MATH 462 LECTURE NOTES

ADAM M. OBERMAN

CONTENTS

Date: September 24, 2021.

1. Introduction to binary classification

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

1.1. Binary classification.

In the general classification problem, the target set $\mathscr{Y}$ is a set of discrete labels. Here we consider the case of binary classification consisting, so there are two labels, which we denote by $-1, +1$, and we write

$$\mathscr{Y} = \mathscr{Y}_{\pm} = \{-1, +1\}$$

We are given a dataset (S_m) consisting of m pairs of (x_i, y_i), $i = 1, \dots, m$, of data, $x_i \in \mathscr{X}$ and labels, $y_i \in \mathscr{Y}$,

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

- Hypotheses consisting of a parameterized family of models, $h_w : \mathscr{X} \to \mathbb{R}^K$.

The most important loss for classification is the zero-one loss,

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

For a given classification function, $c(x)$ we define the (average) error on (S_m), by

$$\widehat{L}_{0-1}(c) = \frac{1}{m} \sum_{i=1}^{m} \ell_{0-1}(c(x_i), y_i) \tag{EE}$$

However, this loss is not amenable to optimization, so usually we optimize our models with a different, differentiable, loss ℓ_{class} in (EL-C), which will be defined and study below. This classification loss should be (piece-wise) differentiable as a function of h. There is an added complication that our classification losses will be defined on the pair (s,c) for $s \in \mathbb{R}, c \in \mathscr{Y}$.

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

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

However, there is something new here that we don't see in the case of regression (where $\mathscr{Y} = \mathbb{R}$). We need to check that our loss minimization (which is defined $h \in \mathbb{R}$) results in an effective *classification*. In other words we care about the average classification error (the 0-1 loss, defined below).

1.2. **Binary classification approaches.** The main approaches to (supervised) binary classification we study are:

(Vote) Bin the classes and use a majority classifier in each bin. (Plus a more general version of this). The binning can happen in various ways (e.g. cluster), in our context, it is a black box function.

(Score) Score based. Learn a score function from features. The class is determined by the sign of the score. The score can be Support Vector Machines, [DFO20].

(Prob) Probability based. Learn a probability function from features. The class is determined by the probability. In this case, we convert a linear model to a probability, using the logisitic function `https://en.wikipedia.org/wiki/Logistic_function`.

The following framework applied to both (Score) and (Prob).

Binary Classification via loss minimization (abstract)

Inputs:

- Dataset (S_m) with $y_i \in \mathscr{Y}_{\pm} = \{-1, +1\}$ and $x_i \in \mathbb{R}^d$ features.
- Hypothesis class of models $h_w(x) : \mathbb{R}^d \to \mathbb{R}$, parameterized by w.
- Classifier: $c : \mathbb{R} \to \mathscr{Y}_{\pm}$, a rule which converts the model value to a class.
- Classification loss (ℓ_{class}).

Goal:

- Given x, predict $y = c(h_w(x))$

Method:

- Find the parameterized model which best fits the data by minimizing (EL-C).

Reality check: did it work?

Inputs:

- Dataset (S_m) with $y_i \in \mathscr{Y}_\pm = \{-1, +1\}$ and $x_i \in \mathbb{R}^d$ features.
- Model h_{w^*} which minimizes the empirical loss (EL-C).
- Classifier: $c : \mathbb{R} \to \mathscr{Y}_\pm$, a rule which converts the model value to a class.

Goal:

- Estimate the model error, $\widehat{L}_{0-1}(c(h_w^*))$, given by (EE).

In other words, (even though we called ℓ_{class} a classification loss), what can we say about the error of the classifier. We need to characterize the effect of the loss on the errors. There are two approaches to this question: (i) we can prove *a priori* the properties of the classifier, or (ii) we can simply check *a posteriori* that the results are good. The purpose of understanding classification losses is to achieve the first goal.

2. Majority classifiers

We start with a very simple setting. In this setting, we assume that we are given a binning map, which puts similar samples x into the same bins. For now, this is a black box map, in the sense that we do not concern ourselves with how it was obtained. Refer to Figure 1.

