6.7900 Fall 2024: Lecture Notes 08

1 Two views of ML models

We have talked about linear regression models so far. To facilitate our understanding towards more complex models that will be covered in the following classes, we would like to discuss two views of ML models.

1. *Distributional.* Take $x \in \mathbb{R}^d$ as an input. Say we have a linear model as $a = \theta^\top x$. We add a transformation that projects a into a probability distribution on the label y. In Figure 1 we generate a Bernoulli distribution on a discrete label y (we have to pick an operation to turn the a value into a probability). One can also generate $\mathcal{N}(y|a, \sigma^2)$ when facing continuous labels as we saw in the linear regression case.

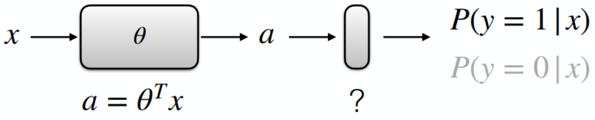

Figure 1: Distributional

2. *Constructive.* Take $x \in \mathbb{R}^d$ as an input. Say we have a linear model as $a = \theta^\top x$. We sample an independent noise ϵ, add it to a, and generates the label. In the figure we generate a 0-1 label $y \in \{0, 1\}$. One can also directly generate $y = a + \epsilon$ when facing continuous labels.

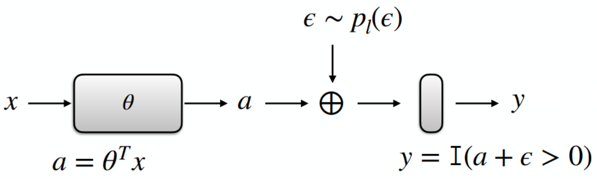
$\epsilon \sim p_l(\epsilon)$
x
θ
a
$\oplus$
y
$a = \theta^T x$
$y = \mathbb{I}(a + \epsilon > 0)$

Figure 2: Constructive

The two models are equivalent mathematically, but the second view can sometimes provide new insights, as we will see in the classification task that will be discussed in this lecture.

2 Classification

In many machine learning tasks we need to predict discrete labels instead of continuous outcomes. However there can be different categories of classification tasks. Let's say in healthcare we have some established treatments and a new treatment.

- *Binary classification.* Will the new treatment be effective?
- *Multi-way classification.* Which treatment is the most effective?
- *Ordinal regression.* Give each treatment a score; the larger the more effective.
- *Ranking problems.* Give a ranking of all treatments by the effectiveness.

Please also refer to Slides pp.2-3 for more examples.

There are also many self-supervised learning problems that are classification tasks: next-word prediction, masked words uncovering, cross-modal matching (e.g., associating images to captions). Please also refer to Slides pp.4-5 for more examples.

2.1 Binary classification as regression

Suppose we have i.i.d. data $\{x^{(i)}, y^{(i)}\}$ with $y^{(i)} \in \{0, 1\}$. We want to build a classification model and make predictions for new x. **Q: Can we solve the classification problem by regression?** A naive approach is to first solve

$$\min_\theta \sum (y^{(i)} - \theta^\top x^{(i)})^2$$

and then make predictions via the rule:

$$y = \mathbb{1}\{\theta^\top x \geq 0.5\}.$$

(Note that here for simplicity we drop θ_0 appeared in the slides. One can always augment x to $[1, x]$ and θ to (θ_0, θ).)

There are at least two drawbacks of this approach.

1. From an MLE perspective, this approach is implicitly assuming y follows a Gaussian distribution around $\theta^\top x$ (with known variance). This does not make sense since $y \in \{0, 1\}$.
2. From a robustness perspective, the fitted parameter and the associated decision rule can be highly sensitive to the training data and make very bad performance on the training/test data, even if the training data can be well separated as in Figure 3 (see also Slides pp.10).

Therefore, we need to find a model and loss function that fits the problem better.

