MATH 462 LECTURE NOTES

ADAM M. OBERMAN

Contents

Date: September 10, 2021.

1. Week 1: Introduction

This note covers lectures 1 and 2. Reference [SSBD14, Chapter 9], Linear Predictors. Section 9.2.

1.1. **Data and features.**

Data can be represented in many ways. We will always work with a vector of features, write $x \in \mathbb{R}^d$ for a vector of features. The x notation emphasizes that it comes from the data.

In this section, regression, we are learning a real number, $y \in \mathcal{Y} = \mathbb{R}$. Write

$$S_m = \{(x_1, y_1), \dots, (x_m, y_m)\} \tag{S_m}$$

for the data set with m pairs (x, y) of data, and values, respectively.

1.2. **Linear models.**

Our first goal is to *fit* the dataset. (Later we will study whether the model generalizes to unseen data).

We will try to learn a model $h : \mathbb{R}^d \to \mathcal{Y} = \mathbb{R}$.

For now, the models will linear functions of $w \in \mathbb{R}^d$. Later we will consider more general models. Write $w \in \mathbb{R}^d$ for the parameters (or weights). We consider the linear model

$$h_w(x) = w \cdot x = \sum_{i=1}^{d} w_i x_i \tag{1}$$

1.3. **Standard regression loss.** Since we are studying a regression problem, we can talk about the error of a model. The error of the model, on data (x, y), is defined to be

$$e = h_w(x) - y$$

We want to make each component of the error small, so we introduce a non-negative loss function, which is a function of the error. It should be increasing in the error. [1]

$$\ell : \mathbb{R} \to \mathbb{R}^+$$

The most important regression loss is the quadratic loss

$$\ell_2(y_1, y_2) = \frac{1}{2}(y_1 - y_2)^2$$

Definition 1.1 (Empirical Loss)**.** Given

(1) the dataset S_m, as in (S_m),
(2) a model $h_w : \mathbb{R}^d \to \mathbb{R}$,

and a loss, ℓ, the empirical loss of the model h_w, on the dataset S_m, is given by

$$\widehat{L}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(h_w(x_i), y_i) \tag{EL}$$

[1] approach : makes sense to start with a working definition and let it evolve to be more refined

Next we can define the problem of *fitting data using a parameterized model*. This data fitting problem is called empirical loss minimization, in a context where we make statistical assumptions on the data.

Given a family of regression models

$$\mathcal{H} = \{h_w(x) : \mathbb{R}^d \to R, w \in \mathbb{R}^d\}$$

We find the best fit to the data, using the empirical loss. We can write this two ways. The first way emphaisizes that we are finding a function

$$\min_{h \in \mathcal{H}} \widehat{L}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h_w(x_i), y_i)$$

The second way emphasizes that we are finding parameters

$$\min_{w} \widehat{L}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(h_w(x_i), y_i)$$

2. Discussion

[Refer to in class discussion] We discussed difference conceptual solution methods.

(1) Analytical: move the problem into a new category where solutions are known. - AE: Give an explicit formula for solution, e.g. $w = M^{-1}b$ - AR: Reduce the problem to a simpler class of problems, where more is know about the solution, - e.g. For example, Theorem 5.3 reduces the quadratic model fitting problem to a system of linear equations. (regression) - e.g. Later we will see in the case of (SVM classification) than the problem is reduced to a linear programming problem (which is a well-understood family of optimization problems)

(2) geometric - Provide a geometrical notion of solution which can be implemented graphically. - E.g. Draw the line for linear regression - E.g. Illustrate the class boundaries for classification - E.g. draw the clusters.

(3) algorithmic: Give an algorithmic solution method - e.g. k-means algorithm - AS a sketch of an algorithm (e.g. row reduction for linear systems) - AI an optimal implementation of an algorithm (e.g. optimized code to solve Mx = b)

(4) abstract: instead of a solution, prove that the problem has a solution. This will cover a wider class of problems. Then in special cases, can offer a solution method. Useful because it tells us that the problem is solvable. Still need to find a solution method.

