q(X)$, from which we can take samples easier. Let $C = \sup\left\{\frac{P(x)}{Q(x)}, \forall x\right\}$, so $\frac{P(x)}{CQ(x)} \leq 1, \forall x$.

We propose a new value x' from $Q(X)$ and accept this new value with probability

$$A(x'|x) = \frac{P(x')}{CQ(x')} \tag{17.13}$$

This is called the *rejection sampling* method.

Remark 17.10.1 If we plug in Equation 17.13 into Equation 17.25, then

$$LHS = P(x)Q(x'|x)\frac{P(x')}{CQ(x')} = P(x)Q(x')\frac{P(x')}{CQ(x')} = \frac{1}{C}P(x)P(x') \tag{17.14}$$

$$RHS = P(x')Q(x|x')\frac{P(x)}{CQ(x)} = P(x')Q(x)\frac{P(x)}{CQ(x)} = \frac{1}{C}P(x')P(x) \tag{17.15}$$

which shows that the rejection sampling ratio in Equation 17.13 is a special solution to the detailed balance equation.

17.11 Markov Chains

Definition 17.11.1 A *Markov chain* is a collection of random variables $\{X^{(0)}, X^{(1)}, X^{(2)}, \cdots\}$, satisfying the Markov property

$$P(X^{(n+1)}|X^{(n)}, \cdots, X^{(0)}) = P(X^{(n+1)}|X^{(n)}) \tag{17.16}$$

Definition 17.11.2 A distribution over the states of a homogeneous[1] Markov chain is *invariant* (or *stationary*) with respect to transition probabilities T if

$$\pi = \pi T \tag{17.17}$$

A Markov chain can have more than one invariant distribution. If T is the identity matrix, for example, then any distribution is invariant.

We are interested in designing a Markov chain (its initial probabilities and transition matrix) for which the distribution we wish to sample from, given by π, is invariant. Hence, if we run the chain for sufficient time, it will converge to π, and then we can sample from it and compute the quantities desired.

Why do we need detailed balance?

Definition 17.11.3 Often, we will use *time reversible* homogeneous Markov chains that satisfy the more restrictive condition of *detailed balance*:

$$\pi(x)T(x, x') = \pi(x')T(x', x) \tag{17.18}$$

[1] The transition matrix is invariant of time.

Remark 17.11.1 One might be tempted to conclude that the detailed balance condition always holds, since

$$\pi(x)T(x,x') = P(x)P(x'|x) = P(x',x) \tag{17.19}$$
$$\pi(x')T(x',x) = P(x')P(x|x') = P(x,x') \tag{17.20}$$

and $P(x',x) = P(x,x')$.

The problem here is, x and x' denote different values, not different random variables. That is, in the argument of $\pi(\cdot)$ or in the first argument of $T(\cdot,\cdot)$, it is the value of the random variable X_0 (stochastic process at current time); and in the second argument of $T(\cdot,\cdot)$, it is the value of the random variable X_1 (stochastic process at the next time step). For example, we may have

$$\pi(X_0 = 1)T(X_0 = 1, X_1 = 2) = P(X_0 = 1)P(X_1 = 2|X_0 = 1) = P(X_1 = 2, X_0 = 1) \tag{17.21}$$

and

$$\pi(X_0 = 2)T(X_0 = 2, X_1 = 1) = P(X_0 = 2)P(X_1 = 1|X_0 = 2) = P(X_1 = 1, X_0 = 2) \tag{17.22}$$

which are usually not the same. Here $x = 1$ and $x' = 2$.

Note, the detailed balance condition implies π is an invariant distribution:

$$\sum_{i=1}^{n} \pi(x_i)T(x_i,x) = \sum_{i=1}^{n} \pi(x)T(x,x_i) = \pi(x)\sum_{i=1}^{n} T(x,x_i) = \pi(x) \tag{17.23}$$

It is possible for a distribution to be invariant without detailed balance holding. For example, the uniform distribution ($= 1/3$) on the state space $\{0,1,2\}$ is invariant with respect to the homogeneous Markov chain with transition probabilities $T(0,1) = T(1,2) = T(2,0) = 1$ and all others zero, but detailed balance does not hold.

Definition 17.11.4 A Markov chain is *ergodic* (or *irreducible*) if it is possible to go from every state to every state (not necessarily in one move).