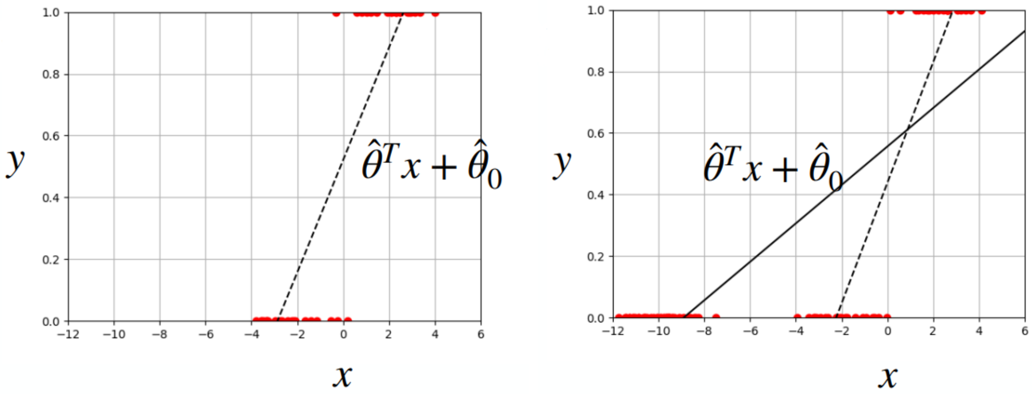

Figure 3: Classification as Regression

2.2 A typical binary classification setting

- Training set: $\{x^{(i)}, y^{(i)}\}_{i=1}^N$, $(x^{(i)}, y^{(i)}) \sim P(x,y)$ i.i.d., $y^{(i)} \in \{0,1\}$, P unknown
- Test cases: $(x,y) \sim P(x,y)$, i.i.d., same unknown P.

We do training via MLE:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log P(y^{(i)}|x^{(i)}, \theta),$$

We do prediction via the following rule:

$$y = \mathbb{1}\{P(y|x,\hat{\theta}) \geq 0.5\}$$

We do testing via 0-1 loss:

$$\mathbb{E}_{(x,y)\sim P}[\mathbb{1}\{P(y|x,\hat{\theta}) < 0.5\}]$$

One can observe that here we do training using one criterion, but we do evaluation using another criterion. Fortunately, this does not incur inconsistency (formally, under some regularity conditions, as $N \to \infty$, the optimal predictor under the training criterion coincides with the optimal classifier under the testing criterion)

3 Logistic regression

The basic idea is to model $P(y=1|x,\theta)$ as a function $\sigma(\theta^\top x)$. Here $\sigma(\cdot)$ is called the *link function*. In logistic regression, we take the so-called sigmoid/logistic function as the link function.

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

This is equivalent to modelling the log odds ratio as a linear function of the covariate x:

$$\log \frac{P(y=1|x,\theta)}{P(y=0|x,\theta)} = \theta^\top x.$$

We sometimes also write

$$\mu_\theta(x) \triangleq \sigma(\theta^\top x).$$

One can also take other forms of σ, as long as $\sigma(\cdot) : \mathbb{R} \to [0,1]$ is strictly increasing. For example, one can take $\sigma(a)$ as the normal cdf $P(Z \leq a)$ where $Z \sim \mathcal{N}(0,1)$. This yields the probit model. Note that different link functions can imply different statistical assumptions.

3.1 Decision rule

Now assume we have a new sample x. In some applications, knowing the distribution suffices. In other applications, we may be forced to make a binary prediction. Then to minimize expected loss, we should do

$$\hat{y} = \mathbb{1}\{P(y=1|x,\theta) > 0.5\} = \mathbb{1}\{\theta^\top x > 0\}.$$

In fact, the decision boundary if we do a "forced prediction" is in the following form:

$$\{x : \theta^\top x = 0\}.$$

Or more generally, the decision boundary can be written as

$$\{x : \theta^\top \phi(x) = 0\}$$

where $\phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ is a feature mapping. That is, the decision boundary is a hyperplane with regard to the feature vector $\phi(\cdot)$. **Q: Is there any natural data generating process that allows a logistic regression model and as a result a decision boundary of hyperplane?** In fact, under class-conditional Gaussian data (see Slides pp.21), this holds true. You actually do get this in HW1 Problem 7 — a linear decision boundary under the same covariance matrix, and a quadratic decision boundary under different covariance matrices.

Finally, it is important to keep in mind that with a different loss function there might be a different optimal decision boundary (e.g., when the loss function is not symmetric). The discussion above gives you intuition, while it would be helpful if you derive the formulas yourself case by case.

