MATH 462 LECTURE NOTES

ADAM M. OBERMAN

CONTENTS

Date: September 23, 2021.

1. Introduction to binary classification losses

Reference for this section [Mur12, Chapter 8] (mostly the first equation) or [Mur22, Section 5.1.2].

1.1. **Introduction to binary classification.** In the classification problem, the target set $\mathscr{Y}$ is a set of discrete labels. For the supervised classification problem, we are given a dataset (S_m) consisting of m pairs of (x_i, y_i), $i = 1, \ldots, m$, of data, $x_i \in \mathscr{X}$ and labels, $y_i \in \mathscr{Y}$,

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

The main difference is that we are learning one of K discrete classes, which we simply represent as one of the integers, $1, \ldots, K$. In order to keep notation similar to the previous case, we write

$$y \in \mathscr{Y} = \mathscr{Y}_K = \{1, \ldots, K\}. \tag{$\mathscr{Y}_K$}$$

The notation $\mathscr{Y}_K$ will be used to emphasize that this is a classification problem with K classes. There are conceptual differences depending on the number of classes involved. Two classes is called *binary classification*, more than two is called multi-classification. In fact we can consider the following:

- Binary classification for two classes, where the classes are balanced (e.g. identify a cat/dog picture). We write $\mathscr{Y}_2 = \{1, 2\}$ as in $(\mathscr{Y}_K)$, above.
- When there are more than two classes, we will usually consider the case where the classes are balanced (same number of samples per class), e.g. MNIST dataset of 10 digits, or CIFAR10/CIFAR100, with 10 and 100 classes, respectively.
- We also consider the case of binary classification consisting of a test, which can be positive of negative, which we can write as $\mathscr{Y}_\pm = \{-1, +1\}$. In this case, the probability of each class may be different (e.g. a medical test for a rare medical condition). In this case we care about the *error types*, False Positives or False Negatives.
- A classification loss $\ell(h, y)$, which measures how far our hypothesis is from being correct. It should be (piece-wise) differentiable as a function of h.
- Hypotheses consisting of a parameterized family of models, $h_w : \mathscr{X} \to \mathbb{R}^K$.

- We will still train our model to fit data by minimizing the expected loss (EL-C)

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

1.2. **Comparison with regression.** Our approach to classification is *score-based*, which means there are a number of similarities to the case of regression.[1]

At the abstract level we will still have:

These similarities mean that our approach of minimizing the loss (and the gradient based algorithms we will study to find the minimizers) will have many similarities to the case of regression.

However, there are also a number of differences which make the problem more subtle. The first difference is that we need a way to convert our linear model into one of a number of discrete classes. We will study each of the cases (Binary and multi-classification) from this perspective.

There is an added complication that our classification losses will be defined on the pair (s, c) for $s \in \mathbb{R}, c \in \mathscr{Y}$.

$$\ell_{class} : \mathbb{R} \times \mathscr{Y} \to \mathbb{R}^+$$

1.3. **Discussion on classification losses.** In this section we are studying classification losses. Most textbooks present the classification losses as already fixed, they don't derive them. The main approaches to (supervised) binary classification are

(DB) Model the Distance to the linear classification Boundary (using a specific notion of distance), usually using Support Vector Machines, [DFO20]. In this case, we combine a margin loss (which we will see later) with a linear model. The loss minizimization problem (EL-C) is piecewise differentiable and convex, it corresponds to a linear program.

[1] there are other classification methods, for example probabilistic, or clustering, see the discussion which follows. See also `https://en.wikipedia.org/wiki/Statistical_classification`

(CP) Model class probabilities. In this case, we convert a linear score to a probability, using the logisitic function `https://en.wikipedia.org/wiki/Logistic_function`. The apply a loss on the probability to match the class probabilities for the score. This latter approach has the advatange that the problem (EL-C) is smoothly differentiable.

Both these problems are usually presented in the two class case, and then generalized to multiple classes.
We will see that we can relate the two problems in the following sense: we can show that

(1) The loss in (CP) can be interpreted as a smoothed out margin loss, relating it to (DB)
(2) A version of the SVM model is invariant to relabeling the scores, so we can *calibrate* the scores to be probabilities. In other words we find a non-decreasing map $p(s)$ which gives the class probability as a function of the score. This shows that (DB) can be interpreted as finding class probabilities.

