Random Thoughts on Machine Learning

A Layman's Perspective

Yu Liu[1]

July 7, 2016

[1] https://github.com/ohliumliu/thoughts_learning.git

Contents

Preface

This is a collection of ongoing thoughts I gathered on my journey to better understand machine learning. I have been trying to relate with my past training in math and physics and these thoughts, naive as it may appear to professionals, might be helpful to get my feet into the door of this field.

1

Exponential Family: emergence of sigmoidal function

I was a little shocked when reading the Andrew Ng's lecture notes on the exponential family. Although some of the terminologies obvious were generalized from statistical mechanics, it still seems a little magical to see the power of exponential family. To me, this is a new way of thinking about a problem. This note is my attempt to understand the rationale behine the definition and application of exponential family. I rewrote a majority of the derivations in Ng's notes to make it look more systematic to me.

1.1 The problem

Suppose we attempt to study the relation between a feature x and an observable y. In regression, one has to start with some kind of hypothesis $y = h(x; \theta)$. In reality, of course, y is always a little different from the prediction of the hypothesis, or,

$$y_i = h(x_i; \theta) + \epsilon_i \tag{1.1}$$

for a particular observation (x_i, y_i). We would hope that $\epsilon_i \sim$ i.i.d $\mathcal{N}(0, \sigma^2)$. In fact, this is not always the case, and it is highly recommended to inspect the residual after performing a regression.

Anyway, a more fundamental issue is the origin of the hypothesis. It could come from the underlying physical model. If we only utilize the nature of statistics, what can we say about the hypothesis? For example, sigmodal function is used to do classification analysis, but why is sigmodal favored by us among all the possible smooth approximation to the step function.

In many cases, we do know, with certain confidence, the distribution of observable y. For example, due to central limit theoreom, it is probably safe to assume that a continous observable is Gaussian if it reflects the combined/summed effect of many underlying processes. Another example is classification; we do know the sample could be either positive or negative following Bernoulli distribution. Given this information, the objective of regression analysis is to find the relation between the parameters of the distribution and the underlying features x. So, what can we say about this relationship in light of the presumed distribution of y?

1.2 Exponential family

In the most general sense, y is a randome variable following some distribution $\mathcal{G}(y, \eta)$. Here, η represents the yet unknown contribution of features. The so called exponential family is a group of distribution that can be written in a way *such that η and y can be factored in the exponential*:

$$\begin{aligned} \mathcal{G}(y, \eta) &\propto b(y) e^{\eta y}, \qquad (1.2) \\ &= b(y) e^{\eta y - a(\eta)} \qquad (1.3) \end{aligned}$$

The second equation only adds a normalization term $e^{-a(\eta)}$ which is a function of η. $a(\eta)$ is called log partition function, because this idea originated from satistical physics. In particular, this is a generalization of Boltzman equation, if one relates y to energy, and η to $\beta \equiv 1/k_B T$. From a pratical point of view, the whole idea of partition function and Boltzman distribution is that moments of energy or y can be calculated as the derivative of log partition function with respect to β or η. In essense, β or η is the Lagrangian multiplier. In order to sastisfy this, the only coupling term in $\mathcal{G}(y, \eta)$ should be $e^{\eta y}$.

As an example, let's look at $\langle y \rangle$.

$$\begin{aligned}
\langle y \rangle &= \int y\mathcal{G}(y,\eta)\, dy && (1.4)\\
&= \int b(y)e^{\eta y - a(\eta)} y\, dy && (1.5)\\
&= \int b(y)\frac{\partial}{\partial \eta}\left[e^{\eta y}e^{-a(\eta)}\right] dy + \int b(y)e^{\eta y - a(\eta)}\frac{\partial a(\eta)}{\partial \eta}\, dy && (1.6)\\
&= (\frac{\partial}{\partial \eta} + \frac{\partial a(\eta)}{\partial \eta})\int \mathcal{G}(y,\eta)\, dy && (1.7)\\
&= \frac{\partial a(\eta)}{\partial \eta}. && (1.8)
\end{aligned}$$

The exponential family is a generalization of Boltzman distriubtion because of the extra factor $b(y)$. This generalization doesn't affect the above property regarding partition function and expectation value.

Remember our motivation is to find out constraints on η using only mathematical facts. We already have one. The derivative of log partition function is the expectation of y. That means, whatever model we choose to predict the distribution of y, if the distribution of y is a member of the exponential family, then $\langle y \rangle = da(\eta)/d\eta$. The hypothesis $h(x, \theta)$ of course predicts the expectation of y (in the case of least square error, not other metrics), and it thus follows

$$h(x, \theta) = \frac{d}{d\eta}a(\eta). \qquad (1.9)$$

To reiterate, Eq. 1.9 tells us, if the distribution of y belongs to the exponential family, our hypothesis must assume a form arising from the derivative of the log partition function. In another word, there could be various ways to get η from x and θ, but the functional form of the hypothesis $h(x, \theta)$ is fixed if it's written in terms of η. Remember that η is the proxy for our hypothesis regarding x and θ.

Eq. 1.9 only requires that the hypothesis agrees with the expected value of y. Being such a general requirement, what information can we extract from it?