Theorem 17.11.1 Let T be the transition matrix for a regular chain. Then as $n \to \infty$, the powers T^n approach a limiting matrix W with all rows the same vector $\mathbf{w}$. The vector $\mathbf{w}$ is a strictly positive probability vector (i.e., the components are all positive and they sum to one).

17.12 Metropolis-Hastings Algorithm

Suppose we can evaluate a distribution $P(X)$, but lack of information about how to directly draw samples from it. We wish to construct a Markov chain, utilizing the detailed balance condition (Equation 17.18), such that the induced invariant distribution is nothing but $P(X)$.

Assume we can take a new sample x' from $Q(x'|x)$, a known *proposal distribution* conditioned on the current state x of the Markov chain, and then we accept this new value x' based on a to-be-determined probability $A(x'|x)$. We define a Markov chain based on this procedure,

$$T(x, x') = Q(x'|x)A(x'|x) \tag{17.24}$$

If this Markov chain further satisfies the detailed balance condition (Equation 17.18),

$$P(x)Q(x'|x)A(x'|x) = P(x')Q(x|x')A(x|x') \tag{17.25}$$

then we can take

$$A(x'|x) = \min\left\{\frac{P(x')Q(x|x')}{P(x)Q(x'|x)}, 1\right\} \tag{17.26}$$

The ratio in the "min" function is known as the "Hastings ratio" (or acceptance ratio).

Now, running the chain for some sufficient time, we would obtain samples from the desired distribution $P(X)$, which is designed to be the same as the induced invariant distribution.

Remark 17.12.1 The $A(x'|x)$ defined above satisfies the detailed balance condition, since

$$P(x)Q(x'|x)A(x'|x) = P(x)Q(x'|x)\min\left\{\frac{P(x')Q(x|x')}{P(x)Q(x'|x)}, 1\right\} = \min\left\{P(x')Q(x|x'), P(x)Q(x'|x)\right\} \tag{17.27}$$

$$P(x')Q(x|x')A(x|x') = P(x')Q(x|x')\min\left\{\frac{P(x)Q(x'|x)}{P(x')Q(x|x')}, 1\right\} = \min\left\{P(x)Q(x'|x), P(x')Q(x|x')\right\} \tag{17.28}$$

and $\min\{P(x')Q(x|x'), P(x)Q(x'|x)\} = \min\{P(x)Q(x'|x), P(x')Q(x|x')\}$.

17.12.1 Metropolis Algorithm

In the Hastings ratio, if the proposal distribution is symmetric $Q(x|x') = Q(x'|x)$, such as a Gaussian distribution, then Equation 17.26 becomes

$$A(x'|x) = \min\left\{\frac{P(x')Q(x|x')}{P(x)Q(x'|x)}, 1\right\} = \min\left\{\frac{P(x')}{P(x)}, 1\right\} \tag{17.29}$$

this special case is called the "Metropolis Algorithm".

17.12.2 Gibbs Sampling

Suppose we wish to sample a random vector $\mathbf{X} = (X_1, \cdots, X_n)$, and its full conditional distribution $Q(X_i|\mathbf{X}_{-i})$ is known, where $\mathbf{X}_{-i} = (X_1, \cdots, X_{i-1}, X_{i+1}, X_n)$.

Then if we sample X_i component-wise from $Q(\mathbf{x}'|\mathbf{x}) = Q(x'_i|\mathbf{x}_{-i})$, and bearing in mind that $\mathbf{x}'_{-i} = \mathbf{x}_{-i}$ when sample x_i, the Hastings ratio becomes,

$$\frac{P(\mathbf{x}')Q(\mathbf{x}|\mathbf{x}')}{P(\mathbf{x})Q(\mathbf{x}'|\mathbf{x})} = \frac{P(x'_i|\mathbf{x}'_{-i})P(\mathbf{x}'_{-i})Q(x_i|\mathbf{x}'_{-i})}{P(x_i|\mathbf{x}_{-i})P(\mathbf{x}_{-i})Q(x'_i|\mathbf{x}_{-i})} \tag{17.30}$$

$$= \frac{P(x'_i|\mathbf{x}_{-i})P(\mathbf{x}_{-i})Q(x_i|\mathbf{x}_{-i})}{P(x_i|\mathbf{x}_{-i})P(\mathbf{x}_{-i})Q(x'_i|\mathbf{x}_{-i})} \tag{17.31}$$

$$= 1 \tag{17.32}$$

17.12.3 Collapsed Gibbs Sampling

17.12.4 Metropolis-Within-Gibbs

If not all full conditional probabilities are known or can be easily sampled from, we can sample those random variables with known conditional probabilities using the Gibbs algorithm, and others using Metropolis-Hastings algorithm. This is called *Metropolis-Within-Gibbs*.