Going from the two class case to the multiclass case introduces an extra dimension to the problem: the number of classes. The multiclass case also allows us to impose extra structure on the loss. In particular, the multiclass version of (DB) and (CP) impose unusual (and different) notion of distances on the score: (DB) is an infinity norm, and (CP) uses the KL-divergence (a non-symmetric notion of distances on probabilities).

Another approach to the the multiclass case uses the 2-norm, which will be discussed in subsection 6.1.

2. Majority classifiers

We start with a very simple setting.

2.1. **Class probabilities in bins.**

Definition 2.1. Let $f : \mathscr{X} \to \mathbb{R}^d$ be a feature map. When the range of f is a discrete set with N elements,

$$f(x) \in \{f_1, \dots, f_N\}, \text{ for all } x \in \mathscr{X}$$

we say f is a binning map. In this context, we call the f_i bins. Define

(CP) $$p_i = \text{Prob}(y = 1 \mid f(x) = f_i), \quad \text{for } (x,y) \in S_m$$

to be the fraction of positive examples in bin i. Define the majority classifier to be the map $c : [0,1] \to \mathscr{Y}_{\pm}$ given by

$$c_{maj}(p) = \begin{cases} +1 & p \geq .5 \\ -1 & p < .5 \end{cases}$$

Define the bin hypothesis class to be functions $f : \mathscr{X} \to \mathscr{Y}_{\pm}$ which are constant on bins,

$$\mathscr{H}_{bin} = \{h(x) = h(f(x))\}$$

Given the dataset (S_m), we can define a simple classifier by majority rule.

Majority rule classifier algorithm

Inputs:

- Dataset (S_m). $y \in \mathscr{Y}_{\pm}$ (i.e. Binary classification)
- Binning map $f : \mathscr{X} \to [f_1, \ldots, f_N]$ which maps data, x, one of N bins

Goal:

- A simple rule for classification on each bin

Method:

- Find the p_i, the fraction of positive labels in each bin
- Set $c(x) = c_{maj}(p_{f(x)})$ to be the majority classifer for the bin.

Example 2.2. Consider the example of Figure 1. We consider (EL-C) with $\ell_{0,1}$ and the bin hypothesis class

$$c(x) = c_{f(x)}, \quad f(x) = 1,2,3,4$$

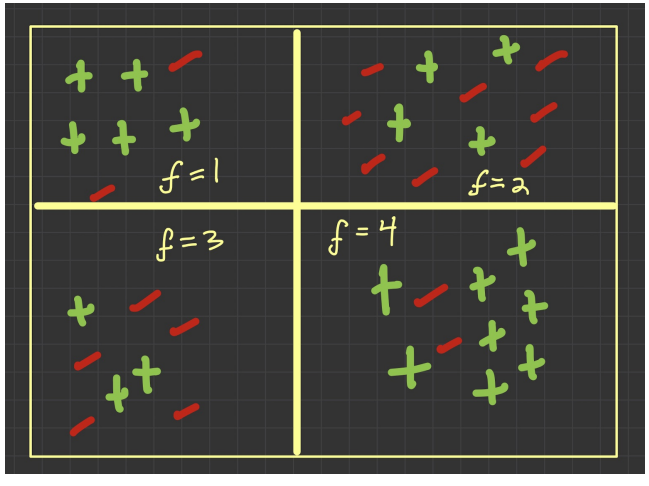

FIGURE 1. Illustration of majority rule classification problem. In this case there are four bins, corresponding to the values of $f = 1,2,3,4$. In each bin, the class is indicated by color and symbol (green = +1, red = -1). The $f = 1$ bin has 5 positive class elements and two negative class elements. The bin probabilities are $p_1 = 2/7, p_2 = 4/12, p_3 = 3/7, p_4 = 8/10$.