1.3 Gaussian

Let's start with Gaussian and apply the idea of Eq. 1.9. First, we try to rewrite a Gaussian distribution according to Eq. 1.3.

$$\begin{aligned} \mathcal{N}(\mu, 1) &= \frac{1}{\sqrt{2\pi}} \exp\{-(y-\mu)^2/2\}, & (1.10) \\ &= \frac{1}{\sqrt{2\pi}} e^{-y^2/2} e^{\mu y} e^{-\mu^2/2}. & (1.11) \end{aligned}$$

Obviously, Gassusian distribution $\mathcal{N}(\mu, 1)$ is a member of the exponential family by setting

$$\begin{aligned} b(y) &= \frac{1}{\sqrt{2\pi}} e^{-y^2/2}, & (1.12) \\ \eta &= \mu, & (1.13) \\ a(\eta) &= \mu^2 = \eta^2 & (1.14) \end{aligned}$$

Using Eq. 1.9, our hypothesis shoud assume

$$\begin{aligned} h(x, \theta) &= \frac{d}{d\eta}(\eta^2/2) & (1.15) \\ &= \eta & (1.16) \\ &= \mu & (1.17) \end{aligned}$$

To recap what has happened this section, we started with an assumption of the distribution of an observable $y \sim \mathcal{N}(\mu, 1)$, and then confirmed that it belongs to the exponential family by simply rearrange some terms. Then following Eq. 1.9, we found that the hypothesis must assume $h(x, \theta) = \eta$ when written in terms of η. In particular for Gassian, η is equivalent to the mean of the distribution, so what we learn from this exercises is that the hypothesis much agrees with the average of y. We are still free to choose whatever $h(x, \theta)$ that is theoretically possible. Apparently, not much was gained from this exercises. But it's not the case for other distributions (See Section. 1.4.

Before we move on to show the power of this analysis, let's consider a slightly more general case when we starts from $\mathcal{N}(\mu, \sigma^2)$ with σ known. Of course, this can be reduced to the earlier case by scaling y with σ. More formally, we would relax the requirement of exponential family such that the coupling term is $e^{\eta T(y)}$. Earlier, $T(y) = y$. Following the same procedure that led to Eq. 1.8, we have

$$\langle T(y) \rangle = \frac{\partial}{\partial} a(\eta). \qquad (1.18)$$

So our hypothesis must satisfy the functional form arising from this equation. In terms of $\mathcal{N}(\mu, \sigma)$, it turns out $T(y) = y/\sigma$, $\eta = \mu/\sigma$ and $a(\eta) = \eta^2/2$.

Since our hypothesis $h(x,\theta)$ should be the same as $\langle y\rangle$, it follows $\langle T(y)\rangle = h(x,\theta)/\sigma = \eta$ and we recover the same thing as before.

A further generalization is to assume $\mathcal{N}(\mu,\sigma)$ with both μ and σ unknown. The definition of exponential family can be further extended to have as the coupling term $e^{\vec{\eta}\cdot\vec{T}(y)}$. Here, the observable $T(y)$ and the parameter η become vectors. And Eq. 1.8 becomes

$$\langle T(y)_i\rangle = -\frac{\partial}{\partial\eta_i}a(\vec{\eta}) \tag{1.19}$$

As for Gassuian, $\vec{\eta} = (\mu/\sigma^2, -1/2\sigma^2)^T$ and $a(\vec{\eta}) = -\eta_1^2/4\eta_2 + (1/2)\ln|1/2\eta_2|$. Following Eq. 1.19, the hypothesis should ensure that $\langle y\rangle = \mu$ and $\langle y^2\rangle = \sigma^2 + \mu^2$. Again, not very informativel. But it explains why we would want $y = h(x,\theta) + \mathcal{N}(\mu,\sigma)$.

1.4 Bernoulli and logistic

The Bernoulli distribution is a natural choice to describe the probablity of a binary outcome y. Here, we believe that y assumes either 1 or 0 with the following probability:

$$\text{Prob}(y) = \begin{cases}\phi, & y = 1\\ 1-\phi, & y = 0\end{cases} \tag{1.20}$$

Apparently, we don't know ϕ and how to estimate it from our features, but as before, we wonder if something could be deduced realizing that Bernoulli distribution belongs to the exponential family.

First of all, we rewrite the probability mass function in terms of Eq. 1.3.

$$\begin{aligned}\text{Prob}(y) &= \phi^y(1-\phi)^{1-y}, && (1.21)\\ &= \exp\{y\ln\phi + (1-y)\ln(1-\phi)\} && (1.22)\\ &= e^{y\ln\left(\frac{\phi}{1-\phi}\right)}e^{\ln(1-\phi)}. && (1.23)\end{aligned}$$

Indeed, Bernoulli distribution can be cast into a form conforming with the definition of exponential family with

$$\begin{aligned}b(y) &= 1, && (1.24)\\ \eta &= \ln\frac{\phi}{1-\phi}, && (1.25)\\ a(\eta) &= -\ln(1-\phi) = \ln\left(1+e^{\eta}\right). && (1.26)\end{aligned}$$

Following the idea used in the previous section, the expectation of y is given by Eq. 1.8, or,

$$\phi = \langle y\rangle = \frac{e^{\eta}}{1+e^{\eta}}. \tag{1.27}$$

Here we used the fact that the average of y is ϕ since our starting point is $y \sim$ Bernoulli(ϕ). This equation tells us that the parameter ϕ has to be calcuated from a sigmoidal function of η. There are different hypothesis of η, for example, $\eta = \vec{\theta} \cdot \vec{x}$, but Eq. 1.27 has to be used to get ϕ from η. This is in contrast with Gaussian. In that case, η turns out to be μ, and there is no functional constraints between the two.

Eq. 1.27 is very powerful. Among all the smooth approximations of step function, this function is chosen by merely invoking some fundamental properties of statistical distribution and inference, namely, (i) the distribution of the observalbe belongs to the exponential family, and (ii) the hypothesis should agreen with the expectation of the observable. Suppose we choose another function to replace Eq. 1.27, we wouldn't be able to justify it purely based on the the two properties we just used to reach Eq. 1.27.

