Lecture 6 probabilistic models

wbg231

December 2022

1 overview

- if we learn models as a statistical inference, we have both a unified frame work that covers many classes of models
- and a principled way to incorporate prior beliefs about the data into the model
- this can either be done by learning conditional models or generative models

conditional models

linear regression

- given training data $\mathcal{D} = ((x_1, y_1) \cdots (x_n.y_n))$
- we want to learn a parameter $\theta \in mathbbr^d$ and predict y as

$$h(x) = \sum_{i=1}^{n} \theta_i x_i = \theta^t x$$

- we cna add the bias term by setting $x_0 = 1$ (that is we add a term to each data vector which is constant)
- we can minimize squared loss over this method

$$j(\theta) = \frac{1}{n} \sum_{n=1}^{N} (y^n \theta^t x^n)^2 = (X\theta - y)^t (X\theta - y)$$

- this has a closed form $\hat{\theta} = (X^t X)^{-1} X^t y = \Sigma_x X^t y = \Sigma_X P_X(y)$ if x is normalized

assumptions of linear regression

- we assume that x and y are linearly related ie $y = \theta^t x + \epsilon$ where ϵ is our residual error (or noise)
- we assume that the error is iid and

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

- so then what is the distribution of $\tilde{y}|\tilde{x} = x$? if X is held constant, than $\tilde{y}|\tilde{x} = x$ only depends on the noise and thus

$$P(\tilde{y} = y|\tilde{x} = x, \tilde{\theta} = \theta) \sim \mathcal{N}(\theta^T x, \sigma^2)$$

- the notation $P(y|x;\theta)$ can be though of as the likelihood of y (ie a certain outcome) given our input data x is fixed and θ ie our model parameter is true
- so, each point that we are predicting for is a gaussian random variable.
- the maximum likelihood principle says that we would like to max the conditional likelihood of our data that is

$$\mathcal{L}(\theta) = P(D, \theta) = \Pi_{n=1}^{n} P(\tilde{y}^n = y|\tilde{x}^n = x, \tilde{\theta} = \theta) = \Pi_{n=1}^{N} P(y^n|x^n, \theta)$$

- in practice we work with the log likelihood since it is more stable(sine the product of many probabilities will be very small)

mle and linear regression

- for the sake of time at this point i am just going to include pictures of derivations that i think are kinda clear

$$
\begin{aligned}
\ell(\theta) &\stackrel{\text{def}}{=} \log L(\theta) \\
&= \log \prod_{n=1}^{N} p(y^{(n)} \mid x^{(n)}; \theta) \\
&= \sum_{n=1}^{N} \log p(y^{(n)} \mid x^{(n)}; \theta) \\
&= \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(n)} - \theta^T x^{(n)}\right)^2}{2\sigma^2}\right) \\
&= N \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(y^{(n)} - \theta^T x^{(n)}\right)^2
\end{aligned}
$$

-
- so by the assumptions of linear regression that is our log likelihood function
- now we want to maximize it so we can see

  $$\nabla \ell_\theta(\theta) = -\frac{1}{\sigma^2} \sum_{i=1}^{n} (y^n - \theta^t x^n) x^n = -\frac{1}{\sigma^2} (X^t y - \theta^t X X^t) \Rightarrow \theta^* = (XX^t)^{-1}(X^t y)$$

- so in other words probabilistic linear regression yields the same closed form as that obtained through erm with squared loss
- however assuming that noise is gaussian is not always reasonable as is the case in classification
- so we are going to build logistic regression

logistic regression

- consider a binary classification problem where $y \in \{0, 1\}$ what should the distribution of $\tilde{y}|\tilde{x} = x$ look likelihood
- perhapses a bernoulli with parameter $\theta = h(x)$ ie

  $$P(y|x) = h(x)^y (1 - h(x))^{1-y}$$

  so note here if $y = 0$ then $P(y = 0|x) = (1 - h(x)) = h(x)^1(1 - h(x))^0 = h(x)^1(1 - h(x))^{1-1} = h(x)^1(1 - h(x))^{1-y}$
- so how can we learn $h(x)$? we know that $h(x) \in (0, 1)$ as it is a probability

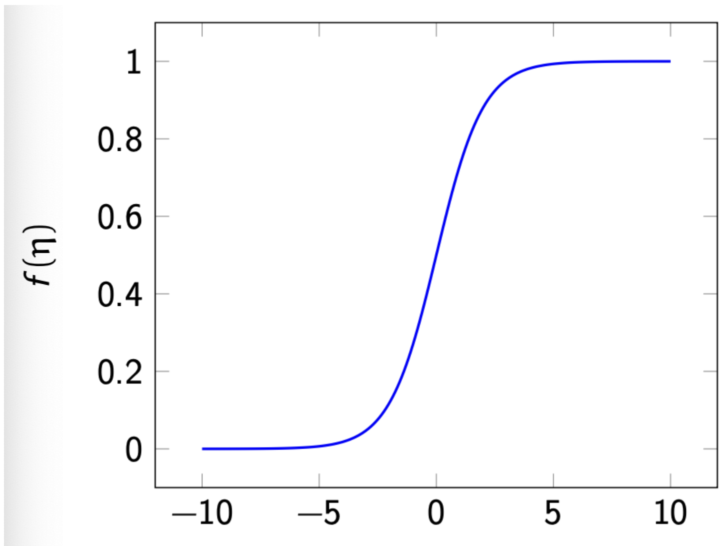
- recall that the linear problem with gaussian noise