3. Case Study: regression loss design for soft grading

We discussed loss design for building a soft grading scheme, which would allow a low (outlier) grade to have a mitigated effect on the average. [Half a lecture on this]

4. (Lecture 3) Matrix notation

This note covers matrix-vector notation. See the sections in [DFO20].

Remark 4.1. The matrix notation above emphasizes the fact that we have access to all the data. This is consistent with may learning problems, and with modern code implementations (e.g. supervised learning datasets stored on computer, and PyTorch). In other learning problems, called 'online', data arrives in a stream, as a sequence, and don't have access to to the full dataset. This can affect the way we look at the problem, but not the math at this stage.

4.1. **Matrix vector notation.** We sometimes want to think of the feature data, which consists of m feature vectors, $(x_1, \dots x_m)$ as a matrix

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_{11} \dots x_{1d} \\ x_{21} \dots x_{2d} \\ \vdots \\ x_{m1} \dots x_{md} \end{bmatrix}$$

Here X has m rows, and each row is a data (or feature) vector in $\mathbb{R}^d$.

We also write the y-values as a column vector (of size m),

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix},$$

or, to save space, $y = [y_1, \dots, y_m]^T$.

Likewise, we can write the model values as

$$h_w(X) = [h_w(x_1), \dots, h_w(x_m)]^T$$

The error vector, $e = [e_1, \dots, e_m]^T$ as

$$e = h_w(X) - y$$

The loss vector (overloading notation) as

$$\ell(e) = [\ell(e_1), \dots, \ell(e_m)]$$

Finally, the empirical loss is given by

$$\widehat{L}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell_i$$

which we could write as $\frac{1}{m} 1 \cdot \ell(e)$.

4.2. **Linear model and quadratic loss.** In the important case of the linear model and the quadratics loss, write the model values as the matrix vector product

$$h_w(X) = Xw$$

The error vector as

$$e = h_w(X) - y = Xw - y$$

For the case of quadratic loss (EL) becomes $\widehat{L}(w) = \|e\|^2/2$, or

$$\widehat{L}(w) = \frac{1}{m}\|Xw - y\|^2$$

5. (Lecture 4): Gradients and minimizers

In this section we review material (which most students have forgotten) on conditions for a minimum. We focus on taking the gradient of a function of several variables (the function will be $\widehat{L}(w)$ and the variables are $w \in \mathbb{R}^d$.)

5.1. **Calculus review.**

- for a function of one variable $\widehat{L}(w)$, $w \in \mathbb{R}$, a critical point is when $\widehat{L}'(w) = 0$
- Every local minimum is a critical point. A critical point can be a local minimum, local maximum, or saddle point.
- If the second order condition holds $\widehat{L}''(w) > 0$, then the critical point is also a local minimum
- If the function is convex (for example when $\widehat{L}''(w) \geq 0$ at all w, then every critical point is a *global* minimum.

5.2. **Analytical solution methods for the minimizer in one variable.**

Consider minimizing

$$\widehat{L}(w) = \frac{1}{m}\sum_{i=1}^{m} q(wx_i - y_i).$$

This problem corresponds to (EL) as in Definition 6.2, when $d = 1$. We want to find a critical point, w, which satisfies

$$\widehat{L}'(w) = 0$$

By the chain rule, we have

$$\widehat{L}'(w) = \frac{1}{m}\sum_{i=1}^{m} q'(wx_i - y_i)x_i$$

In the case of the quadratic loss, we have $q(e) = e^2/2$, we obtain

$$0 = \frac{1}{m}\sum_{i=1}^{m}(wx_i - y_i)x_i$$

which simplifies to

$$\sum_{i=1}^{m} wx_i^2 = \sum_{i=1}^{m} y_i x_i$$

5.3. **Vector calculus facts.** Now consider a function of d variables, $\widehat{L} : \mathbb{R}^d \to \mathbb{R}$. The gradient of the function is a vector defined at each w,

