MATH 462 LECTURE NOTES

ADAM M. OBERMAN

Contents

Date: September 8, 2021.

1. Week 1

1.1. **Introduction.** This note covers lectures 1 and 2.

Reference [SSBD14, Chapter 9], Linear Predictors. Section 9.2.

1.2. **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

$$S_m = \{(x_1, y_1), \dots, (x_m, y_m)\} \tag{1}$$

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

1.3. **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

$$h_w(x) = w \cdot x = \sum_{i=1}^{d} w_i x_i \tag{2}$$

1.4. **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 1.1. 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

1.5. **Regression Losses and errors.** 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

$$e = h_w(x) - y \tag{4}$$

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 1.2 (Empirical Loss)**.** Given

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

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

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

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

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

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

1.6. **Other regression losses.**

Definition 1.3 (Other regression losses)**.** The ℓ_1 loss,

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

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

$$h_\delta(e) = \begin{cases} e^2/2 & |e| \le \delta \\ \delta(|e| - \delta/2)) & |e| \ge \delta \end{cases}$$

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

2. Solution Methods

2.1. **Analytical solution methods for the minimizer when $d = 1$.** *In class we did the simple example calculation for $d = 1$. Here it is a one-dimensional calculus problem to minimize $\widehat{L}(w)$ by solve $\widehat{L}'(w) = 0$.*

Consider (EL) as in Definition 1.4. Suppose $d = 1$. It is a calculus exercise to find the condition for the minimizer. The solution is given by differentiating,

$$0 = \widehat{L}(w)' = \frac{1}{m}\sum_{i=1}^{m} \ell'(wx_i, y_i)(wx_i)' \tag{5}$$

$$= \frac{1}{m}\sum_{i=1}^{m} \ell'(wx_i, y_i)x_i \tag{6}$$

In the case of the quadratic loss, we have $\ell(wx_i, y_i) = (wx_i - y_i)^2/2$, so we obtain

$$0 = \frac{1}{m}\sum_{i=1}^{m}(wx_i - y_i)x_i \tag{7}$$

or

$$\sum_{i=1}^{m} w x_i^2 = \sum_{i=1}^{m} y_i x_i \tag{8}$$

In what follows, we consider the case $d > 1$.

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

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

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

where X has m rows, where each row is a feature vector in $\mathbb{R}^d$.

In this context, it is natural to write the y-values as a column vector (of size m),

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

Likewise, we can write the model values as

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

Since the model is linear, we can write

We can also write the model values $= (h_w(x_1), \dots h_w(x_m))$ as the matrix vector product

$$h_w(X) = Xw$$

Also write, the error vector, $e = [e_1, \dots, e_m]^T$ as

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

Remark 2.1. 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.

For the case of quadratic loss, the model loss (EL) becomes

$$\widehat{L}(w) = \frac{1}{m}\|Xw - y\|^2 \tag{9}$$

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 (10). 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.

2.3. **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 2.2. *Consider the problem* (EL) *with linear hypothesis. Then the loss minimizer is given by the solution of the linear equation*

$$X^T X w = X^T y \tag{10}$$

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

3. Using the solution

Suppose now that we have solved the problem above, and we have a model $h_w(x)$, which best fits the data.

We are going to use the model to *predict* y given new, unseen data x.

In other words, we perform prediction.

Definition 3.1 (Model outputs)**.** Suppose we have the model h_w which is obtained by loss mimimization of (EL). Given a new x, the model predicts

$$y = h_w(x)$$

What can we say about y?

We can use *Statistical learning theory* to make assumptions about

- how the data was generated
- properties of the class of models

in order to make probabilistic predictions on the error of a learned model.

In other words, we want to estimate the error of the solution.

This is a different approach to *function fitting* in traditional applied math, where we make assumptions about the true function $y = f(x)$. (e.g. smoothness), in order to prove that the model we learn fits well.

In learning theory, we usually do not assume much about the true solution (agnostic).

4. 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 2.2 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.