17.13 Slice Sampling

17.13.1 Elliptical Slice Sampling

17.14 Split-Merge Sampling

17.15 Hamiltonian Monte Carlo

17.16 Data Fusion and Particle Filter (Sequential MCMC)

17.17 Reversible jump MCMC

17.18 Convergence Diagnostics

Chapter 18

Bayesian Nonparametrics

The concept of functions can be generalized to that of algorithms, which describe procedures, with loops and conditional tests, of how to generate output from input.

A draw from a finite dimensional Gaussian distribution is a real number, while a real-valued function can be considered as a sequence of (uncountably) infinite number of real numbers. A Gaussian Process (GP) is an infinite dimensional generalization of a Gaussian distribution. It defines a prior over real-valued functions, and a sample of it is a particular example of such functions.

A draw from a finite dimensional Dirichlet distribution is a (discrete) probability measure. A Dirichlet Process (DP) is an infinite dimensional generalization of a Dirichlet distribution. It defines a prior over probability measures, and a sample of it is a probability measure. Distributions drawn from a Dirichlet process are discrete, but cannot be described using a finite number of parameters, thus the classification as a nonparametric model.

Note, that we do not have a measurement of the function, as in the GP case but a sample of the true probability measure; this is the main difference between GP and DP.

18.1 Introduction

This note is based on Peter Orbanz's BNP notes:

http://stat.columbia.edu/~porbanz/npb-tutorial.html

18.2 Notation

Bold upper case letters represent matrices, e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{\Theta}$. Bold lower case letters represent vector-valued random variables and their realizations (we do not

distinguish between the two), e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\theta}$. Curly upper case letters represent spaces (i.e., possible values) of random variables, e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \Theta$.

18.3 Terminology

18.3.1 Parametric and nonparametric models

In a set of probability spaces $\{(\mathcal{Y}, \mathcal{F}, \mathcal{P}_\Theta)\}$, a *statistical model* $\mathcal{M}$ on a sample space $\mathcal{Y}$ is a set of probability measures $\mathcal{P}_\Theta$ on $\mathcal{Y}$. If we write $PM(\mathcal{Y})$ for the space of all probability measure on $\mathcal{Y}$, a model is a subset $\mathcal{M} \subset PM(\mathcal{Y})$. Every element of $\mathcal{M}$ has a one-to-one mapping (hence the model is *identifiable*) with its parameter $\boldsymbol{\theta}$ with values in a parameter space Θ, that is,

$$\mathcal{M}(\mathbf{y}) = \{P_{\boldsymbol{\theta}}(\mathbf{y}) | \boldsymbol{\theta} \in \Theta\}, \quad \mathbf{y} \in \mathcal{Y}. \tag{18.1}$$

For example, a first order polynomial is a model, and a second order polynomial is another model. We can of course fit a model to the observed data, but *model* itself is an abstract concept, where the parameter values of a model need not be specified. We call a model *parametric* if Θ has finite dimension, and *nonparametric* if Θ has infinite dimension.

To formulate statistical problems, we assume that n observations $\mathbf{y}_1, \ldots, \mathbf{y}_n$ with values in $\mathcal{Y}$ are observed, which are drawn i.i.d. from a measure $P_{\boldsymbol{\theta}}$ in the model, i.e.,

$$\mathbf{y}_1, \ldots, \mathbf{y}_n \sim_{iid} P_{\boldsymbol{\theta}} \qquad \text{for some } \boldsymbol{\theta} \in \Theta \tag{18.2}$$

The objective of statistical *inference* is then to draw conclusions about the value of $\boldsymbol{\theta}$ (and hence about the distribution $P_{\boldsymbol{\theta}}$ of the data) from the observations.