$$g(w) = \nabla\widehat{L}(w) = [g_1(w), \dots g_d(w)]^T$$

where each component is partial derivative

$$g_j(w) = \frac{\partial}{\partial w_j}\widehat{L}(w)$$

- The gradient vector $g(w) = \nabla\widehat{L}(w)$ points in the direction of greatest increase of the function $\widehat{L}$ at w.
- A critical point w is a point where $g(w) = 0$.
- As in the one variable case, every local minimum is a critical point. A critical point can be a local minimum, local maximum, or saddle point.
- As in the one variable case, there is a condition for a critical point to be a local minimum: the Hessian matrix $H(w)$ is positive-definite. Here $H(w)_{ij} = \frac{\partial^2}{\partial_i\partial_j}\widehat{L}$. (This condition can be difficult to check).

- As in the one variable case, if the function is convex, then every critical point *global* minimum.

Remark 5.1. In our case, we will be able to show directly that our function $\widehat{L}$ are convex, provided the loss function ℓ is convex and the model is linear. We will get to that later. This simplifies the problem of finding a minimum, since we only need to find a critical point.

5.4. **Finding a critical point the loss.** Now for $d > 1$, we can apply the vector chain rule to find a critical point of (EL).

$$\widehat{L}(w) = \frac{1}{m}\sum_{i=1}^{m} q(w \cdot x_i - y_i).$$

where

$$w \cdot x_i = \sum_{k=1}^{d} w_k x_{ik}$$

We will be using the chain rule, but this time the vector case. Let's start the calculation as follows. (We are reversing the steps to make it easier).

We have

$$e_i = w \cdot x_i - y_i = \sum_{k=1}^{d} w_k x_{ik} - y_i$$

So

(*) $$\frac{\partial}{\partial w_j} e_i = x_{ij}$$

Next, by the chain rule

$$\begin{aligned}\frac{\partial}{\partial w_j} q(e_i) &= q'(e_i) \frac{\partial}{\partial w_j} e_i \\ &= q'(e_i) x_{ij} && \text{by (*) above}\end{aligned}$$

We have just computed one component, g_j, of the gradient. Combining these, we obtain

$$\nabla q(e_i) = q'(e_i) x_i$$

Note, the last equation is the vector $x_i \in \mathbb{R}^d$, multiplied by the scalar $q'(e_i)$.

Next, if we consider

$$\widehat{L}(w) = \sum_{i=1}^{m} \ell(e_i)$$

we obtain

$$\frac{\partial}{\partial w_j} \widehat{L}(w) = \sum_{i=1}^{m} \frac{\partial}{\partial w_j} \ell(e_i)$$

so

$$\frac{\partial}{\partial w_j} \widehat{L}(w) = \sum_{i=1}^{m} \ell'(e_i) x_{ij}$$

which again, is one component of the gradient. Now we combine the components, to obtain

$$\nabla \widehat{L}(w) = \sum_{i=1}^{m} \ell'(e_i) x_i$$

This last equation means: for each data point $x_i \in \mathbb{R}^d$, we are multiplying it by $\ell'(e_i)$, and summming to obtain the gradient in w.

We have just proved the following result, which we record.

Theorem 5.2. *Consider* (EL) *with a linear model. Then*

(EL')
$$\nabla\widehat{L}(w) = \sum_{i=1}^{m} \ell'(e_i)x_i$$

As a special case, we have the following. In the quadratic case, $\ell'(e_i) = e_i = (w \cdot x_i - y_i)$. So using (EL') a critical point is characterized by

$$\sum_{i=1}^{m}(w \cdot x_i)x_i = \sum_{i=1}^{m} y_i x_i$$

5.5. **Linear equation for the minimizer.** In the important case of the quadratic loss, we can characterize the best fitting linear hypothesis using a linear equation involving the data matrix.