5. Case Study: regression loss design for soft grading

We discusses loss design for building a soft grading scheme, which would allow a low (outlier) grade to have a mitigated effect on the average.

6. Week 2

6.1. **Linear regression, feature vectors.**

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 (2) 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).

6.2. **Interpreting the losses.** [This section is original material]

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.

To illustrate this consider the following model problem.

- Use dataset (1).
- Define $h_w(x) = w$.
- Study (EL).

This corresponds to

$$\min_w \frac{1}{m} \sum_{i=1}^{m} \ell(w, y_i) \tag{CVIL}$$

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

Definition 6.1. 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 6.2. *Consider the problem* (CVIL)*, 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. 1. First, the problem (EL) becomes

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

where we have used the notation

$$e_i = w - y_i$$

Assuming that h is differentiable at w^*, (we will need to check this), this means that we can find the minimum by solving

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

which corresponds to

$$\sum_{i=1}^m h'(e_i) = 0 \tag{11}$$

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 (11). 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). □

6.3. **Minimizing** (EL). Consider (EL) with a linear model.

Suppose we fix all the variables in w except one. The we compute the partial derivative

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

or, in gradient notation (which is just each component)

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

So this means that, setting the sum to zero *at a minimizer, the derivatives sum to zero*

6.4. **Abstract Regression Theory.**

Definition 6.3. [Regression loss design] Let's generalize the losses we saw to include the following. Assume $\ell(y_1, y_2) = h(y_1 - y_2)$ where $h(t)$ is
(i) non-negative, with $h(t) = 0 \iff t = 0$ (so that $\ell(y_1, y_2) = 0 \iff y_1 = y_2$)
(ii) an increasing function of t,
(iii) a convex function of t.

[TODO: Example check the losses we have seen all satisfy this.]

Theorem 6.4 (sketch)**.** *Consider the regression problem, as defined above, with a loss satisfying the assumptions of Definition 6.3. Then for $\lambda > 0$, there is a unique minimizer w^*. Moreover, in the case the quadratic loss this holds for $\lambda > 0$.*

Future theorem:

(1) Algorithm to find w^*. Gradient descent works.
(2) Generalization If the data is generated i.d.d. from a distribution, then we can bound the loss on new data.

Gradient descent:

$$w_{new} = w_{old} - h\nabla_w \widehat{L}_\lambda(w_{old})$$

Sketch of why it works: secant of convex function, decreases with small enough h.

$$\widehat{L}_\lambda(w_{new}) \approx \ldots Taylor\ expansion$$

6.5. **Discussion.**

6.5.1. *Optimization versus rules.* Which is better, using (CVIL) 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. Classification

7.1. **TODO: Logisitic Regression.** Notes TODO Refer to [SSBD14, Chapter 9], Linear Predictors. Sections 9.2.

7.2. **TODO: SVM.** Refer to [DFO20] Chapter.

7.3. **TODO : Classification with Softmax-KL.**

8. Exercises

8.1. **Example calculations.**

(1) (One variable quadratic regression) Consider (EL) with $X = \{1, 2, 3\}$, $Y = \{2, 4, 5\}$. Use linear model $h = w*x$. Solve the problem with the quadratic loss. Hint: write down $\hat{L}(w)$ and find the solution by solve $\hat{L}'(w) = 0$.

(2) (One variable ℓ_1 regression) Same setup as in the previous problem: consider (EL) with $X = \{1, 2\}$ $Y = \{1, 3\}$. Use linear model $h = w*x$. Solve the problem with the ℓ_1 loss. Hint: plot the function $\hat{L}(w)$, which is piecewise linear, and find the minimum value (by finding the intersection of two lines).

(3) (Polynomial regression). Suppose our data points are $x = 0, .1, \dots, .9, 1$. Let $f(x) = (1, x)$ (affine linear regression). Set Choose Y to be random data on $[0, 1]$. (i) Set up the data matrix, F. What are the sizes of F, $F^T F$, and w ? and plot the error and solution. Is the fit good? What is the size of the solution vector w^*. (ii) Same problem with $f(x) = (1, x, x^2, x^3)$.