2.1. **Class probabilities in bins.**

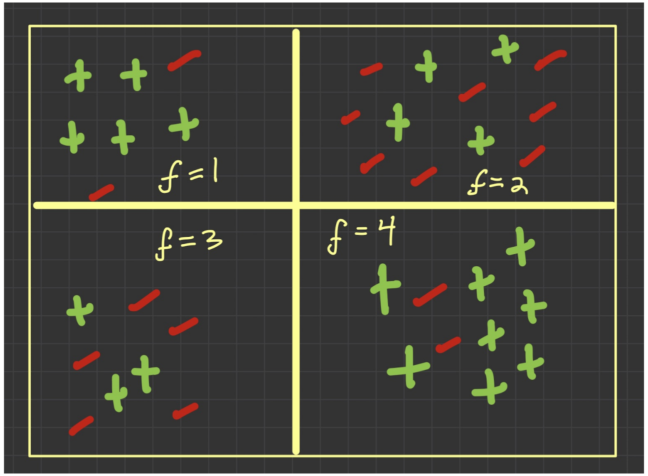

FIGURE 1. Illustration of majority rule classification problem. In this case there are four bins, indicated by yellow squares, corresponding to the values of $f = 1,2,3,4$. In each bin, the data is represented by a symbol: + (green +) for the class $y = +1$ and - (red -) for the class $y = -1$. For example, the $f = 1$ bin has 5 positive class elements and two negative class elements. We use p_i to represent the fraction of positive examples in each bin (CP). In this example

$$p_1 = 5/7, \quad p_2 = 4/12, \quad p_3 = 3/8, \quad p_4 = 8/10.$$

Definition 2.1. Let $f : \mathscr{X} \to \{1,2,\ldots,N\}$ be a discrete valued function. The set

$$B_j = \{x \in \mathscr{X} \mid f(x) = j\}$$

form a partition of $\mathscr{X}$, which we call bins, we call f a binning map. Given (S_m), define

$$\text{(CP)} \qquad p_j = \frac{|\{y = +1 \mid x \in B_j\}|}{|B_j|} \quad \text{for } (x,y) \in S_m$$

to be the fraction of positive examples from the dataset in bin j.

Define the bin hypothesis class to be functions $c : \mathscr{X} \to \mathscr{Y}_\pm$ which are constant on bins, parameterized by $w \in \mathscr{Y}^N$,

$$\mathscr{H}_{bin} = \{c_w(x) \mid c_w(x) = w_{f(x)},\ w_j \in \{-1,+1\}\}$$

Definition 2.2. Define the majority classifier to be the map $c : [0,1] \to \mathscr{Y}_\pm$ given by

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

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 $f : \mathscr{X} \to [1, \dots, N]$ 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 classifier for the bin.

Example 2.3. In Figure 1, setting $w = (-1, +1, +1, +1)$ results in $h_w(x) = -1$ if $f(x) = 1$, in other words, if x is in the first bin, and $h_w(x) = +1$ otherwise.

Example 2.4. Consider the example of Figure 1. We consider (EE) with the bin hypothesis class. 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 bin error count is: 2, 4, 3, 2, for bins 1, 2, 3, 4, respectively.

2.2. **Loss minimization for majority rule.** Here we show that we can obtain the majority rule classifier by loss minimization using the 0-1 loss ($\ell_{0,1}$)

Theorem 2.5. *Consider* (EL-C) *with the zero-one loss* ($\ell_{0,1}$). *Given a binning map, f, the majority rule classifier is the minimizer of* (EE) *over the binning hypothesis class, $\mathscr{H}_{bin}$.*

$$\min_{c_w \in \mathscr{H}_{bin}} \widehat{L}_{0-1}(c_w)$$

Proof. Write

$$\widehat{L}_{0-1}(c_w) = \frac{1}{m}\sum_{i=1}^{m} \ell_{0-1}(w_{f(x_i)}, y_i) \tag{EL-01}$$

We can separate the problem into each bin

$$\widehat{L}_{0-1}(c_w) = \frac{1}{m}\sum_{j} \widehat{L}_j(c_w)$$

where

$$\widehat{L}_j(c_w) = \frac{1}{m}\sum_{(x,y)\in B_j} \ell_{0-1}(w_j, y)$$

Using the definition (CP), the last sum is given by

$$\widehat{L}_j(c_w) = p_j \ell_{0-1}(w_j, +1) + (1-p_j)\ell_{0-1}(w_j, -1)$$

where we have summed over the positive and negative labels. Thus we want to

$$\min_{w_j \in \mathscr{Y}_\pm} L_j(c_w)$$

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. □

Binary classifier via loss minimization (binning approach)

Inputs:

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

Model:

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

Goal:

- Given x, find $y(x)$.