Theorem 5.3. *Consider the problem* (EL) *with linear hypothesis. Then the loss minimizer is given by the solution of the linear equation*

(MEL)
$$X^T X w = X^T y$$

Proof. See Exercises or refer to textbook Example 5.11 in [DFO20]. □

5.6. **Example calculations and HW.** TODO:

In tutorial, we also did a two dimensional example $d = 2$, an emphasize that the calculation can be performed two ways

(1) by computing partial derivatives with respect to w_1 and w_2 and solve the resulting system
(2) by considering the linear system (MEL). The advantage of the latter formulation is that the matrix $X^T X$ can be compute once, and the linear system can be solved, for different values of y.

6. (Lecture 3): Non-standard regression losses

In this section we try to understand the different behaviour of regression losses. Each loss has a different behaviour with respect to averages, smoothness, and outliers.

6.1. **Other regression losses.**

Definition 6.1 (Other regression losses)**.** The ℓ_1 loss,

$$\ell_1(y_1, y_2) = |y_1 - y_2|.$$

The Huber loss, with scale δ, $\ell_H(y_1, y_2) = q_\delta(y_1 - y_2)$, where

$$q_\delta(e) = \begin{cases} e^2/2 & |e| \leq \delta \\ \delta(|e| - \delta/2)) & |e| \geq \delta \end{cases}$$

[`https://en.wikipedia.org/wiki/Huber_loss` insert plot of huber loss]

The Huber loss incorpates a scale, δ and it is designed to be less sensitive to *outliers*, which lie more that δ away from the central value. We will study it further in the sequel.

Definition 6.2. [Regression loss basics] A regression loss takes the form Assume $\ell(y_1, y_2) = q(y_1 - y_2)$ where $q(e)$ is
(i) non-negative, with $q(e) = 0 \iff t = 0$ (so that $\ell(y_1, y_2) = 0 \iff y_1 = y_2$)
(ii) an increasing function of $|e|$.

To illustrate this consider the following model problem.

- Use dataset (S_m).
- Define $h_w(x) = w$. Let the loss be $\ell(h_w(x), y) = q(e)$, where $e = h_w(x) - y$
- Study (EL).

The problem we study is to minimize

$$\widehat{L}(w) = \frac{1}{m}\sum_{i=1}^{m} q(e_i), \quad e_i = w - y_i \tag{EL.CV}$$

which is a problem which finds the stands for the Central Value induced by the loss, (EL.CV). (Refer to `https://en.wikipedia.org/wiki/Central_tendency` for more examples).

Lemma 6.3. *A critical point of* (EL.CV) *is given by*

$$\widehat{L}'(w) = \sum_{i=1}^{m} q'(e_i) = 0 \tag{2}$$

Proof. Following the type of calculations in the section above, we differentiate to obtain (2). □

Definition 6.4. Define the median to be: the middle value (after sorting), if there are an odd number of values, and the average of the middle values (when even).

For the Huber loss, given a w, define an outlier to be any y_i with $|e_i| \geq \delta$, and otherwise y_i is an inlier. Define the Huber central values by: the w^* chosen to that the average of: errors (for inliers) and δ times sign of the errors (for the outliers) is equal to zero.

Note: this doesn't cover every case, but it's a good working definition. The true definition will be given by the theorem.

Theorem 6.5. *Consider the problem* (EL.CV), *the solution w^* is given as follows.*

(1) *Using the ℓ_2 loss, $w^* = \frac{1}{m}\sum_{i=1}^{m} y_i$, the average of y.*
(2) *For the ℓ_1 loss, the median is a solution (e.g. (1, 2, 3, 4), 2.5 is a solution, but so are 2 and 3).*
(3) *For the Huber loss, we can characterize the solution as follows. Define the central values to be those y_i within δ of w^*. Define the outliers to be the rest. Then w^* is a weighted average of the following: the mean of the*

central values, weighted by their number, and $-1, +1$ *for each of the outliers (depending on if they are above or below).*

Also, we can see how the Huber loss combines the other losses, since we can also characterize the median as finding a value so that there are equal above and below.