(4) (Two variable quadratic regression) Set $X = \{(3, 0), (0, 2), (1, 1)\}$ $Y = \{6, 2, 5\}$ Setup the quadratic regression problem (i) by minimizing (EL) directly (i.e. take derivatives with respect to w_1 and w_2 and solve. (ii) by setting up the matrix equation (10) and solving it.

(5) Consider `https://en.wikipedia.org/wiki/Winsorized_mean` Give an example with 10 numbers where the 10% Winsorided mean is the same as the minimizer of the Huber loss (with, say $\delta = 2$). Explain the main difference between the Winsorized mean and the minimizer of the Huber loss? (Hint: the huber loss has a scale δ which determines the outliers, but the Winsorized mean has a fraction of values).

8.2. **Theory exercises.**

(1) Exercise: verify that in the case of the quadratic loss, (EL) becomes (9).

(2) Show that when $d = 1$ (10) reduces to (8)

(3) (Compare the regression loss functions) (i) Explain the statement **??** is a special case of (EL). What is the corresponding hypothesis? (iii) Consider $y_1, \dots y_{10}$ consisting of nine 0 and one value y. What is the solution (as a function of y) of (CVIL) for each of the three main regression losses (take $\delta = 1$ in the Huber).

(4) Prove Theorem 2.2. Hint: two ways to do this, 1. cite the matrix theory fact that $\min_w (Fx-b)^2$ is given by (10). 2. prove directly by taking derivatives of the loss with respect to each w_i and identify the equation.
(5) (Huber loss) Show that the Huber loss is continuous, and differentiable. Find the second derivative of the function.
(6) Prove Theorem 6.2.
(7) Check that the three main regression losses are all convex.

8.3. **Loss design exercises.**

(1) (Loss designs for grading scheme) Consider a grading scheme where there are five assignments. Suppose we want a grading scheme that is less sensitive to outliers, e.g. with a score of $.9, .9, .9.9, 0$ (one missed assigmnent) we don't want the hard penalty given by the average. At the same time, we want every grade to have a small effect (to encourage performance when possible). (i) propose a simple scheme to do this. (ii) Suppose we want a missed assigment to have an effect of no more than $\delta = .2$ on the average. Show that the Huber loss with $\delta = .2$ accomplished this (at least in the example above).
(2) (Loss design: flipped huber) Design a 'flipped' Huber loss function, which is quadratic for $|t| \geq \delta$ and equals $|t|$ for otherwise. Set up the quadratic so that the loss is continuously differentiable.
(3) (Loss-design Smooth-Huber). In this problem we find a smooth version of the Huber loss. Consider the function $h(t) = \log(\cosh(t))$. Prove that the function is even, and find the first nonzero term in the Taylor expansion. Conclude that it is nearly quadratic near $t = 0$. Show that the function asymptotes to $|x|$ as $|x| \to \infty$. Can you introduce a scale parameter δ as in the Huber function? Do it so that the function becomes nearly quadratic (or linear) for extreme values of δ.

9. Future lectures: Convex Learning Problems [SSBD14] Chapter 2

9.1. **Regularized Empirical Loss.** The regularized loss (will be explained later) is

$$\widehat{L}_\lambda(w) = \widehat{L}(w) + \lambda\|w\|^2$$

The empirical loss minimization (ELM) problem is to minimize $\widehat{L}(w)$, or more generally

$$\min_w \widehat{L}_\lambda(w) \tag{12}$$

We will work on proving the following theorem (will build up the theorem, and the details). We will use : convex analysis / optimization

10. Convexity Theory

Definitions of convex. Jensen's inequality. Supporting hyperplanes. Gradient descent. Sum of convex is convex. Convex functions have a unique minimum value. Strictly convex functions have a unique minimizer.

Gradient descent: Can always control the remainder term of the taylor expansion.

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.