So the model is constant on each of the four bins. Majority rule for each bin corresponds to

$$c(x) = \begin{cases} +1 \text{ (green)}, & f(x) = 1,4 \\ -1 \text{ (red)}, & f(x) = 2,3 \end{cases}$$

The the zero-one loss error count is 30, with 24 false positives and 6 false negatives.

2.2. **Loss minimization for majority rule.** Here we show that we can obtain the majority rule classifier by loss minimization using the simplest classification loss, the 0-1 loss.

$$\ell_{0,1}(c,y) = \begin{cases} 0 & c = y \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Theorem 2.3. *Consider* (EL-C) *with the zero-one loss* (1). *Given a binning map, f, and the corresponding binning hypothesis class, $\mathscr{H}_{bin}$. Then the majority rule classifier is the minimizer of*

$$\min_{h \in \mathscr{H}_{bin}} \frac{1}{m} \sum_{i=1}^{m} \ell_{0-1}(h(f(x_i)), y_i) \tag{2}$$

Proof. We can separate the problem into each bin

$$L(h) = \frac{1}{m} \sum_j L(c_j)$$

where

$$L(c_j) = \frac{1}{m} \sum_{f(x_i)=j} \ell_{0-1}(c_j, y_i)$$

The last sum is given by

$$L(c_j) = p_j \ell(c_j, +1) + (1 - p_j)\ell(c_j, -1)$$

where we have summed over the positive and negative labels, and used (CP).

Thus we want to

$$\min_{c_j = 1, -1} L(c_j)$$

There are two cases to check, and clearly the minimum is when we choose the label with more examples, which corresponds to the majority classifier. □

Majority rule classifier via loss minimization

Inputs:

- Dataset (S_m). $y \in \mathscr{Y}_{\pm}$ (i.e. Binary classification)
- Binning map $f : \mathscr{X} \to [f_1, \ldots, f_N]$ which maps data, x, one of N bins

Model:

- hypothesis class of models $c(x) = c(f(x))$, which are constant on each feature.

Goal:

- Minimizer the errors using a piece-wise constant model.

Loss minimization method:

- Minimize (2) over the hypothesis class. The minimizer $c^*(x)$ corresponds to majority rule.

2.3. **Cost-sensitive classification.** In the binary classification case, we may have a preference for different error types. We define the error types as follows.

Definition 2.4. Given $c, y \in \mathscr{Y}_{\pm}$, where c is the model class prediction and y is the true label. Define:

- False positive when $c = 1$ and $y = -1$,
- False Negative corresponds to $c = -1$, $y = +1$.

We also use the terms true positive and true negative in the other cases.
The odds ratio is a useful quantity when comparing probabilities.

Definition 2.5. Given $p \in (0,1)$ the odds ratio is given by

$$r(p) = \frac{p}{1-p}$$

Exercise 2.6. *Show that the odds ratio function $r(p)$ is invertible on $\mathbb{R}^+$ with inverse $p(r) = r/(r+1)$.*

Example 2.7. Suppose we are classifying a disease, and a false negative is considered to be 10 times worse than a false positive. Then, in other words, we want the classification threshold to be when the odds ratio $r(p) = 10$, or when the probability $p(r) = 10/11$.

Define the odds ratio classifier as

$$c_{odds}(p, r_0) = \begin{cases} +1 & r(p) \geq r_0 \\ -1 & ow \end{cases}, \quad \text{where } r(p) = \frac{p}{1-p} \tag{3}$$

We summarize as follows.

Odds ratio classifier

Inputs:

- Dataset (S_m). $y \in \mathscr{Y}_{\pm}$ (i.e. Binary classification)
- Binning map $f : \mathscr{X} \to [f_1, \ldots, f_N]$ which maps data, x, one of N bins

Goal:

- Find the a simple classifier on each bin which depens on the cost of error types

Method:

- Find the p_i, the fraction of positive labels in each bin
- Set $c(x) = c_{odds}(p_{f(x)})$ to be the odds ratio classifer for the bin.