It may seem a little magic at the first sight, but it actually makes sens. Eq. 1.27 assumes a fractional form, namely $\phi = \mathcal{P}/1 + \mathcal{P}$ where $\mathcal{P}$ is some unknown function of the features. Being a Bernoulli process, the probability of an outcome should be determined by its associated probability as a fraction of the sum of all possible outcomes' probabilities. This is also the key idea behind Bayesian. From this point of view, Eq. 1.27 is not that strong a conclusion.

1.5 Multinomial

As another example, let's consider a multinomial distribution problem, where we want to predict the outcome of a multi-choice classification experiment. As usual, we know that the probability of observing a particular outcome $i \in [1, \cdots, n]$ is ϕ_i and the purpose is to find out the structure of ϕ_i. We write the probability in terms of a vector $\vec{y}$ with $y_j = \delta_{ji}$ if the outcome is j,

$$\begin{aligned} \mathrm{Prob}(\vec{y}) &= \phi_1^{y_1} \cdots \phi_n^{y_n}, && (1.28)\\ &= \phi_1^{y_1} \cdots \phi_{n-1}^{y_{n-1}} \phi_n^{y_n}, && (1.29)\\ &= \phi_1^{y_1} \cdots \phi_{n-1}^{y_{n-1}} (1 - \sum_{i<n} \phi_i)^{y_n}, && (1.30) \end{aligned}$$

where we used the fact that ϕ_i's sum to unity. But if we would like to follow the idea that lead to Eq. 1.26, we should further write y_n in terms of y_i with $i < n$ and try to convert the expression to Eq. 1.3:

$$\begin{aligned} \mathrm{Prob}(\vec{y}) &= \phi_1^{y_1} \cdots \phi_{n-1}^{y_{n-1}} (1 - \sum_{i<n} \phi_i)^{1-\sum_{i<n} y_i}, && (1.31)\\ &= e^{y_1 \ln \phi_1} \ldots e^{y_{n-1} \ln \phi_{n-1}} e^{(1-\sum_{i<n} y_i) \ln(1-\sum_{i<n} \phi_i)}, && (1.32)\\ &= e^{y_1 \ln(\phi_1/\phi_n)} \ldots e^{y_{n-1} \ln(\phi_{n-1}/\phi_n)} e^{\ln \phi_n}. && (1.33) \end{aligned}$$

Here, we use ϕ_n to denote $1 - \sum_{i<n} \phi_i$ just for convenience. Apparently, we can recover the exponential family distribution by setting

$$\begin{aligned}
b(\vec{y}) &= 1, && (1.34)\\
\eta_i &= \ln(\phi_i/\phi_n) = \ln \frac{\phi_i}{1 - \sum_{i<n} \phi_i}, && (1.35)\\
a(\eta) &= \ln(1/\phi_n) = \ln \frac{1}{1 - \sum_{i<n} \phi_i}. && (1.36)
\end{aligned}$$

In order to use Eq. 1.19, we need to write $a(\eta)$ in terms of η or use the chain rule. Let's try to do the former. Since $\eta_i = \ln(\phi_i/\phi_n)$, we have $\phi_i = \phi_n e_i^\eta$ and

$$\begin{aligned}
\sum_{i<n} \phi_i &= \sum_{i<n} \phi_n e^{\eta_i}, && (1.37)\\
&= \phi_n \sum_{i<n} e^{\eta_i}, && (1.38)\\
&= \left(1 - \sum_{i<n} \phi_i\right) \sum_{i<n} e^{\eta_i}, && (1.39)\\
\sum_{i<n} \phi_i &= \frac{\sum_{i<n} e^{\eta_i}}{1 + \sum_{i<n} e^{\eta_i}}, && (1.40)\\
a(\eta) &= \ln\left(1 + \sum_{i<n} e^{\eta_i}\right) && (1.41)
\end{aligned}$$

By using Eq. 1.19, one has the structure of ϕ_i in terms of η_i,

$$\begin{aligned}
\phi_i &= \langle y_i \rangle && (1.42)\\
&= \frac{\partial a(\vec{\eta})}{\partial \eta_i}, && (1.43)\\
&= \frac{e^{\eta_i}}{1 + \sum_{i<n} e^{\eta_i}} && (1.44)
\end{aligned}$$

Here, $i \in [1, \cdots, n-1]$, and $\phi_n = 1/(1 + \sum_{i<n} e^{\eta_i})$. Given the discussion at the end of the previous section, this result is not that surprising.

2

Gradient descent: a first-order approximation

2.1 A "derivation" of the gradient descent algorithm

Given a cost function $J(\theta)$ and we want to minimize it by changing θ. Of course, the straightforward evaluation of derivative may provide a closed-form solution, but it doesn't scale very well in most practical cases. An "engineering" approach, gradient descent, is often used. It is essentially an iterative method to search for local minimum step by step.

Given an initial guess θ_0, we would like to change it by $d\theta$ to make $J(\theta)$ smaller. In the most general term,

$$d\theta = \mathcal{F}(J(\theta_0), J'(\theta_0), \cdots). \tag{2.1}$$

If all the derivatives were known, it would be very likely that we know $J(\theta)$ by Talyor series, and the minimization problem would be solved. In reality, we have to settle with much less information.

It makes sense to assume that $d\theta = 0$ if $J'(\theta_0) = 0$, because θ_0 reaches extremum. It is thus a first order approximation to prescribe

$$d\theta = kJ'(\theta_0). \tag{2.2}$$

We also would like $J(\theta_0 + d\theta) \leq J(\theta_0)$, which translates to

$$d\theta J'(\theta_0) \leq 0. \tag{2.3}$$

Eq. 2.2 and Eq. 2.3 together implies $k \leq 0$. In a more common format, the gradient descent iteration reads

$$\theta_j = \theta_i - \alpha J'(\theta_i), \tag{2.4}$$

where $\alpha > 0$ is called learning rate. This is one of the places people try to use "learning" to describe the progression of a fitting procedure.

