Random Thoughts on Machine Learning

A Layman's Perspective

Yu Liu[1]

May 25, 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

Gradient descent: a first-order approximation

1.1 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,

$$d\theta = \mathcal{F}(J(\theta_0), J'(\theta_0), \cdots). \tag{1.1}$$

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

$$d\theta = kJ'(\theta_0). \tag{1.2}$$

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

$$d\theta J'(\theta_0) \leq 0. \tag{1.3}$$

Eq. 1.2 and Eq. 1.3 together implies $k \leq 0$. In a more common format, the gradient descent iteration reads

$$\theta_j = \theta_i - \alpha J'(\theta_i), \tag{1.4}$$

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. 1.4 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.

1.2 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

$$\theta_j = \theta_i - \frac{f(\theta_i)}{f'(\theta_i)}. \tag{1.5}$$

Realizing that minimizing $J(\theta)$ is equivalent to solving $J'(\theta) = 0$ (to some extent), Eq. 1.5 suggests the following iteration

$$\theta_j = \theta_i - \frac{J'(\theta_i)}{J''(\theta_i)}. \tag{1.6}$$

Comparing Eq. 1.4 and Eq. 1.6, one finds that Newton's method is equivalent to gradient descent with a learning rate determined by the curvature,

$$\alpha = \frac{1}{J''(\theta)}. \tag{1.7}$$

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 thatthe 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.

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,

$$\begin{aligned} \theta_j &= \theta_i - \frac{f(\theta_i)}{f'(\theta_i)}, & (1.8)\\ &= \theta_i - \frac{2f(\theta_i)f'(\theta_i)}{2[f'(\theta_i)]^2}, & (1.9)\\ &= \theta_i - \alpha J'(\theta_i), & (1.10) \end{aligned}$$

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

$$\theta_j = \theta_i - \frac{2J(\theta_i)}{J'(\theta_i)}. \tag{1.11}$$

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!)

$$\begin{aligned} \theta_j &= \theta_i - \frac{f'(\theta_i)}{f''(\theta_i)}, & (1.12)\\ &= \theta_i - \frac{J'(\theta_i)}{J''(\theta_i) - \frac{[J'(\theta)]^2}{2J(\theta)}} & (1.13) \end{aligned}$$

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.

2

Regularization: the “hidden" facts?

2.1 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.

2.2 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:

$$h(\vec{x}; \vec{\theta}) = \theta_0 + \vec{\theta} \cdot \vec{x} \tag{2.1}$$

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

$$J(\vec{\theta}) = \sum_{i=1}^{m} \left[y^{(i)} - h(\vec{x}^{(i)}; \vec{\theta}) \right]^2 . \tag{2.2}$$

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

$$J(\vec{\theta}) = \sum_{i=1}^{m} \left[y^{(i)} - h(\vec{x}^{(i)}; \vec{\theta}) \right]^2 + \lambda \sum_{i=0}^{n} \theta_i^2, \tag{2.3}$$

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

$$\vec{\epsilon} = \vec{y} - \left[\vec{\theta_0} + \vec{\theta}^T \cdot \mathbf{X} \right], \tag{2.4}$$

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

$$J(\vec{\theta}) = \vec{\epsilon} \cdot \vec{\epsilon} + \lambda(\theta_0^2 + \vec{\theta} \cdot \vec{\theta}). \tag{2.5}$$

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

$$\left[v^{(1)} - \left(\theta_0 + \vec{\theta}\cdot\vec{v}^{(1)}\right)\right]^2 = \lambda\theta_1^2. \tag{2.6}$$

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 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

[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.

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.

2.3 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,

$$h(\vec{\theta}, x) = \sum_{i=1}^{n} \theta_i x^i, \tag{2.7}$$

or, in matrix format,

$$h(\vec{\theta}; x) = \vec{\theta} \cdot \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^n \end{pmatrix}. \tag{2.8}$$

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:

$$\sum_{i=1}^{m} \Big[h(\vec{\theta}, x_m) - y_m \Big]^2 = \sum_{i=1}^{n} \theta_i^2. \tag{2.9}$$

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

$$y_m = \sum_{i=1}^{n} \theta_i \langle x^n \rangle, \tag{2.10}$$

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. 2.9 is reduced to the following matrix format,

$$||\vec{\theta} \cdot \mathbf{X}||^2 = ||\vec{\theta}||^2, \tag{2.11}$$

where

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \\ x_1^2 & x_2^2 & \cdots & x_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \cdots & x_m^n \end{pmatrix}. \tag{2.12}$$

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. 2.11. This is because Eq. 2.11 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, $\mathrm{X} = \sqrt{\lambda} I$, and its null space is the whole feature space. This is stronger than Eq. 2.11. 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. 2.11. In particular, the cost function associated with these two points is

$$\begin{aligned} J &= (\theta_1 + \theta_2)^2 + (\theta_1 - \theta_2)^2 && (2.13) \\ &= 2(\theta_1^2 + \theta_2^2). && (2.14) \end{aligned}$$

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. 2.11. But can we find such a set that is independent of $\vec{\theta}$?

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 $(1, 0)$.

2.4 Support Vector Machine

A lot of heuristic will show up here ...

[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.

3

Principle component analysis: Moment of inertia and rotational spectroscopy

3.1 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.

$$\mathbf{I} = \begin{pmatrix} I_{xx} I_{xy} I_{xz} \\ I_{yx} I_{yy} I_{yz} \\ I_{zx} I_{zy} I_{zz} \end{pmatrix}, \tag{3.1}$$

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,

$$I_{xx} = \int_V (y^2 + z^2)\, dV, \tag{3.2}$$

and

$$I_{xy} = I_{yx} = -\int_V xy\, dV. \tag{3.3}$$

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.

3.2 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.

$$\mathbf{C} = \begin{pmatrix} C_{xx} C_{xy} C_{xz} \\ C_{yx} C_{yy} C_{yz} \\ C_{zx} C_{zy} C_{zz} \end{pmatrix}, \tag{3.4}$$

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,

$$C_{xx} = \langle (x - \bar{x})^2 \rangle, \tag{3.5}$$

and

$$C_{xy} = \langle (x - \bar{x})(y - \bar{y}) \rangle. \tag{3.6}$$

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

4

Entropy cost: Boltzmann or Shannon?

4.1 Boltzmann

5

Feature scaling: the Π theorem

5.1 Buckingham's contribution

6

k-mean: the Harmonic mean

6.1 Interpretation in terms of harmonius average