18.3.2 Bayesian and Bayesian nonparametric models

In Bayesian statistics, all parameters are considered as random variables. Hence under a Bayesian model, data are generated in two stages, i.e.,

$$\boldsymbol{\theta} \sim P(\boldsymbol{\theta}) \tag{18.3}$$

$$\mathbf{y}_1, \ldots, \mathbf{y}_n \mid \boldsymbol{\theta} \sim_{iid} P_{\boldsymbol{\theta}}(\mathbf{y}) \tag{18.4}$$

The objective is then to determine the *posterior distribution* – the conditional distribution of $\boldsymbol{\theta}$ given the observed data,

$$\pi(\boldsymbol{\theta} | \mathbf{y}_1, \ldots, \mathbf{y}_n) \tag{18.5}$$

A *Bayesian nonparametric* model is a Bayesian model whose parameter space Θ has infinite dimension. To define a Bayesian nonparametric model, we have to define a prior π on an infinite-dimensional space, which is a stochastic process with paths (i.e. realizations) in Θ.

18.4 Clustering and the Dirichlet process

18.4.1 Finite mixture models

The basic assumption of a clustering problem is that each observation $\mathbf{y}_i$ belongs to a single cluster $k \in \{1, \cdots, K\}$, which has a cluster distribution

$$P_k(\mathbf{y}_i | z_i = k) \tag{18.6}$$

where we have defined a latent variable z_i, indicating the cluster assignment of observation $\mathbf{y}_i$. Note that under the Bayesian framework, the latent variable z_i itself has a distribution

$$p_k^i \equiv P(z_i = k) \tag{18.7}$$

The marginal distribution of the observation $\mathbf{y}_i$ is then

$$P(\mathbf{y}_i) = \sum_{k=1}^{K} P(z_i = k) P_k(\mathbf{y}_i | z_i = k) \tag{18.8}$$

A model of this form is called a *finite mixture model*.

18.4.2 Bayesian mixture models

Suppose we know there are K clusters, we first sample the cluster parameters from some base measure:

$$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \sim_{iid} G(\boldsymbol{\beta}) \tag{18.9}$$

We then independently sample the latent cluster assignment vectors and the actual observations:

$$(p_1^i, \ldots, p_K^i) \sim \text{Dirichlet}_K(\alpha) \tag{18.10}$$

$$z_i \sim \text{Categorical}(p_1^i, \ldots, p_K^i) \tag{18.11}$$

$$\mathbf{y}_i \sim P_k(\mathbf{y}_i | \boldsymbol{\theta}_k, z_i = k) \tag{18.12}$$

18.4.3 Dirichlet Process

Definition 18.4.1 If $\alpha > 0$ and if G is a probability measure on Ω_ϕ, the random discrete probability measure Θ generated by

$$V_1, V_2, \cdots \sim_{iid} \text{Beta}(1, \alpha) \tag{18.13}$$

$$C_k = V_k \prod_{j=1}^{k-1} (1 - V_k) \tag{18.14}$$

$$\Phi_1, \Phi_2, \cdots \sim_{iid} G \tag{18.15}$$

is called a *Dirichlet process (DP)* with base measure G and concentration α, and denote its law by $\text{DP}(\alpha, G)$.

Appendix A

Glossary

http://alumni.media.mit.edu/~tpminka/statlearn/glossary/

- Model Evidence: Model evidence, or *marginal likelihood*, or *normalization constant*, is a likelihood function in which all parameter variables have been marginalized, usually denoted by $P(D|M)$, or $P(D)$ for short. For example, in polynomial regression, M may denotes the degree of the regression function, and parameter variables are the coefficients for any given M.

- Filtering is the task of tracking the posterior distribution of the latent state varialbe in state space models.

Appendix B

Q & A

Question B.0.1 Why is the difference between Statistics and Machine Learning?

Answer B.0.1 Statistics focuses on interpretability while Machine Learning cares more about predictability.

Question B.0.2 Why is the name "statistical" machine learning / data mining / pattern recognition?

Answer B.0.2 It means the data is in vector form and not in, for example, strings, where it will be called "syntactical/structural" pattern recognition.

Question B.0.3 How can I categorize machine learning models and methods?

Answer B.0.3 In general, machine learning models can be categorized according to their strategies for generating features: – fixed basis function models - basis functions are pre-designed; – adaptive basis function models (CART, Neural Networks) - form or parameters of basis functions learned from data; – kernel models (SVM, GP) - basis functions are implicitly defined, dimension of which is essentially infinite; – latent variable models / dimension reduction (HMMs, Sate Space Models; PCA, ICA).

Question B.0.4 How can I sample a random variable x marginally if I know how to sample jointly $p(x, y)$?

Answer B.0.4 Simply discard y and keep x in the joint samples.

Question B.0.5 What's the difference between /learning/ and /inference/.