Sketch of proof. Case 1. In the case of quadratic loss: this equation is:

$$\sum_{i=1}^{m} e_i = 0$$

which means the errors sum to zero. This is equivalent to $w = \frac{1}{m}\sum y_i$, the average. So w^* is the average.

Case 2. In the case of the ℓ_1 loss, it is differentiable if w^* is not equal to one of the y_i, in which case we have $h'(e_i) = sgn(e_i)$. This leads to

$$\sum_{i=1}^{m} sgn(e_i) = 0$$

We claim that we can take the median as a solution. However, in the case of odd number of data points, when the median is the middle value, then we are not differentiable. In this case, we can verify directly that the middle value is a minimizer.

Case 3. In the case of Huber loss, we want to solve (2). Geometrically Suppose we know the minimizer w^*. Then define an outlier to be any y_i with $|e_i| \geq \delta$. For the inliers, $h'(e_i) = e_i$. For the outliers, $h' = \delta sgn(w - y_i)$. So the w^* is characterized by: average of: errors (for inliers) and δ times sign of the errors (for the outliers). □

6.2. **Discussion.**

6.2.1. *Optimization versus rules.* Which is better, using a loss to define w^*, or giving a formula for a central value? There is no correct answer. Giving a formula can be simpler, e.g. Windsorizing truncate the top 10 and bottom 10 percent of values, then take the average. However, giving an optimization problem may better specify what we want.

For example, Huber allows us to specify the width of the central part, versus the fraction of points in there. So that Huber can reduce to quadratic (when all points are inliers) or l1 (when all points are outliers). But Windsorizing will always be different, it will always have outliers.

7. Linear regression, feature vectors

7.1. **Data versus features.** If we want to emphasize the distinction between the raw data, x, and the features, $f(x)$. Here we consdier $f : \mathcal{X} \to \mathbb{R}^d$ to be a feature mapping.

Write

$$S_m = \{(f_1, y_1), \dots, (f_m, y_m)\}$$

We will adopt the convention that d for the dimension of w, $x, f \in \mathbb{R}^d$.

When specify features, we write

$$h_w(x) = w \cdot f = \sum_{i=1}^{d} w_i f_i(x) \tag{3}$$

Example 7.1. At first glance, this setting does not appear to allow for an affine model $h_w(x) = w \cdot x + b$. However, we can use the idea of a feature map to include this. For $x \in \mathbb{R}^k$ define $d = k + 1$ and define the feature map to be

$$f(x) = (x, 1) \in \mathbb{R}^d$$

Then the linear model (3) correponds to the affine function of x,

$$w \cdot f = \sum_{i=1}^{d} w_i f_i = \sum_{i=1}^{d-1} w_i x_i + w_d$$

where setting $w_d = b$ recovers the affine model.

we will discuss features in more detail later

7.2. **Polynomial regression.** Consider one dimensional data, and let the features be polynomials. For example, let $x \in \mathbb{R}$. Consider the feature map, $f(x) = (1, x, x^2, x^3)$, so that $f : \mathcal{X} = \mathbb{R} \to \mathbb{R}^3$. This feature map corresponds to (second order) polynomial regression. Note that the features are nonlinear. However the model (1) is linear in the weights, which is the important part for optimization: because the data (and the features) are fixed, we will be optimizing over the weights.

The quality of the features is, of course, very important. It will affect: (i) how well we can optimize (in other words, how much we can reduce the loss using a linear fit, and (ii) how well we can generalize (in other words, how well the model will work on new data).

7.3. **Neural network features.** We can consider the setting where have a trained neural network features $f(x)$. Then we perform linear regression using these features.

References

[DFO20] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.