3.2 Properties & Optimization

Properties: $\sigma(\cdot)$ has some nice properties: symmetric tails and convexity (see Slides pp.22)
Optimization: The negative log-likelihood is convex, and so one can do stochastic gradient descent to obtain the maximum likelihood estimation of θ (see also Slides pp.25-26). Recall that we want to obtain $\arg\max_\theta P(\{(x^{(i)}, y^{(i)}\}, \theta)$:

$$P(\{(x^{(i)}, y^{(i)}\}, \theta) = \prod_i P(y^{(i)} \mid x^{(i)}, \theta) P(x^{(i)}) \propto \prod_i P(y^{(i)} \mid x^{(i)}, \theta)$$

Using our model, for a given point $(x^{(i)}, y^{(i)})$ we have $P(y^{(i)} \mid x^{(i)}, \theta) = \sigma(\theta^\top x^{(i)})$ if $y^{(i)} = 1$ and $1 - \sigma(\theta^\top x^{(i)})$ otherwise. We can write this (using a trick because our labels are 0 and 1) as $\sigma(\theta^\top x^{(i)})^{y^{(i)}} (1 - \sigma(\theta^\top x^{(i)}))^{1-y^{(i)}}$.

So, we're looking for

$$\begin{aligned} &\arg\max_\theta \prod_i \sigma(\theta^\top x^{(i)})^{y^{(i)}} (1 - \sigma(\theta^\top x^{(i)}))^{1-y^{(i)}} \\ &= \arg\min_\theta \sum_i -\log\left(\sigma(\theta^\top x^{(i)})^{y^{(i)}} (1 - \sigma(\theta^\top x^{(i)}))^{1-y^{(i)}}\right). \end{aligned}$$

Now we can calculate the gradient for any $(x^{(i)}, y^{(i)})$:

$$\begin{aligned} &\nabla_\theta(-\log \sigma(\theta^\top x^{(i)})^{y^{(i)}} (1 - \sigma(\theta^\top x^{(i)}))^{1-y^{(i)}}) \\ &= \nabla_\theta(-y^{(i)} \log \sigma(\theta^\top x^{(i)}) - (1 - y^{(i)}) \log(1 - \sigma(\theta^\top x^{(i)}))) \\ &= \nabla_\theta H(y^{(i)}, \sigma(\theta^\top x^{(i)})) \quad (H(\cdot, \cdot) \text{ is cross entropy}) \\ &= \underbrace{(\sigma(\theta^\top x^{(i)}) - y^{(i)})}_{\text{prediction error}} x^{(i)}. \quad \text{(left as an exercise)} \end{aligned}$$

In each t, sample (x, y) from the training dataset and update θ with a suitably chosen $\eta_t = \Theta(1/\sqrt{t})$:

$$\theta \leftarrow \theta - \eta_t(\sigma(\theta^\top x) - y)x.$$

3.3 A constructive view

We have been focusing on the distributional view so far. In fact, fitting data via logistic regression can be regarded as assuming the following data generating process (see also Figure 2):

$$x \to \theta^\top x \to \theta^\top x + \epsilon \to \mathbb{1}\{\theta^\top x + \epsilon > 0\} = y.$$

Here, ϵ is a noise term that follows the logistic distribution with the cdf:

$$P(\epsilon < -\theta^\top x) = \sigma(-\theta^\top x) = \int_{-\infty}^{-\theta^\top x} p_l(a) da = \int_{\theta^\top x}^{+\infty} p_l(a) da.$$

Here $p_l(a) \triangleq \sigma'(a) = \sigma(a)\sigma(-a)$. If one assumes ϵ follows the Gaussian distribution, then we get a probit model.

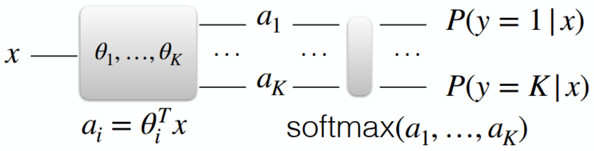

Figure 4: Multi-way classification

4 Extension to multi-way classification

We can generalize the 2-class case to the multi-class case.

$$\text{softmax}(a_1, \cdots, a_K) = \left(\frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)} \right)_{k=1}^{K}.$$

In other words, we are modelling the log odds ratios as

$$\log \frac{P(y = i|x, \theta)}{P(y = j|x, \theta)} = \theta_i^\top x - \theta_j^\top x$$

where $\theta = (\theta_1, \cdots, \theta_K)$.
A Constructive View.

$$\begin{aligned} x &\to \theta_1^\top x, \cdots, \theta_K^\top x \\ &\to \theta_1^\top x + \epsilon_1, \cdots, \theta_K^\top x + \epsilon_K \\ &\to \arg\max\{\theta_i^\top x + \epsilon_i\} = y. \end{aligned}$$

Here, each ϵ_i follows a Gumbel distribution with pdf Gumbel$(x|0, 1) = \exp(-x - \exp(-x))$. The Gumbel distribution is interesting and will come up again later.

Ordinal regression and ranking prediction not covered in class. Please refer to Slides pp.32-36.