Exercise 2.8. *For each of the bins in Figure 1, determine the odds ratio $p/(1-p)$. Determine the smallest value of r needed to make: (i) all the bins positive (ii) two of the bins positive (iii) only one bin positive.*

Exercise 2.9. *Suppose it costs \$2 to make a bet which pays \$25 dollar if you win. We say the odds are $r = 25/2$. Find the probability p of the best which means, on average, making the bet many times, you will break even. (Hint: relate this to the function $p(r)$.)*

Conversely, suppose you believe that you have a 1 in 3 chance of winning a bet. What would be the corresponding fair odds for the bet?

2.4. **Loss minimization for odds ratio classifier.** Let r_0 be the desired odds ratio (for false negative versus false positive). Here we show that the odds ratio classifier can be obtained using minimization of (EL-C), for the following loss.

Define the cost-sensitive loss to be

$$\ell_{0-r}(c,y) = \begin{cases} r & c = 1, y = -1 \\ 1 & c = -1, y = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Theorem 2.10. *Consider the loss* (4) *in the context of a binning map and the bin hypothesis class defined in Definition 2.1. Let p_i be the fractions of positive examples, given by* (CP)*. Conisder the cost-sensitive loss* (4)*. Then the problem* (EL-C) *in this context corresponds to*

$$\min_{h \in \mathscr{H}_{bin}} \frac{1}{m} \sum_{i=1}^{m} \ell_{0-r}(h(f(x_i)), y_i) \tag{5}$$

The minimizer is given by the odds ratio classifier (3) *on each bin*

$$c(x) = c_{odds}(p_{f(x)}, r_0)$$

Proof. The key steps in the proof:

1. The first step is the same as in Theorem 2.3. We can separate the sum in (5) into each bin.
2. The second step is to establish, in a given bin with bin probability p_i, that the loss minimizer is given by (3).

Fixing a bin, $f(x) = i$, and summing those terms in (5) with $f(x) = i$, we get

$$L(c_i) = \min_{c_i} p_i \ell(c_i, +1) + (1 - p_i)\ell(c_i, -1)$$

There are two cases to check, which corresponds to $c_i = 1, -1$. Thus we get

$$\min((1 - p_i)r, p_i)$$

Which are equal when $r(p_i) = p_i/(1 - p_i) = r$. Thus if $r(p_i) \geq r$ we should set $c_i = 1$ (otherwise set it to be -1. This corresponds to the odds ratio classifier (3), as desired. □

2.5. **Loss design to learn a fraction from examples.** Reference: [GR07].

In this section we solve a simple problem. We will need it in the next section.

Loss for probabilities: Problem definition

Inputs:

- A data set of the form (6)

$$S_m = \{y_1, \ldots, y_m\}, \quad \text{where} y_i \in \mathscr{Y}_\pm = \{-1, +1\} \tag{6}$$

- Define $q(S_m)$ to be the fraction of $y_i = 1$ in S_m.

Goal:

- Use differentiable loss minimization to find $q(S_m)$.

Definition 2.11 (proper losses for learning probabilities)**.** Given a loss of the form

$$\ell_{\text{Prob}} : [0, 1] \times \mathscr{Y}_\pm \to \mathbb{R}^+$$

and a dataset S_m define

$$\widehat{L}(p) = \frac{1}{m}\sum_{i=1}^{m} \ell_{\text{Prob}}(p, y_i) \tag{EL-P}$$

The loss is *proper* (in the sense of [GR07]) if for every S_m of the form (6),

$$\min_{p\in[0,1]} \widehat{L}(p) = q(S_m)$$

otherwise the loss is improper.

We consider the following losses. To simplify notation, write $y^+ = \max(y, 0)$

$$\ell_2(p, y) = (p - y^+)^2/2$$

and

$$\ell_1(p, y) = |p - y^+|$$

and

$$\ell_{\log}(p, y) = \begin{cases} -\log(p) & y = 1 \\ -\log(1-p) & y = -1 \end{cases}$$

Theorem 2.12. *The losses ℓ_2 and $\ell_{\log}$ are proper. The loss ℓ_1 is not.*

Proof. Step 1. As in previous proofs, we consider (EL-P) and collect terms, breaking the sum into two parts, depending on the value of y.