Loss minimization method:

- Minimize the expected zero-one loss (EL-01) over the hypothesis class. The minimizer $c^*(x)$ corresponds to majority rule.

2.3. **Important function : the odds ratio.** The odds ratio is a useful quantity when comparing probabilities.

Definition 2.6. Given $p \in (0,1)$ the odds ratio is $r(p) = \frac{p}{1-p}$. Given $r \in \mathbb{R}^+$, define $p(r) = r/(r+1)$.

Exercise 2.7. *Show that the odds ratio function $r(p)$ is invertible on $\mathbb{R}^+$ with inverse $p(r) = r/(r+1)$.*

Important function: odds ratio

- The odds ratio $r(p) = p/(1-p)$ maps probabilities to numbers.
- The inverse of the odds ratio is $p(r) = r/(r+1)$.

Example 2.8. 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 $p(r) = 10/11$.

Exercise 2.9 (TODO: better wording)**.** *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 bet 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. **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 2.10. 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.
Define the odds ratio classifier as

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

We summarize as follows.

Odds ratio classifier

Inputs:

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

Goal:

- A simple classifier on each bin which depends 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 classifier for the bin.

Exercise 2.11. *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.*

2.5. **Loss minimization for odds ratio classifier.** Let r be a given odds ratio. Here we show that the odds ratio classifier can be obtained using minimization of the form (EL-C), for the following loss.

Define the cost-sensitive binary classification loss to be

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

Then (EL-C) with the cost-sensitive loss (ℓ_{0-r}) can be written

$$\widehat{L}_{0-r}(c) = \frac{1}{m}\sum_{i=1}^{m} \ell_{0-r}(c(x_i), y_i) \tag{EL-0r}$$

Define the bin odds ratio classifier (1) by

$$c^*(x) = c_{odds}(p_{f(x)}, r)$$

Theorem 2.12. *Consider the loss* (ℓ_{0-r}) *in the context of a binning map and the bin hypothesis class defined above. Let p_j be the fraction of positive examples, in bin j, given by* (CP). *Then*

$$\arg\min_{c_w \in \mathcal{H}_{bin}} \widehat{L}_{0-r}(c) = c^*$$

Proof. The key steps in the proof:

1. The first step is the same as in Theorem 2.5. We can separate the sum in (EL-0r) 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 (1). Following the same steps as in Theorem 2.5, fixing a bin, B_j, and summing those terms in (EL-0r), we arrive at

$$\widehat{L}_j(c) = p_i \ell_{0-r}(c_j, +1) + (1 - p_i)\ell_{0-r}(c_j, -1)$$

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 (1), as desired. □

Odds ratio classifier via loss minimization

Inputs:

- Dataset (S_m). $y \in \mathcal{Y}_{\pm}$ (i.e. Binary classification)
- Binning map f which maps data, x, in bins.

Model:

- hypothesis class of models which are constant on each bin.

Goal:

- Find the classifier which minimizes the cost-sensitive errors over the hypothesis class.

Loss minimization method:

- Minimize the expected loss (EL-0r) over the hypothesis class. The minimizer $c^*(x)$ corresponds to majority rule.

3. Proper Scoring rules: how to learn a probability of an event

Proper scoring rules are a type of loss used to learn the probability of an event. Their use predates machine learning: they were used to score forecasters for sporting events and weather. So the terminology is different from ML terminology.

Reference: [GR07].

3.1. **Loss design to learn a probability from samples.** Here we consider the problem

Loss for probabilities: Problem definition

Inputs:

- A data set of the form (2)

$$S_m = \{y_1, \dots, y_m\}, \qquad \text{where } y_i \in \mathscr{Y}_\pm = \{-1, +1\} \tag{2}$$

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

Goal:

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

Although this seems like a simple problem, we will make a definition for the type of losses which work.

Definition 3.1 (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 (2),

$$\operatorname*{arg\,min}_{p\in[0,1]} \widehat{L}(p) = q(S_m)$$

otherwise the loss is improper.

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

Definition 3.2 (Candidates)**.**

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

and

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

and

$$\ell_{\log}(p,y) = \begin{cases} -\log(p) & y=1 \\ -\log(1-p) & y=-1 \end{cases} \tag{3}$$

Theorem 3.3. *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.

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

collect terms

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

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 3.4. *Fill in details of proof*

Loss for probabilities: solution summary

Inputs:

- A data set of the form (2)

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 $\ell_{\log}$.

Warning: not all losses work

- Using ℓ_1 in the problem above will not work.