The choice of α is up to the user and has many variant. A constant learning rate may or may not be good enough to converge. An interesting feature of Eq. 2.4 is that, given a constant learning rate, the change in θ is large when $J(\theta)$ is steep, and small when $J(\theta)$ is small. This seems to imply a danger of overshooting in the first case, and slow convergence in the latter.

2.2 The superficial relation with Newton's method

Newton's method is often used to find the root of a function $f(\theta)$. In particular, the iteration process is

$$\theta_j = \theta_i - \frac{f(\theta_i)}{f'(\theta_i)}. \tag{2.5}$$

Realizing that minimizing $J(\theta)$ is equivalent to solving $J'(\theta) = 0$ (to some extent), Eq. 2.5 suggests the following iteration

$$\theta_j = \theta_i - \frac{J'(\theta_i)}{J''(\theta_i)}. \tag{2.6}$$

Comparing Eq. 2.4 and Eq. 2.6, one finds that Newton's method is equivalent to gradient descent with a learning rate determined by the curvature,

$$\alpha = \frac{1}{J''(\theta)}. \tag{2.7}$$

This makes sense because when the curvature is big, a smaller time step seems fit. This relation also reveals that α and $J''(\theta)$ has the same sign. That means, when $J'(\theta) > 0$ and $J''(\theta) < 0$, $d\theta < 0$, meaning that the algorithm may converge to local maximum. In another word, the requirement that $\alpha > 0$ may not hold. This is a very serious issue. Of course, we could use $|J''(\theta)|$ as a band-aid.

Another way to come up with Eq. 2.6 is to approximate $J(\theta)$ around θ_i to the second order,

$$J(\theta) = J(\theta_i) + J'(\theta_i)(\theta - \theta_i) + \frac{1}{2}J''(\theta_i)(\theta - \theta_i)^2, \tag{2.8}$$

and it follows that

$$\theta_j = \theta_i - \frac{J'(\theta_i)}{J''(\theta_i)} \tag{2.9}$$

is the best estimate of θ given local information at θ_i. In high dimension, a Hessian matrix is used instead of J''.

An alternative to relate with Newton's method is to realize that solving $f(\theta) = 0$ actually minimizes $J(\theta) = f^2(\theta)$. According to Newton's formulae,

$$\begin{aligned}
\theta_j &= \theta_i - \frac{f(\theta_i)}{f'(\theta_i)}, & (2.10)\\
&= \theta_i - \frac{2f(\theta_i)f'(\theta_i)}{2[f'(\theta_i)]^2}, & (2.11)\\
&= \theta_i - \alpha J'(\theta_i), & (2.12)
\end{aligned}$$

where $\alpha = \frac{1}{2[f'(\theta_i)]^2}$. It might seem that this correlation makes more sense because $\alpha \leq 0$ and α is bigger when $f(\theta)$ or $J^{1/2}(\theta)$ is steeper. In fact, if we write everything in terms of $J(\theta)$, the iteration becomes

$$\theta_j = \theta_i - \frac{2J(\theta_i)}{J'(\theta_i)}. \tag{2.13}$$

This is not reasonable because it diverges when $J'(\theta) = 0$. The fundamental flaw is that $f(\theta) = 0$ is sufficient but not a necessary condition to minimize $J(\theta)$.

Yet anothe alternative is to solve $f'(\theta) = 0$ using Newton's method to minimize $J(\theta) = f^2(\theta)$. Here and in the above, when $J = f^2$, it is non-negative, normally satisfied by the use of sum-of-square cost functions. It is almost impossible to have $f(\theta) = 0$ which means there is zero error. Applying Newton's idea again, the updating rule becomes ($c$heck the math!)

$$\begin{aligned} \theta_j &= \theta_i - \frac{f'(\theta_i)}{f''(\theta_i)}, && (2.14) \\ &= \theta_i - \frac{J'(\theta_i)}{J''(\theta_i) - \frac{[J'(\theta)]^2}{2J(\theta)}} && (2.15) \end{aligned}$$

In summary, this exercise attempted to relate Newtwon's method and gradient descent. It looks like little insight was gain after all. To improve gradient descent, conjugate gradient was developed. I will read up on that one.

3

Linear regression: some noteworthy results

3.1 Motivation

This is coolection of notes I am gathering regarding linear regression. The following is my focus:

- Useful math tools.
- Useful interpretations, especially geometrical ones.
- Useful practical considerations.

A lot of the results are from "Elements of Statistical Learninig".

3.2 Problem definition

We have n experiments characterized by $\mathbf{x_i}$ with respective to p features. For each measurement, there is single output y_i. The task is to find the best linear model characterized by its parameters β_i where $i = 1, \cdots, p$.

From the perspective of maximum likelyhood, we assume that the observed value is the prediction by the model plus a random noise,

$$y_i = \beta_0 + x_{ij}\beta_j + \epsilon_i, \tag{3.1}$$

where the last terms is the noise term and Einstein notation is assumed. Here, x_{ij} is the value for feature j in the i-th measurement. One can write down the joint probability of observing a particular set of y_i given a set of $\mathbf{x}_i$. Note that the only source of randomness is from ϵ. Under the following conditions (sufficient but may not be necessary):

- Each measurement is independent
- $\epsilon_i \sim \text{i.i.d}\mathcal{N}(0, \sigma^2)$

the joint probability is maximized by minimizing the following residual sum of square,

$$RSS = \sum_{i=1}^{n} \left[y_i - (\beta_0 + x_{ij}\beta_j)\right]^2. \tag{3.2}$$

There are ways to simplify Eq. 3.2. A common practice is to arrange $\mathbf{x_i}$ in a $n \times (p+1)$ matrix X whose columns are $\mathbf{x_i}$'s and the first column being 1's. In this notation, the prediction is simply $X\beta$ and

$$\begin{aligned} RSS &= ||Y - X \cdot \beta||^2, & (3.3)\\ &= (Y - X \cdot \beta)^T (Y - X \cdot \beta), & (3.4)\\ &= (Y - X \cdot \beta) \cdot (Y - X \cdot \beta). & (3.5) \end{aligned}$$

Here, Y is a column vector formed by y_i $(i = 1, \cdots, n)$ and β is a column vector formed by β_i $(i = 0, \cdots, p)$. The second to last line is in terms of matrix and the last one is in terms of vector/tensor inner product.