Answer B.0.5 Learning is to fit parameters θ to a set of observations (x_i, y_i) while inference is to identify the input x for a particular observation y, using the learned parameters θ. EM can be thought of as iterating between learning and inference.

Question B.0.6 What is curse of dimensionality?

Answer B.0.6 The curse of dimensionality is that when the dimensionality increases, the volume of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality. Also, organizing and searching data often relies on detecting areas where objects form groups with similar properties; in high dimensional data, however, all objects appear to be sparse and dissimilar in many ways, which prevents common data organization strategies from being efficient.

Question B.0.7 Why over-fitted polynomial models (without regularization) tend to have larger coefficients?

Answer B.0.7 Because coefficients essentially are measures of "derivatives" of different orders, the larger the coefficients, the larger the derivatives, hence the more the function changes.

Question B.0.8 What is bias-variance trade-off, and its solution?

Answer B.0.8

$$E[(y-\hat{f})^2] = [Bias(\hat{f})]^2 + Var(\hat{f}) + \sigma^2 \tag{B.1}$$

where $Bias(\hat{f}) = E[\hat{f} - f]$ and $Var(\hat{f}) = E[(\hat{f} - E(\hat{f}))^2] = E[\hat{f}^2] - (E[\hat{f}])^2$.

One way of resolving the trade-off is to use mixture models and ensemble learning. For example, boosting combines many "weak" (high bias) models in an ensemble that has lower bias than the individual models, while bagging combines "strong" learners in a way that reduces their variance.

Question B.0.9 What are parametric models, non-parametric models, and etc.?

Answer B.0.9 A **parametric model** $\mathcal{M}$ is a collection of probability distributions $P_{\boldsymbol{\theta}}$, each of which is described by a *finite dimensional* (vector) parameter $\boldsymbol{\theta}$. A parametric model is called identifiable if the mapping $\boldsymbol{\theta} \to P_{\boldsymbol{\theta}}$ is invertible.

$$\mathcal{M} = \{P_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^k\} \tag{B.2}$$

A **non-parametric model** may refer to two interpretations: 1) it may refer to models that do not rely on data belonging to any particular distribution[1], but reply on comparative properties (statistics) of the data, or population, such as the "order statistics", or 2) it may refer to models that do not assume the structure of a model is fixed, i.e., the model grows in size to accommodate the complexity of the data. In these techniques, individual variables are typically assumed to belong to parametric distributions, and assumptions about the types of connections among variables are also made.

[1] Distribution-free methods are such examples, but they are not equivalent concepts.

Question B.0.10 What are generative models, discriminative models?

Answer B.0.10 For a supervised learning problem in which we wish to approximate an unknown target function $f : \mathcal{X} \to \mathcal{Y}$, or equivalently $P(Y|X)$, one approach is to model $P(Y|X)$ directly, which is called a discriminative model; another approach is to model $P(X, Y)$, or equivalently $P(Y)$ and $P(X|Y)$, and then use the Bayes' rule, to obtain $P(Y|X)$ for each $X = x$ query.

Question B.0.11 What are log-linear models?

Answer B.0.11 Such models have many names, including maximum-entropy models, exponential models, and Gibbs models; Markov random fields are structured log-linear models, conditional random fields are Markov Random Fields with a specific training criterion.

Question B.0.12 Bayesian v.s. Frequentism?

Answer B.0.12 There are two main schools of statistical inference: Frequentist and Bayesian[2]. The controversies arise when it comes to how to interpret the randomness of data point generating process.

Bayesians consider parameters to be *random*, and the observed data are conditioned on a realization of such random variables. Notice the natural hierarchical structure in this interpretation. The goal is to inference $p(\boldsymbol{\theta}|D_{\boldsymbol{\theta}_0})$, where $\boldsymbol{\theta}$ are the parameters and $D_{\boldsymbol{\theta}_0}$ the observed data set conditioned on realized values $\boldsymbol{\theta}_0$ of $\boldsymbol{\theta}$. Frequentists consider parameters to be *unknown but fixed*, and the observed data set is just a sample from the population. As in [12], Efron said "... Bayesian averages involve only the data value $\bar{x}$ actually seen, rather than a collection of theoretically possible other $\bar{x}$ values."