$$m\widehat{L}(p) = \sum_{i=1}^{m} \ell(p, y_i) = \sum_{y_i=1} \ell(p, 1) + \sum_{y_i=-1} \ell(p, -1)$$

collect terms

$$m\widehat{L}(p) = q\ell(p, 1) + (1-q)\ell(p, -1)$$

Step 2. For each choice of the loss, we minimize the last equation.
Verify these statements.

$$\min_p q(1-p)^2 + (1-q)(p)^2$$

gives $p = q$.

$$\min_p q \log p + (1-q)\log(1-p)$$

gives $p = q$
But

$$\min_{p\in[0,1]} q(1-p) + (1-q)p$$

has gives $p = 0$ or $p = 1$ as minimizer □

Exercise 2.13. *Fill in details of proof*

Exercise 2.14. *TODO: give other examples of proper loss and prove they are proper. Consider the spherical loss*

$$\ell_s(p,y) = \begin{cases} p(1-2p+p^2)^{-1/2} & y = +1 \\ (1-p)(1-2p+p^2)^{-1/2} & y = -1 \end{cases}$$

Determine if it is proper.
Rewrite as $p_i/\|(p_1,p_2)\|$

Loss for probabilities: solution summary

Inputs:

- A data set of the form (6)

Goal:

- Find $q = q(S_m)$, the fraction of $y_j = 1$ using differentiable loss minimization.

Solution:

- minimize (EL-P) using the loss ℓ_2 or the loss $\ell_{\log}$.

2.6. **Differentiable loss minimization for majority rule.** So far we have used the zero-one loss (and the cost-sensitive loss) to obtain bin-based classifiers. We showed they could be obtained by minimizing an expected loss. However, the loss minimization was direct - we did not differentiate the loss.

When we want to build more sophisticated models, we will need to differentiate the loss. So next, we will try to recover those classifiers using a differentiable loss.

Next we show that we can learn the bin probabilities (CP) using a differentiable loss.

Definition 2.15. Our models are probability valued, and constant on each bin,

$$\mathscr{H} = \{h : \mathscr{X} \to [0,1] \mid h(x) = p_{f(x)},\} \tag{7}$$

Here $(p_1, \ldots, p_N)$ is a vector of bin probabilities.

Majority rule classifier via differentiable loss minimization: setup

Inputs:

- Dataset (S_m). $y \in \mathscr{Y}_{\pm}$ (i.e. Binary classification)
- Binning map $f : \mathscr{X} \to [f_1, \ldots, f_N]$ which maps data, x, to one of N bins

Goal:

- Find the bin probabilities using a differentiable loss.
- Given the bin probabilities $p(x)$, define the class by majority, $c(p(x)) = c_{maj}(p(x))$

Model:

- Hypothesis class of models given by (7) which are constant on each bin.

Method:

- Minimize a loss of the form (EL-C) over the hypothesis class, using a suitable loss.

Note, in the above, could also use the odds ratio classifier if we have a cost-sensitive loss.

3. Learning class probabilities

In this section we study how to learn the class probabilities using a linear model.

3.1. **Logistic function.** First, we need a function which converts number $x \in \mathbb{R}$ to probabilies $p(x) \in [0,1]$. In this context, the number are called *logits*.

The function is called the logistic function

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Plot looks like Figure 2.

3.1.1. *Properties of the logistic function.* These logistic function has an inverse which maps probabilities to numbers (logits)

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Here write $f(x) = \sigma(x)$.

- $2f(x) = 1 + tanh(x/2)$
- $f(x) = \frac{\exp x}{1+\exp x}$
- $1 - f(x) = f(-x)$ (so $f(x) - 1/2$ is an odd function)
- $f'(x) = f(x)(1-f(x))$

The ratio $\frac{p}{1-p}$ is called the *odds ratio*.

Now consider a loss which has the same shape as the hinge, but is strongly convex, and smooth.

$$\ell_{sc}(s,y) = -\log\left(\frac{1}{1+\exp(-x)}\right)$$

3.2. **TODO: put in the linear model.** In this section : we have features. We add a linear model on features. We estimate the probability of the class, as a function of the features. The bins go away: instead we have the score function.

So this is how we combine it. Could do: make bins from the score.

Explain: better score will better separeate the prob.