3.3 Estimator and interpretations

3.3.1 Geometric interpretation in sample space

Eq. 3.2 can be understood in terms of the linear superposition of $p+1$ vectors in $\mathbb{R}^n$. Recall that the design matrix X is $n \times (p+1)$, and there are $p+1$ column vectors each with n components. $X \cdot \beta$ is nothing but a compact way to generate a linear superposition of these $p+1$ vectors with coefficients β_i. Our purpose is to find the coefficients so that the linear superposition is closest to our target vector Y.

When $p+1 = n$, β can be solved by solving a linear equation $Y = X\beta$. When $p+1 > n$, there could be multiple solutions, leading to overfitting. When $p+1 < n$, there is no guarantee that we can have a solution to $Y = X\beta$. This is the case in most practical cases of linear regression, where we have fewer parameters than the number of points.

Focusing on the last case, what would be the best choice of *beta* to recover the target vector Y from a linear combination of $\mathbf{x}_i$'s. Apparently, it boils down to the definition of closeness in $\mathbb{R}^n$, and the adoption of Eq. 3.2 implies the use of L_2 norm (Euclidean distance).

3.3.2 A quick derivation using Einstein notation

I will briefly write down a derivation of the estimator of β. I am not going to reference results written in matrix operation. Instead, I will use Einstein notation before converting the final results to matrix notation. This is the approach I learned from fluid mechanics class. In fact, the neat equations in terms of matrix operation is proved by using Einstein notation anyway.

Eq. 3.2 is rewritten as a function of β.

$$RSS(\beta) = \sum_{i=1}^{n} \left[y_i - x_{ij}\beta_j \right]^2 , \tag{3.6}$$

where j runs from 0 to p. For a particular k, the stationary condition is

$$\begin{aligned} \frac{\partial}{\partial \beta_k} RSS(\beta) &= 0, & (3.7)\\ \sum_{i=1}^{n} \left(y_i - x_{ij}\beta_j \right) x_{ik} &= 0, & (3.8)\\ x_{ik}x_{ij}\beta_j &= x_{ik}y_i, & (3.9)\\ (X^T X\beta)_k &= (X^T y)_k. & (3.10) \end{aligned}$$

The last line was reached by noting that $X_{ij} = x_{ij}$ by our construction. Since Eq. 3.10 applies to k different components, it is equivalent to

$$X^T X\beta = X^T y, \tag{3.11}$$

which has an obvious solution

$$\beta = (X^T X)^{-1} X^T y. \tag{3.12}$$

It is useful to check that the dimension of the right hand side of Eq. 3.12 is $(p+1) \times 1$ as expected.

Eq. 3.12 also provides the starting point to estimate the covariance matrix of β_i. Note that β is a linear transformation of Y, so $COV(\beta)_{ij}$ can be related to $COV(Y)$ by the following,

$$\begin{aligned} COV(\beta)_{ij} &= E[(\beta_i - \bar{\beta}_i)(\beta_j - \bar{\beta}_j)], & (3.13)\\ &= E[(H(Y - \bar{Y}))_i (H(Y - \bar{Y}))_j], & (3.14)\\ &= E[H_{il}(Y - \bar{Y})_l H_{jm}(Y - \bar{Y})_m], & (3.15)\\ &= H_{il} H_{jm} COV(Y)_{lm}, & (3.16)\\ &= H_{il} COV(Y)_{lm} H^T_{mj}. & (3.17) \end{aligned}$$

Since this applies to each elements of $COV(\beta)$, we have in general

$$COV(\beta) = H\ COV(Y)\ H^T. \tag{3.18}$$

In particular, $H = (X^T X) X^T$ based on Eq. 3.12. If $COV(Y) = \sigma^2 I$, the covariance matrix of β is simply

$$COV(\beta) = (X^T X)^{-1} \sigma^2. \tag{3.19}$$

The derivation above can be easily extended to the case of regularization. For this purpose, we need to limit ourselves to the discussion of the slopes β_i $(i = 1, \cdots, p)$ as in Chapter 3. Just to reiterate the rationale, the intercept shouldn't be subjected to regularization because that is a constant shift in the model; a constant shift to all the data points leads to a change in β_0 only and shouldn't introduce more or less penalty due to regularization. The target function we are trying to minimize becomes,

$$\mathcal{F}(\beta) = RSS(\beta) + \lambda \beta_i \beta_i. \tag{3.20}$$

The stationary condition for β_k becomes

$$\begin{aligned} \frac{\partial}{\partial \beta_k} \mathcal{F}(\beta) &= 0, & (3.21)\\ \sum_{i=1}^{n} 2\,(y_i - x_{ij}\beta_j)\,(-x_{ik}) + 2\lambda\beta_k &= 0, & (3.22)\\ (x_{ik}x_{ij} + \delta_{kj}\lambda)\beta_j &= x_{ik} y_i, & (3.23)\\ ((X^T X + \lambda I)\beta)_k &= (X^T y)_k. & (3.24) \end{aligned}$$

The last line is again written in matrix notation

$$(X^T X + \lambda I)\beta = X^T Y, \tag{3.25}$$

which solves to

$$\beta = (X^T X + \lambda I)^{-1} X^T Y. \tag{3.26}$$

Eq. 3.26 is related to the original motivation of regularization. In order to guarantee that $X^T X$ is not singular, one could add a term λI to make it non-singular. This may also make the matrix inversion more stable numerically. Of course, regularization has alternative, and perhaps more fundamental motivations, which are summarized below:

- Make $X^T X$ non-signular as discussed just now.
- Extra data points as discussed in Chapter 3. This idea can be used in the derivation above to Eq. 3.26 as well.
- As suggested in Chapter 3 and more formally in Eq. 3.20, regularization can be thought of as the reminant of a prior probability. In fact, Eq. 3.20 is the log-likelyhood of having β given the observation Y and a prior of $\beta_i \sim \mathcal{N}(0, \lambda^{-1})$ (after ignoring the denominator).