Also, as answered by Michael Hochster in [8] and [16]: Suppose h is the unknown constant, and H is the statistic computed from a sample. For Frequentists, it is valid to write $P(L \leq h \leq U) = 95\%$ or $P(70 \leq H \leq 74) = 95\%$, but not $P(70 \leq h \leq 74) = 95\%$ (this is 0 or 1). So the correct way to say is either "if the same experiment procedure is repeated 100 times, 95 times of the CIs will cover the unknown true value h", or "before the experiment, the probability is 95% that the CI to be obtained will cover h".

Wasserman said in [7] that the two schools of inference differ in their *goals*, not the *methods*: the goal of Frequentist inference is to construct procedures with frequency guarantees, and the goal of Bayesian inference is to quantify and manipulate degrees of beliefs.

Further Readings: Stein's example, Likelihood principle [9], [10], [14], [15].

Question B.0.13 What are Learning, Machine Learning, Data Mining, Statistics, and their differences?

Answer B.0.13 Learning is a process of improving performance with experience. There are two main types of learning: deductive learning and inductive learning. Deductive learning learns to apply generalization concepts (rules) to

[2]Another being Fiducial inference, or Fisherian inference.

examples; inductive learning learns to generalized concepts (rules) from examples.

Machine Learning is an example of inductive learning of machines (computers).

A big difference between Machine Learning and Data Mining is Reinforcement Learning. While one of the main goals of statistics is hypothesis testing, one of the main goals of data mining is the construction of hypotheses.

Question B.0.14 What is the difference between learning and inference?

Answer B.0.14 Inference reasons about unknown probability distributions; (parameter) learning is finding point estimates of quantities in the model. In Statistics, no distinction between learning and inference only inference (or estimation); and in Bayesian Statistics, all quantities are probability distributions, so there is only the problem of inference. Inference in the Machine Learning community also includes making predictions.

So your inference algorithm gives you posteriors in functional forms, and learning algorithm estimates parameter values from data, and you then inference about predictions using the fitted model.

Appendix C

Useful Resources

C.1 Data Sets

C.2 Packages and Source Codes

C.3 Important Papers

Bibliography

[1] James Stewart *Calsulus - Early Transcendentals*. Cengage Learning, 2012

[2] Walter Rudin *Principles of Mathematical Analysis*. McGraw-Hill Companies, Inc., 1976.

[3] H. L. Royden *Real Analysis*. Pearson Eduction, Inc., 1988.

[4] Erwin Kreyszig *Introductory Functional Analysis with Applications*. Wiley, 1989.

[5] Gerald B. Folland *Real Analysis: Modern Techniques and Their Applications*. Wiley, 1999.

[6] Alberto Torchinsky *Real Variables*. Westview Press, 1995.

[7] http://normaldeviate.wordpress.com/2012/11/17/what-is-bayesianfrequentist-inference/

[8] http://www.quora.com/What-is-the-difference-between-Bayesian-and-frequentist-statisticians

[9] http://www.bayesian-inference.com/advantagesbayesian

[10] http://www.bayesian-inference.com/likelihood#likelihoodprinciple

[11] Rossi P, Allenby G, McCulloch R. *Bayesian Statistics and Marketing (pp. 4)*. John Wiley & Sons, 2005.

[12] Efron, Bradley. *Controversies in the Foundations of Statistics*. The American Mathematical Monthly, Vol. 85, No. 4 (Apr., 1978), pp. 231-246.

[13] Efron, Bradley. *A 250-year Argument: Belief, Behavior, and the Bootstrap*. Bull. Amer. Math. Soc. 50 (2013), 129-146.

[14] http://www.quora.com/Statistics-academic-discipline/What-is-a-confidence-interval-in-laymans-terms

[15] http://www.quora.com/What-is-the-difference-between-Bayesian-and-frequentist-statisticians

[16] http://en.wikipedia.org/wiki/Confidence_interval#Meaning_and_interpretation

[17] Thomas P. Minka. *Old and New Matrix Algebra Useful for Statistics*. December 28, 2000.

[18] http://en.wikipedia.org/wiki/Matrix_calculus. Accessed on January 13, 2019

[19] S. R. Searle and H. V. Henderson. *A Primer on Differential Calculus for Vectors and Matrices*. BU-1047-MB, 1993.

[20] Steven W. Nydick. *A Different(ial) Way Matrix Derivatives Again*. May 17, 2012.

[21] Steven W. Nydick. *With(out) A Trace Matrix Derivatives the Easy Way*. May 16, 2012.

[22] Sam Roweis. *Matrix Identities*. June 1999.

[23] Terry Tao. *Matrix identities as derivatives of determinant identities*. January 13, 2013