MATH 462 LECTURE NOTES

ADAM M. OBERMAN

Contents

1. Classification: data and features 1
1.1. Threshold models 2
1.2. Binary classification, hinge loss 4
1.3. Multiclass case 5
1.4. Binary classification: Error Types 6
1.5. Linear models 7
References 7

1. Classification: data and features

What has changed from the regression case?

Data can be represented in many ways. We will always work with a vector of features, write The x notation emphasizes that it comes from the data.

Date: September 16, 2021.

In this section, classification, we are learning one of K discrete classes, which we simply represent as one of the integers, $1, \ldots, K$,

$$y \in \mathcal{Y} = \mathcal{Y}_K = 1, \ldots, K$$

The features are still represented as $x \in \mathbb{R}^d$ for a vector of features.

We still have the dataset

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

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

1.1. **Threshold models.** In this example, we have only two classes. In the binary classification case, we adopt the convention that

$$\mathcal{Y}_{binary} = \{-1, 1\} = \{negative, positive\}$$

Here $y = 1$ corresponds to a positive result, and $y = -1$ a negative result (e.g. test). (Although a positive result may not be good thing).

We already have a score, x, and we just need to choose a threshold. The threshold model is

$$h_w(x) = x - w$$

From the model we define a classifier

$$c(x) = \begin{cases} +1 & h(x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

When we already have a threshold model, we can consider the natural 0-1 loss.

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

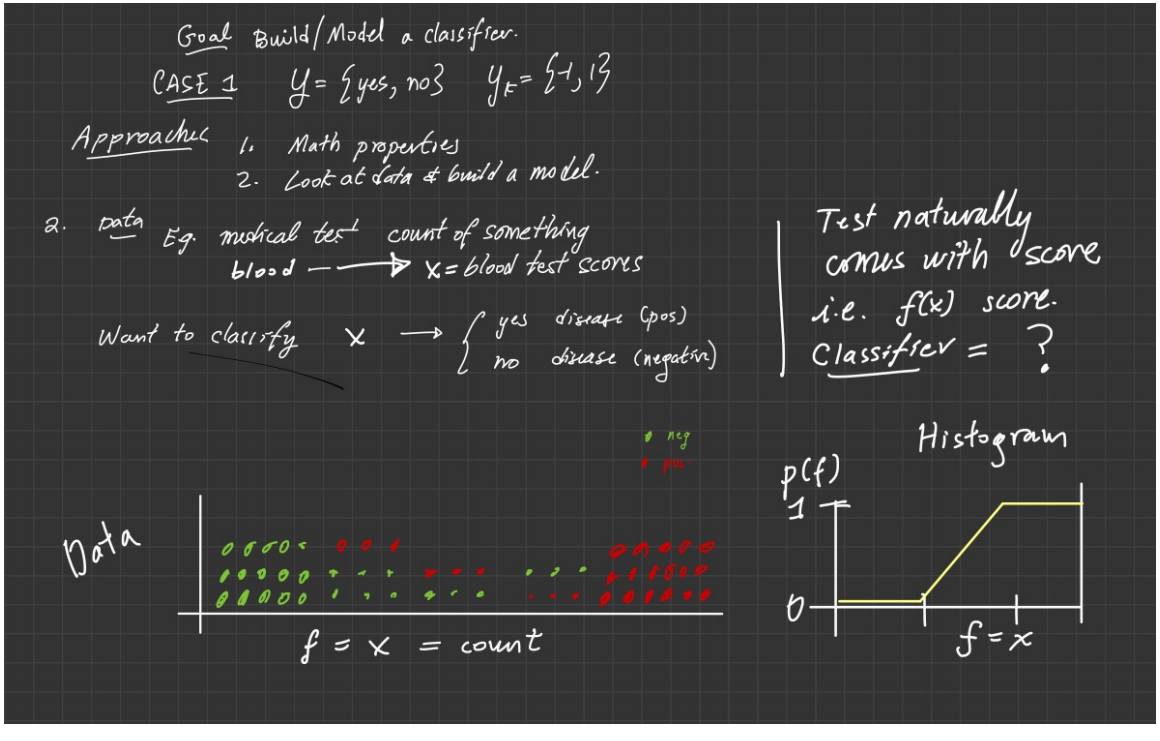

FIGURE 1. Classification example

Then choosing a threshold is a fairly simple problem, which we can pose as an optimization problem

$$\min_w \frac{1}{m} \sum_{i=1}^{m} \ell_{0,1}(c(h_w(x_i)), y_i) \tag{1}$$

Definition 1.1. In binary classification, a consistent score function means: higher score, higher chance of a positive classification.

With a consistent score function, this is a simple search for the threshold that minimizes the errors. Example/Exercise: minimize (1). Show that with a consistent score function, this chooses the threshold where change in the population fraction of the set $x < w$ is $p = .5$. I.e. "derivative" of the 0-1 loss is zero.

1.2. **Binary classification, hinge loss.** Now let's change to a (piecewise) differentiable loss, so we can optimize using gradients.

Suppose we have score function. How do we design a classification loss?

First, we can reduce to a single case by defining $g : \mathbb{R} \times Y_{binary} \to \mathbb{R}$ by

$$g(s, y) = sy = \begin{cases} s & y = 1 \\ -s & y = -1 \\ 0 & ow \end{cases}$$

Next define

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

Note

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

Note, importantly,

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

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

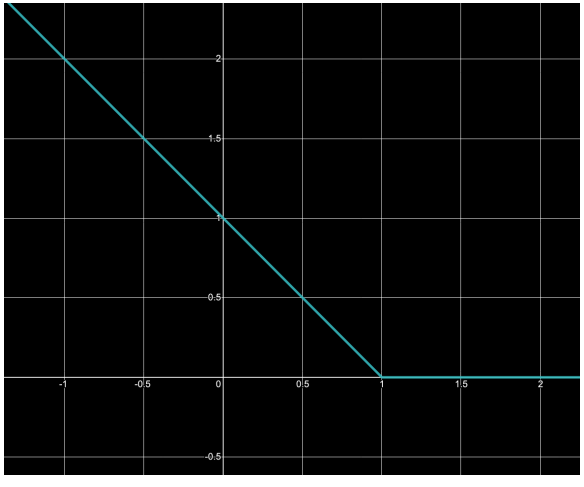

FIGURE 2. Hinge loss, this loss is differentiable except at corner, and lies above the 0-1 loss

1.3. **Multiclass case.** 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 $c : \mathbb{R}^K \to \mathcal{Y}_k$ 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.

Now we can define, as in the binary case,

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

In fact [Exercise] this 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.

1.4. **Binary classification: Error Types.** In the binary classification case, we can define error types. A False Positive (FP) is the error that occurs when $c = 1$ and $y = -1$. Similarly False Negative corresponds to $c = -1$, $y = +1$

For some tests, may care more about different types of errors. (E.g FP worse than FN if means miss a disease).

With the 0-1 loss, and balanced classes, there is no preference for each type of errors.

In this case the loss function can be extended to measure the cost of different error types.

Define C_{FP}, C_{FN} constants, (e.g. $C_{FP} = 10, C_{FN} = 1$).

Then can solve

$$\min_w \frac{1}{m} \sum_{i=1}^{m} \ell_{FP}(c(h_w(x_i)), y_i)$$

where

$$\ell_{FP}(c, y) = \begin{cases} C_{FP} & c = 1, y = -1 \\ C_{FN} & c = -1, y = 1 \\ 0 & \text{otherwise} \end{cases}$$

1.5. **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