3.3.3 Geometric interpretation

As discussed earlier, we tried to minimize the difference, in $\mathbb{R}^n$, between our target vector Y, and a linear combination of $\mathbf{x}_i$ or $X\beta$. In this so called column space of X, we are trying to minimize the Euclidean distance between Y and $X\beta$. The ideal case is that $d \equiv Y - X\beta = 0$ has a solution. This is equivalent to requiring $d_i = 0$ or $d \cdot \hat{e}_i = 0$ for $i = 1, \cdots, n$ with $\hat{e}_i$'s a complete set of basis. Of course, for linear regression, this is not possible, then what compromise should be adopt? Since our only knowledge is from the observation $\mathbf{x}_i$'s, it thus makes sense to use x_i to replace $\hat{e}_i$. Note that $\mathbf{x}_i$ is the i-th column of X by definition, and thus $(\mathbf{x}_i)_j = (X)_{ji} \equiv x_{ji}$. For any $k = 0, \cdots, p$,

$$\mathbf{x_k} \cdot (Y - X\beta) = 0, \tag{3.27}$$

$$\sum_{i=1}^{n} x_{ik}(y_i - x_{il}\beta_l) = 0. \tag{3.28}$$

This was obtained earlier.

As an aside, one more word on the column space of X. X can be thought of as a row vector whose components are the column vectors $\mathbf{x}_i$ ($i = 0, \cdots, p$). $X\beta$ can thus be thought of as the inner product of the "row vector" X with a column vector β. The result is a linear combination of the elements of X. The result should have the same dimension as the elements of X, or a column vector $n \times 1$. This line of thought comes in handy when dealing with higher dimensions, especially common in tensor algebra.

Now going back to the geometric interpretation. Because we are actually solving β in the compromised condition Eq. 3.27, the prediction $\hat{Y} \equiv X\beta$ for the training points must be in the subspace spanned by the column vectors $\mathbf{x}_i$ $(i = 0, \cdots, p)$. In particular, Eq. 3.27 implies that $Y - \hat{Y}$ is the norm of the column space, so it is perpendicular to any linear combination of the basis of this column space, including $\hat{Y}$.

With this interpretation in mind, it is thus easy to understand what would happen when some features are correlated. Mathematically, it means some of the column vectors are parallel. In the extreme case, it would make the matrix $X^T X$ singular and non-invertible. Geometrically, it would make it more difficult to uniquely determine the norm to the column space.

3.3.4 Gram-Schmidt

4

Regularization: the “hidden" facts?

4.1 Motivation of regularization

It is well known that introducing too many features in a model will almost always lead to perfect score with the training and validation set, but miserable failure with the test set. In another word, a feature-rich model has a very small bias (toward a particular model), but a very high variance (when used for prediction). The high variance probably comes from the observation that the confidence region of the fitted parameters is big enough to overwhelm the average value.

Another scnenario is from the so-called wide data. It's conventional to arrang the data so that each row represents an observation and each column represents a feature. Wide data simply indicates that the number of features is bigger than the number of samples. This is often found in imaging analysis (each pixel is a feature), array data and etc.

If there is no way or rationale to reduce the number of features, regularization (as well as lasso) is often applied to solve the problem of over-fitting. I have a feeling that this amounts to the introduction of extra data points to the modeling problem, and this chapter is a summary of my attempt to formulate this feeling.

Before my anaylsis, it should be noted that, so far I have seen two types of retionale for regularization from textbooks. The first is from the observation that large fitting parameters with opposite signs may provide good fitting results if the corresponding features pairs happen to correlate, as is common the case in multi-variate models. Of course, one would like to limit the unwanted growth of slopes, thus the introduction of a penalty term proportional to sum of square of the parameters. The second is based on the practical concern that the design matrix (the matrix formed by the column vectors of observations) may be close to singular, making it unsable to invert, as needed for the estimation of slopes. Numerically, one would add some non-zero diagonal term to the design matrix to make sure it is not signular, and this is equivalent to adding a penalty term in the cost function that is proportional to sum of square of the parameters.

4.2 Linear model

I will start with the simple case of a a linear model with a feature space $\vec{x} \equiv (x_1, \cdots, x_n)^T$ and the corresponding set of coefficient vector $\vec{\theta} \equiv (\theta_1, \cdots, \theta_n)^T$. The prediction of the model is simply linear with a constant offset θ_0:

$$h(\vec{x}; \vec{\theta}) = \theta_0 + \vec{\theta} \cdot \vec{x} \tag{4.1}$$

Given m data points $(\vec{x}^{(m)}, y^{(m)})$ with $\vec{x}^{(m)} = (\vec{x}_1^{(m)}, \cdots, \vec{x}_n^{(m)})^T$, the cost function without regularization reads

$$J(\vec{\theta}) = \sum_{i=1}^{m} \left[y^{(i)} - h(\vec{x}^{(i)}; \vec{\theta}) \right]^2. \tag{4.2}$$