Exercise 3.5. *Show that using using ℓ_1 in the problem above gives $p^* = round(q)$ instead of q.*

4. Learning class probabilities

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

Binary Classification via loss minimization (Probability approach)

Inputs:

- Dataset (S_m) with $y_i \in \mathscr{Y}_\pm = \{-1,+1\}$ and $x_i \in \mathbb{R}^d$ features.
- Hypothesis class of models $p_w(x) : \mathbb{R}^d \to [0,1]$, parameterized by w.

Goal:

- Given x, find $c(x)$

Output:

- The model $p_{w^*}(x)$ to $y = c_{maj}(p_{w^*}(x))$

Method:

- Find w^* by minimizing (EL-PC).

4.1. **Important function : the logistic function.**

Definition 4.1. The logistic function $\sigma : \mathbb{R} \to (0,1)$

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

maps numbers to probabilities. Define

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

In this context, the numbers coming from a probability are called *logits*. See Figure 3 for a sketch of σ.

Exercise 4.2. *Show that* σ *and* logit *are inverses. Hint: use the facts (i) that the odds ratio,* $r(p)$ *and* $p(r) = r/(r+1)$ *are inverses, (ii)* exp *and* log *are inverses, and write* $\sigma(x) = p(\exp(x))$*, and* $\operatorname{logit}(p) = \log(r(p))$.

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)
- $f'(x) = f(x)(1 - f(x))$

Important function: logistic

- The function $\sigma(x) = \frac{1}{1+\exp(-x)}$ maps $\mathbb{R}$ to $[0,1]$
- The inverse is given by $\operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$.

4.2. **Linear models.** In this setting, for $x \in \mathbb{R}^d$, we consider the linear model

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

We will further be composing the models with σ to obtain a probability

$$\mathscr{H}_{\text{Prob}} = \{p_w(x) \mid p_w(x) = \sigma(w \cdot x)\}$$

4.3. **Probability loss.** We can choose any 'proper' loss for the probability. For classification, the standard choice is the log loss. Thus we take

$$\ell_{\text{Prob}}(h, y) = \ell_{\log}(p, y)$$

given by (3). Then (EL-C) becomes

$$\widehat{L}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell_{\log}(p_w(x_i), y_i) \tag{EL-PC}$$

4.4. **Reality check: did it work.** Once we have the model, we need to we can say about the classifier. Does the probabilistic model give a good classifier?

Discussion

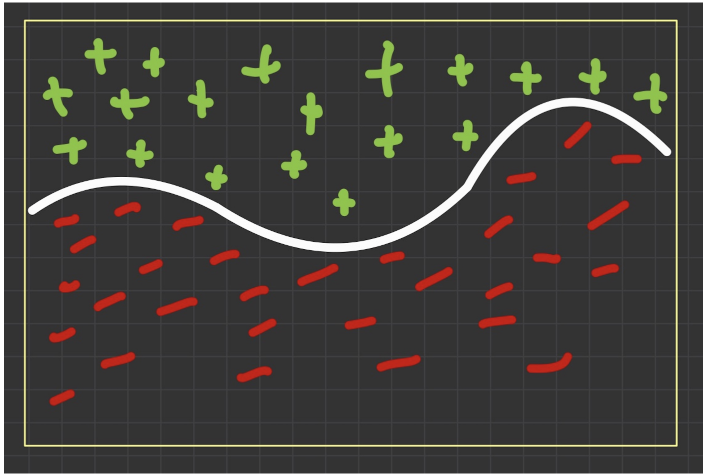

FIGURE 2. Classification problem: nonlinear feature map leads to a nonlinear boundary. However, the model $h_w(x) = w \cdot f(x)$ is still linear as a function of w. White line: the classification boundary $h_w(x) = 0$.

4.5. **Discussion: nonlinear features.** Note that this setting is quite general, since the notation obscures the fact that we include the case where there is a nonlinear feature map $f(x)$, and the model is $h_w(x) = w \cdot f(x) = \sum_{i=1}^{d} w_i f_i(x)$.

Example 4.3. We can have raw data $x \in \mathbb{R}$ and features $f_i(x) = x^i$, for $i = 1, \ldots, d$, which corresponds to fitting one dimensional data with a polynomial. Here we are looking for a nonlinear classification boundary. See Figure 2.