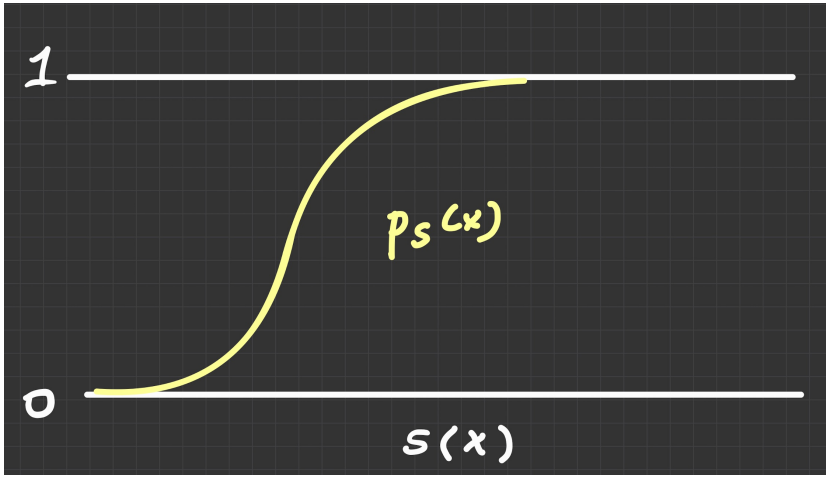

FIGURE 2. Illustration of a score function

4. SCORE BASED LOSSES

In this section, we go to the primary focus, which is score-based losses. Later we will consider linear models to obtain a score from features. But first we formalize our intuition about what we mean by a score.

4.1. **Define a score function.** Intuitively, a score function (for a given class) means that a higher score corresponds to a higher likelihood of class membership. We formalize this notion with the following definition.

Define the probability density of a function $s(x)$ for the class k to be

$$p_s(t) = Prob(y = k \mid s(x) = t)$$

Definition 4.1. We say $s(x)$ is a score function for the class k if $p_s(t)$ is a non-decreasing function. We say the score function $s(x)$ is *separatating* if $p_s(t)$ only takes the values $0, 1$.

Given a score function, and a threshold, w, define the threshold model and threshold classifier, repspectively, by

$$h_w(s) = w - s \qquad c_w(s) = \operatorname{sgn}(h_w(s)) \tag{8}$$

Exercise 4.2. *In the binary classification case $Y_{\pm}$, show that the 0-1 loss with the threshold classifier* (8) *becomes a step function*

$$\ell_{0-1}(c_w(s), y) = 1_{\{\operatorname{sgn}(w-s)=y\}}$$

Exercise 4.3 (check definition)**.** *Show that if s is a separating score function, the, either (i) there is exact one w for which the threshold classifier has zero error, or (ii) there is half-closed interval I (check endpoints) for which c_w has zero error, for all $w \in I$. Here the error means the 0-1 loss.*

A more typical score function looks like Figure 2.

Exercise 4.4 (Relate score classifier to Bayes classifier)**.** *For a score-based classifier, choosing a threshold which is the solution of*

$$p_s(x) = .5$$

results in the Bayes classifier, defined above.

Prove the statement above. Hint: every lower score has a probabilty of less than .5, and similarly for the higher score.

4.2. **Classes from score functions.** Although eventually we will need to study how to *learn* score functions, first we need to study how to (best) convert scores into classes. These are two difference problems because:

- we can have very effective scoring function, which means high scores are more likely to be in the class
- however, he best way to classify using the score will also depend on the distribution of classes (as well as our preference for error types).

There is more than one way to define a classification based on scores.

Example 4.5 (Grading). Consider the classification problem of converting a grade, $x \in [0,100]$ in one of $K = 5$ letter grades F, D, B, C, A. We can use an absolute rule, e.g. $x \in [85,100]$ converts to A, or we can grade on the curve: which means having a fixed percentage of the students in each grade.

In each case, the outcomes are difference and there are arguments for and against each method. For example, if a class is particularly strong compared to other classes, the students are penalized by grading on a curve.

In what follows, we will define the classifier based minimizing (EL-C) using a choice of loss function (and classifier). We will then need to study loss design: how the choice of loss affects the solutions of (EL-C).