With regularization, an extra term is added to penalize non-zero values of θ_i,

$$J(\vec{\theta}) = \sum_{i=1}^{m} \left[y^{(i)} - h(\vec{x}^{(i)}; \vec{\theta}) \right]^2 + \lambda \sum_{i=0}^{n} \theta_i^2, \tag{4.3}$$

As an aside, we can vectorize the cost function by first defining an error vector $(m \times 1)$,

$$\vec{\epsilon} = \vec{y} - \left[\vec{\theta_0} + \vec{\theta}^T \cdot \mathbf{X} \right], \tag{4.4}$$

where $\vec{\theta_0} \equiv \theta_0 \times (1, \cdots, 1)$ and $\mathbf{X}$ is an $n \times m$ matrix which each column as $\vec{x}^{(m)}$. In this way, the cost function is

$$J(\vec{\theta}) = \vec{\epsilon} \cdot \vec{\epsilon} + \lambda(\theta_0^2 + \vec{\theta} \cdot \vec{\theta}). \tag{4.5}$$

It's also worth noting that the penalty term could take other forms. For example, in Lasso regression, the L_1 norm (absolute values) of θ_i is used.

Going back to the regularizaton process, the factor λ is a tunable parameter to control the biasness of the regression. I will try to show that, in this context of simple linear regression, the introduction of regularization term seems to be equivalent to the “artificial" addition of extra data points.

In order to demonstrate this point, we need to find n data points that can lead to the extra penalty term $\lambda(\theta_0^2 + \vec{\theta} \cdot \vec{\theta})$.

For example, we want to find $(\vec{u}^{(1)}, v^{(1)})$ that satisfies

$$\left[v^{(1)} - \left(\theta_0 + \vec{\theta} \cdot \vec{v}^{(1)} \right) \right]^2 = \lambda \theta_1^2. \tag{4.6}$$

An obvious choice that is independent θ_i is $\vec{u}^{(1)} = (\sqrt{\lambda}, 0, \cdots)$ and $v^{(1)} = \theta_0$. In general, we would like to have $\vec{u}^{(i)} = \{\sqrt{\lambda}\delta_{ij}\}_j (j = 1, \cdots, n)$ and $v^{(i)} = \theta_0$[1]. We need to have n points.

With this choice of points, the regularization cost can be compared with adding some extra points to the training set. In the space spanned by the features, these points share the same predicted value or θ_0 and distributed on each of the independent axis. It also includes the origin. The adoption of regularization essentially claims that there is no dependence of the prediction on any of the features. The hypothesis, of course, is just the opposite, hence

[1] You must have realized that the choice of $v^{(i)} = \theta_0$ violates our original requirement that it should be independt of θ_i. To simplify the discussion, let's assume θ_0 is not to be optimized. Another way to put it is to shift y by $\langle y \rangle$ before regression and scale the features by the mean as well, then the model shouldn't involve the constant term.

the hypothesis is associated with some “bias". The combination of agnostic regularization and the biased hypothesis reminds me of the central idea of Bayes statistics. In Bayes, the prior knowledge has to be “unbiased", and the training set makes the posteori knowlege more “biased". Here, the two parts are mixed together in the hope to achieve a balance.

The strength of the agnostic regularization is adjusted by λ. In the interpretation above, it translates to the distance of the extra points from the origin. In the hyper-plane spanned by the features and the prediction, these extra points, $(\vec{u}^{(i)}, v^{(i)})$ $(i = 0, \cdots, n)$, define a hyperplane which is “flat" (agnostic). The size of the plane is $\sqrt{\lambda}$. If λ is bigger, it will be more important in the regression because it covers a larger range of feature values. When it is smaller, it will not be very signficant compared with the original training points. In the extreme case that $\lambda \sim 0$, these extra points gather around $(0, \theta_0)$, no longer serving as extra constraints.

An immediate application of this interpretation is the choice of λ. It should be based on the distribution of feature values. Because there is only one lambda available, it is obviously beneficial to perform feature scaling. Or else, there is no way to choose a universal λ that is appropriate for all features. Of course, we could use a different λ for different θ, but this is adding to the complexity of the model furthermore.

In practice, it may be helpful to study the response of θ_i with respect to the change in λ. As λ increase, some coefficients may be more resilent than others, and these must be more important features. For those θ_i that approaches zero quickly, we might remove them just as well.

Another potential development is to dig deeper into the Bayes interpretation. Is there a way to adjust λ based on training set in iterations: starting from a large λ, as more training data are fed to the program, λ is reduced. I am not sure if this makes sense.

4.3 Polynomial model

Without increasing feature set, one can increase the complexity of the model, for example, by increasing the polynomial order. To simplify the discussion, let’s assume that the features have been properly scaled so that it is reasonable to use the following hypothesis,

$$h(\vec{\theta}, x) = \sum_{i=1}^{n} \theta_i x^i, \tag{4.7}$$

or, in matrix format,

$$h(\vec{\theta}; x) = \vec{\theta} \cdot \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^n \end{pmatrix}. \tag{4.8}$$

Following the same idea used in the linear case, in order to interprete the regularization term as the cost function associated with some extra points, we need to find one or more points (x_i, y_i) independent of θ_i that satisfies the following:

$$\sum_{i=1}^{m} \Big[h(\vec{\theta}, x_m) - y_m\Big]^2 = \sum_{i=1}^{n} \theta_i^2. \tag{4.9}$$

Since these points should reflect our unbiased knowledge, it seems only reasonable to choose a vanishing y_m. In fact, we should choose

$$y_m = \sum_{i=1}^{n} \theta_i \langle x^n \rangle, \tag{4.10}$$

which reduces to 0 only for some cases even though $\langle x \rangle = 0$ due to feature scaling.