5. Score based losses

Binary Classification via loss minimization (Score based approach)

Inputs:

- Dataset (S_m) with $y_i \in \mathcal{Y}_\pm = \{-1,+1\}$ and $x_i \in \mathbb{R}^d$ features.
- Hypothesis class of score models $s_w(x) : \mathbb{R}^d \to \mathbb{R}$, parameterized by w.

Goal:

- Given x, find $c(x)$

Output:

- The model $s_{w^*}(x)$ maps to $y = \operatorname{sgn}(s_{w^*}(x))$

Method:

- Find w^* by minimizing (EL-C).

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

Example 5.1 (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.

5.2. **Linear models and features.** In this setting, for $x \in \mathbb{R}^d$, we consider the hypothesis class consisting of linear models

$$\mathscr{H}_{score} = \{s_w(x) \mid s_w(x) = w \cdot x + w_0\}$$

(Later can absorb w into features, but it's here for emphasis now).

Here $s_w(x)$ corresponds to the score of x. We want higher scores to corresponds to higher probability of correct classification, as in Figure 3.

Given the score s, we define the classifier

$$c(s) = \operatorname{sgn}(s)$$

Now we define a score-based loss which is piecewise differentiable as a function of s. We need to use a scoring loss.

5.3. **The standard score-based loss.** The standard score based loss is the following hinge (or margin) loss. To understand it, we need to define the following error types

Definition 5.2. Given a threshold $t \geq 0$. Let $s \in \mathbb{R}$ be a score, $c(s) = \operatorname{sgn}(s)$. Given $y \in \mathscr{Y}_{\pm}$, define the following score types

- incorrect: $c(s) \neq y$
- marginal: $c(s) = y$ and $|s| < t$
- confident: $c(s) = y$ and $|s| \geq t$

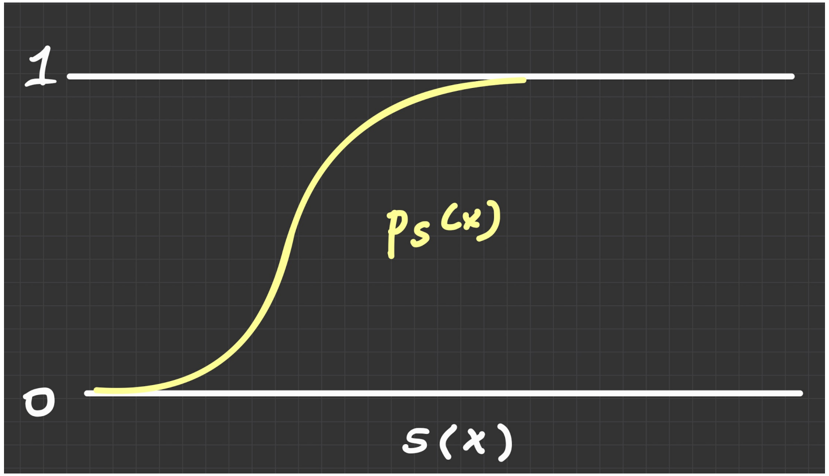

FIGURE 3. Illustration of a score function

Also use the terms marginal positive, marginal negative if correct but marginal. Define the t-margin loss, which penalizes both incorrect and marginal scores.

$$\ell_{margin,t}(s,y) = \begin{cases} 0 & sy \geq t \\ |s/t - 1| & 0 \leq sy \leq t \\ 1 + |s|/t & sy \leq 0 \end{cases} \tag{4}$$

Special cases: we say absolute value loss when $t = 0$ and standard margin loss when $t = 1$.

The cases above correspond to: correct, correct but marginal, incorrect.

Exercise 5.3. *Plot the loss for $y = 1$ and $t = 1$. Show symmetry of loss $\ell_{margin,t}(-s,-y) = \ell(s,y)$. Use this to plot loss for $y = -1$.*