4.3. **Score-based loss.** Now we want to define a score-based loss which is piecewise differentiable as a function of s.

Define the absolute error loss $\ell_{abs}(s,y)$ as

(LAC)
$$\ell_{abs}(s,y) = \begin{cases} 0 & \operatorname{sgn}(s) = y \\ |y-s| & ow \end{cases}$$

Example 4.6. Consider the example of Figure 1. Find the minimizer of (EL-C) with the score-based threshold classifier (8), and the absolute error loss (LAC). Show that it corresponds any choice of w between 1 to 2. (TODO check endpoints), and

$$c(s) = \begin{cases} 0 & s = 1 \\ 1, & s = 2,3,4 \end{cases}$$

Note, this is different from the Bayes classifier.

Exercise 4.7. *Show that in Figure 1, if we relabel the scores from* $1,2,3,4$ *to any other non-decreasing values (e.g. try* $10,15,20,25$*), we get the same classifier. (Hint: can check this directly or use the condition for a minimizer).*

Exercise 4.8. *Relate the problelm* (EL-C) *with the score-based threshold classifier* (8)*, and the absolute error loss* (LAC) *to the central value problem with the absolute value loss (from the regression chapter). Explain!*

This loss is amenable to optimization.

Theorem 4.9. *Consider* (EL-C) *with the score-based threshold classifier* (8)*, and the absolute error loss* (LAC)*. A sufficient condition for a minimizer w^* is that the number of false positives is equal to the number of false negatives.*

Proof. Now consider mimizing (EL-C) with this loss
We get $\pm$ on each error, depending on if PF / FN
[[details in handwritten class notes, to be filled in]]

$$\sum_{FP} 1 = \sum_{FN} 1$$

So the w^* is the threshold which makes $FP = FN$. □

More generally, we can choose the ratio of false positives to false negatives using the the following generalization of the absolute error loss

(LAC-FP)
$$\ell_{abs}(s,y) = \begin{cases} 0 & \operatorname{sgn}(s) = y \\ |y-s| & y = 1, \operatorname{sgn}(s) = -1 \\ C|y-s| & y = -1, \operatorname{sgn}(s) = +1 \end{cases}$$

Exercise 4.10. *Generalize Theorem 4.9 to the case of* (LAC-FP)*. Find the value of C which leads to $FP = 10FN$*

4.4. **Margin-based Losses.** We say an interesting property of the absolute error loss: an increasing relabelling the scores did not change the result. This suggests that the loss may not encourage a wider margin. Later, when we are learnig scores, we want to encourage scores which better separate classes. One approach to this problme is to design a loss with a *margin*

4.5. **Binary classification, margin (hinge) loss.** The idea is to have a penalty for correct classifications, if the distance to the classification boundary is less than one (before, in the absolute loss, it was zero). See Figure 3, for an illustration.

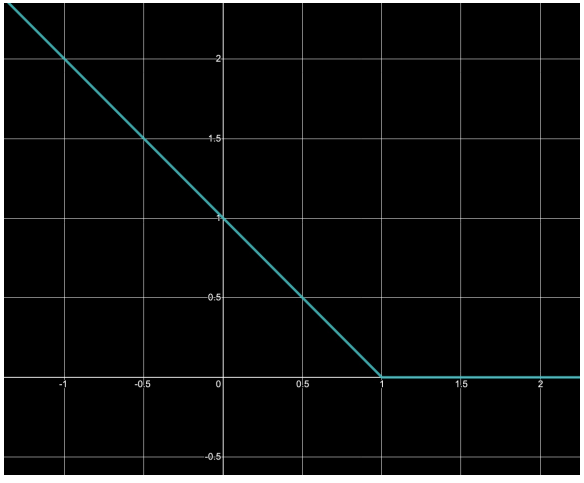

FIGURE 3. Hinge loss, this loss is differentiable except at corner, and lies above the 0-1 loss

So define for $y \in \mathscr{Y}_{\pm}, s \in \mathbb{R}$,

$$g(s,y) = sy$$

Note that $g(s,y)$ is the signed distance to the classification boundary: correct if $g > 0$, incorrect if $g < 0$.