For now, with $y_m = 0$, Eq. 4.9 is reduced to the following matrix format,

$$||\vec{\theta} \cdot \mathbf{X}||^2 = ||\vec{\theta}||^2, \tag{4.11}$$

where

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \\ x_1^2 & x_2^2 & \cdots & x_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \cdots & x_m^n \end{pmatrix}. \tag{4.12}$$

It is not clear to me at this point how to find x_i that is independent of θ_i. In a simpler case where $m = n$, X needs to be a rotation matrix to satisfy Eq. 4.11. This is because Eq. 4.11 only requires that the norm of the l.h.s and r.h.s is the same, and rotation keeps the norm ([2]) If we recall the choice of extra point in the linear case, X $= \sqrt{\lambda} I$, and its null space is the whole feature space. This is stronger than Eq. 4.11. Of course, we could have chosen a more general rotation matrix at that time.

However, it is probably impossible to choose x_i such that X is a rotation matrix, simply because of the strong relation between different rows of X. In an almost trival case of $n = 2$, it we can choose $x_1 = 1$ and $x_2 = -1$ to satisfy Eq. 4.11. In particular, the cost function associated with these two points is

$$\begin{aligned} J &= (\theta_1 + \theta_2)^2 + (\theta_1 - \theta_2)^2 \qquad (4.13) \\ &= 2(\theta_1^2 + \theta_2^2). \qquad (4.14) \end{aligned}$$

If we attempt $m > n$, at least for each choice of $\vec{\theta}$, we should be able to find a set of x_i that can satisfy Eq. 4.11. But can we find such a set that is independent of $\vec{\theta}$?

[2] It's been implicit that $\lambda = 1$. If not, we have to associate a uniform weight of λ with each of these extra points.

Playing with another trivial choice of $(1, 0)$ means that its cost is $(\theta_1 + \theta_2 + \cdots + \theta_n)^2$. We cannot use this term for regularization, because it doesn’t acutally penalize big θ_i. In another word, adding $(1, 0)$ as an extra point for regularization doesn’t disfavor higher order term, because higher order polynomial can still go through $(1, 0)$.

4.4 Support Vector Machine

A lot of heuristic will show up here ...

5

Principle component analysis: Moment of inertia and rotational spectroscopy

5.1 Moments of inertia

When looking at a cloud of data points, it's just very natural to think of them as mass points. A collection of mass points whose locations are fixed is nothing but a rigid body. This is no stranger to anyone who took college physics. For those with exposure to classical mechanics, it's also natural to pull out the definition of inertia tensor.

$$\mathbf{I} = \begin{pmatrix} I_{xx} I_{xy} I_{xz} \\ I_{yx} I_{yy} I_{yz} \\ I_{zx} I_{zy} I_{zz} \end{pmatrix}, \tag{5.1}$$

where I_{xx}, I_{yy} and I_{zz} are moments of inertia with respective to x, y and z axis respectively and off-diagonal terms are products of inertia. In particular,

$$I_{xx} = \int_V (y^2 + z^2)\, dV, \tag{5.2}$$

and

$$I_{xy} = I_{yx} = -\int_V xy\, dV. \tag{5.3}$$

Inertia tensor is then readily used to compute the principal axis which form the basis set of a new space in which the tensor is diagonal. These new axis are the "stable" axis of rotation of the rigid body. If the body rotates around one of its principal axis, its angular momentum is parallel to this axis due to the fact that the principal axis is one of the eigenvectors of the inertia tensor. If the body rotates around an arbitrary axis, its anguluar velocity $\vec{\omega}$ consists of contribute from those along all three principal axis, and its angular momentum, $\vec{L} = \mathbf{I} \cdot \vec{\omega}$, also has contribute from different direction with *different* weight being the corresponding eigenvalue. The angular momentum, in general, would be at an angle with the axis of rotation. Without extra force, the body would gradually assume a rotation axis along the principal axis with the biggest or the smallest eigenvalue. In this sense, not all principal axis are created equal.

A whole textbook could be written about the rotational movement of a rigid top. Not only in mechanics, but also in fields like molecular spectroscopy. My first exposure to this idea was in physics class, and later encountered it again in the study of rotational spectroscopy. The idea of inertia moment was used in the context of quantum mechanics to describe the movement of molecules, which were treated as a collection of points. It's all very natural to think about the distribution of these points in terms of their principal axis and visualize them as tops of various shape, such as prolate or oblate.

5.2 Variance-covariance matrix

Without thinking about physics, data scientist cares about the variation of points. Of course, bigger variations are of more importance, and smaller variations could be neglected without much consequences. The variance-covariance matrix is defined to quantify the distribution of data with respect to different axis.

$$\mathbf{C} = \begin{pmatrix} C_{xx}C_{xy}C_{xz} \\ C_{yx}C_{yy}C_{yz} \\ C_{zx}C_{zy}C_{zz} \end{pmatrix}, \tag{5.4}$$

where C_{xx}, C_{yy} and C_{zz} are variances of feature x, y and z respectively and off-diagonal terms are covariances. In particular,

$$C_{xx} = \langle (x - \bar{x})^2 \rangle, \tag{5.5}$$

and

$$C_{xy} = \langle (x - \bar{x})(y - \bar{y}) \rangle. \tag{5.6}$$

Normally, the features are normalized so that the averages are zero. »»»> 986016af816d552987f0644f981fa76ff62237be

6

Support Vector Machine: A sketch of key points

6.1 Motivation

A sketch of milestones to reach SVM

7

Entropy cost: Boltzmann or Shannon?

7.1 Boltzmann

8

Feature scaling: the Π theorem

8.1 Buckingham's contribution

9

k-mean: the Harmonic mean

9.1 Interpretation in terms of harmonius average