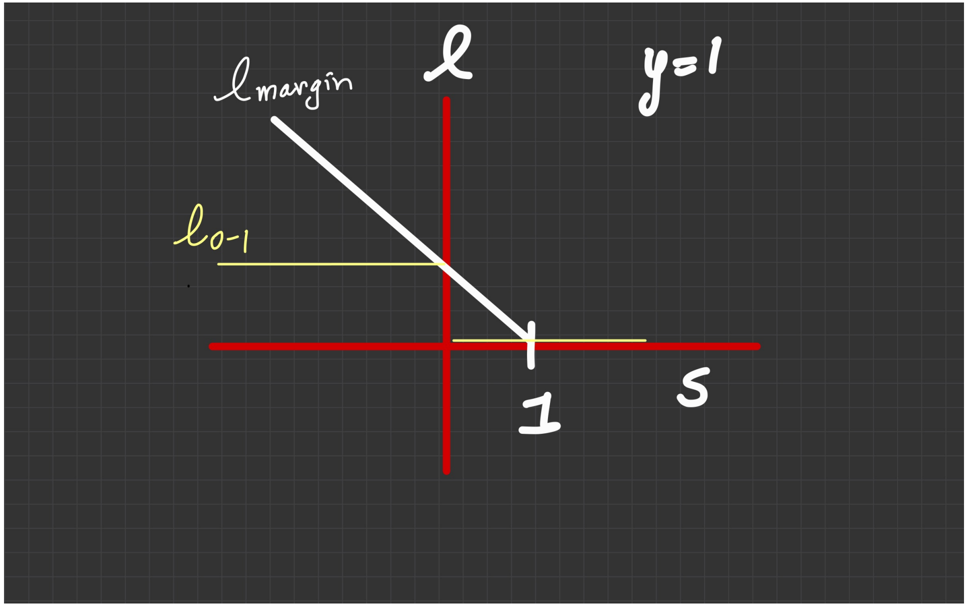

FIGURE 4. Margin loss, this loss is differentiable except at corner, and lies above the 0-1 loss

Exercise 5.4. *In the figure/example, find the margin loss minimizer. Find the value of the loss, and compare to the number of errors.*

5.4. **Analysis of the loss.** The properties of this are as follows:

- This loss is amenable to optimization, since it is piecewise differentiable. (This is not the case for the zero-1 loss.
- The loss lies above the 0-1 loss. So *a posteriori*, after we train, we can estimate the 0-1 loss from the expected loss: it will be an upper bound.

Since ultimately we care about the expected

Theorem 5.5. *dd*

5.5. **Analysis of the loss 2.** In this example, we consider a dataset of scalar data (scores) and labels.

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

In order to analyze the loss.

Theorem 5.6. *Consider minimizing* (EL-C) *with the margin loss* (4) *over the dataset S_m, with the threshold model $s(x) = x - w$ and the classifier $c(s) = \operatorname{sgn}(s)$. Let E_p be the number of false and marginal positives. Let E_n be the number of false and marginal negatives.*

A sufficient condition for a minimizer w^* is that

$$E_n = E_p$$

Proof. Now consider mimizing (EL-C) with this loss

We get ± on each error, depending on cases of false/marginal positive or false/marginal negative.

[[details in handwritten class notes, to be filled in]]

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

So the w^* is the threshold which $E_n = E_p$. □

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

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

Exercise 5.7. *Generalize Theorem 5.6 to the case of* (LAC-FP). *Find the value of C which leads to* $FP = 10FN$

5.6. **Problems and examples.**

Example 5.8. Consider the example of Figure 5.

Find the minimizer of (EL-C) with the score-based threshold classifier and the absolute error loss. 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.

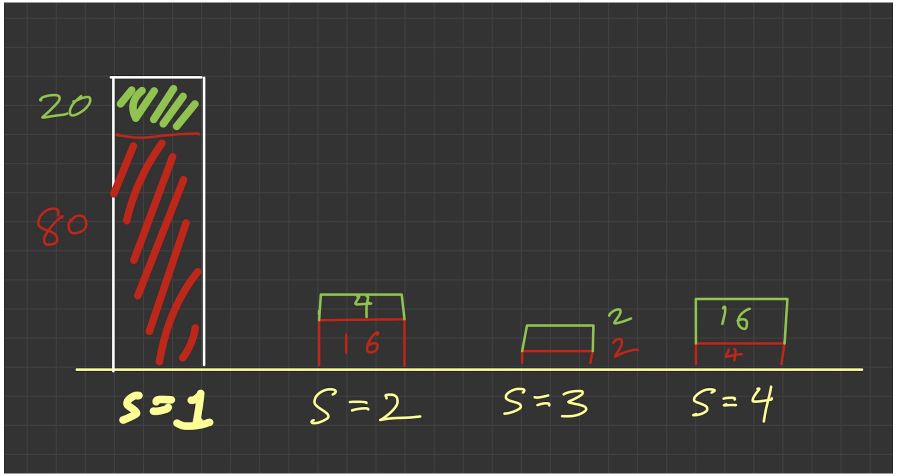

FIGURE 5. Illustration of score classification

Exercise 5.9. *Show that in Figure 5, 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).*

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.