Next define the hinge loss

$$\ell_{hinge}(s,y) = \max(1 - g(s,y), 0) = (1 - g(s,y))^{+}$$

Note

$$\partial_s \ell_{hinge}(s,y) = \begin{cases} +1 & y = 1, s < 1 \\ -1 & y = -1, s > -1 \end{cases}$$

Note, also that,

$$\ell_{hinge}(s,y) \geq \ell_{0-1}(s,y)$$

with equality when $g \geq 1$ and when $g = 0$.

5. Discussion of classification losses

So far we have studied losses for classication

- zero-one loss
- absolute distance loss
- margin loss

Each of these assign a different interpretation to the scores

- zero-one loss: scores have an identity, but are not comparable
- absolute distance loss: scores are ordered, but have no scale
- margin loss: scores are ordered. There is a scale for scores within distance one of the boundary, but nothing beyond. (The loss sees no difference between 1 from boundary and 3 from boundary). This is because the loss is flat (zero) away from the boundary.

Next we will look at losses that are strongly convex - so they continue to give (diminishing) returns the further you go from the boundary.

6. Classification with more than two classes

6.1. **Multiclass classification using cosine similarity.** Another approach to the the multiclass case uses the 2-norm. While is is more naturally geometrically, it is less common. The similarity approach which measures the angle between two feature (score) vectors, is used in Face Recognition and Image Search. The FR problem corresponds to a binary classification: determine if two images represent the same face (for a face/images never seen before). The Image Search

problem is: given an image, and a database of images (e.g. the internet), find similar (looking) images. Both compare the cosine similarity

$$\mathrm{sim}(x_1, x_2) = \frac{s_1 \cdot s_2}{\|s_1\|_2 \|s_2\|_2}$$

When the vectors are close to unit length, this similarity is comparable to

$$\mathrm{sim}(x_1, x_2) = 1 - \frac{1}{2}\|x_1 - x_2\|^2, \quad \text{when } \|s_1\| = \|s_2\| = 1$$

(just expand the squared term).

Now we define a classifier by the highest score. Let's assume we have score functions for each classs.

For $s \in \mathbb{R}^K$, define $c : \mathbb{R}^K \to \mathscr{Y}_k$ by

$$c(s) = \arg\max s_y$$

Define the gap (or margin) in the multi-class case by

$$g(s, y) = s_y - \max_{j \neq y} s_j$$

so this is positive if the model is correct, and negative otherwise. As in the binary case, we can regard g as the signed distance (in the maximum norm) to teh classification boundary.

Now we can define, as in the binary case,

$$\ell_{abs}(s, y) = \max(-g(s, y), 0)$$

which is a penalty for the distance to the classification boundary (measured in the maximum norm).

Likewise, if we want to encourage a margin, we define

$$\ell_{hinge}(s, y) = \max(1 - g(s, y), 0) = (1 - g(s, y))^+$$

which penalizes correct points within distance one of the classification boundary.

Exercise 6.1. *Show the multi-class case with $K = 2$ case reduces the binary case, when we set*

$$(s_1, s_2) = (s/2, -s/2)$$

where LHS is the multiclass score function when $K = 2$, and s is the binary score function.

7. Smooth classification losses

We can define smooth classification losses, which are smooth approximations of the margin loss.

8. Linear models

However, in general data is multi-dimensional, and what we want to learn is a consistent score function.

We will do this, in the multi-class case, by learning a score function for each class.

Goal: learn $h(x; w_y)$ a score function for each class. (I.e. w_y is matrix, one row for each y, each row is a weight vector) (Note the features share lots of information, but the final classifier is one for each class) So K functions, one for each class, and K vectors w_y.

$$h_W(x) = (w_1 \cdot x, \ldots, w_K \cdot x)$$

Result / Exercise:

Compute $\nabla_{w_i} \ell_{hinge}(h_W(x), y)$

Complicated: $\ell' = -1$ if $g(s, y) < 1$ and 0 ow. Gradient: x_y for w_y and $-x_j$ for the maximum component. and 0 ow.

References

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

[GR07] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[Mur22] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2022.