$$E[\tilde{y}|\tilde{x}=1,\theta] = \theta^t x = h(x)$$

  so this has the mean we want, so lets just find a function that maps this linear predictor to (0,1) and use that as a probability

- enter the logistic function

$$f(\eta) = \frac{1}{1+e^{-\eta}}$$



- so we let

$$P(\tilde{y}|\tilde{x}=\tilde{x}) \sim bernoulli(logistic(\theta^t x))$$

- so in other words for each point data point, we think of the outcome of that data point as a bernoulli random variable with some fixed parameter which is the normalized (though the logistic function) mean of our gaussian linear regression problem ie $\theta^t x$

- $P(y|x) =$Bernoulli$f(\theta^t x)$
  - look at the log odds $log(\frac{P(y=1|x)}{P(y=0|x)}) = \theta^T x$
  - this can be expressed as $log(P(y=1|x)) - log(p(y=0|x)$
  - recall that we can write $P(y|x) = P(y|x) = h(x)^y(1-h(x))^{1-y}$
  - so thus we have $log(\frac{P(y=1|x)}{P(y=0|x)}) = \theta^T x = log(P(y=1|x)) - log(p(y=0|x) = log(f(x)^1(1-h(x))^{1-1}) - log(f(x)^{(}0)(1-h(x))^{1-0}) = log(f(x)) - log(1-f(x)) = log(\frac{1}{1+e^{-\eta}}) - log(1-\frac{1}{1+e^{-\eta}}) = log(1) - log(1+e^{-\eta}) - log(\frac{1+\eta^-n-1}{1+\eta^-n} = log(1+e^{-\eta}) - log(\frac{e^-\eta}{1+e^{-\eta}} = log(1+e^{-\eta}) - log(e^{-\eta}) + log(1+e^{-\eta}) = -log(e^{-\eta}) = \eta = \theta^t x$

- so in other words the log odds are a linear function that form a decision boundary, that is a linear decision boundary
- this means the decision boundary is linear, ie the features are linear in the parameter as we increase the value of $\theta^t x$ we get 1, and as we decrees it we get zero

- so lets find the gradient and MLE

Likelihood for a single example: $\ell^n = y^{(n)} \log f(\theta^T x^{(n)}) + (1$

$$
\begin{aligned}
\frac{\partial \ell^n}{\partial \theta_i} &= \frac{\partial \ell^n}{\partial f^n}\frac{\partial f^n}{\partial \theta_i} \\
&= \left(\frac{y^{(n)}}{f^n} - \frac{1-y^{(n)}}{1-f^n}\right)\frac{\partial f^n}{\partial \theta_i} \\
&= \left(\frac{y^{(n)}}{f^n} - \frac{1-y^{(n)}}{1-f^n}\right)\left(f^n(1-f^n)x_i^{(n)}\right) \qquad \text{Exercise:} \\
&= (y^{(n)} - f^n)x_i^{(n)}
\end{aligned}
$$

- The full gradient is thus $\frac{\partial \ell}{\partial \theta_i} = \sum_{n=1}^{N}(y^{(n)} - f(\theta^T x^{(n)}))x_i^{(n)}$.

- with some pretty straight forward calc we see this. there are more details on this in my full lecture notes
- notice that in both formulas we had a pretty similar gradient
- this is a more general property of liner models as will see

linear vs logistic regression

- linear regression
  1. we combine the inputs as a linear combination or weighted sum ie θ^t
  2. we output $y \in \mathbb{R}$
  3. $\tilde{y}|\tilde{x} = x, \theta \sim \mathcal{N}(\theta^t x, \sigma^2)$
  4. our transfer function (ie how we transfer or map the linear combination to a prediction) is the identity map ie $f(\theta^t x) = \theta^t x$
  5. the mean of our conditional distribution is $E[\tilde{y}|\tilde{x} = x, \theta] = \theta^t x = f(\theta^t x)$ (where f is our transfer function)
- logistic regression
  1. we take a linear combination of te inputs $\theta^t x$
  2. our out put is categorical (as this is a classification problem)

3. our conditional distribution $\tilde{y}|\tilde{x} = x, \theta \sim Bernoulli(f(\theta^t x))$
4. our transfer function ie how we map our linear function to our prediction function is the logistic function $f(\theta^t x) = \frac{1}{1+e^{-\theta^t x}}$
5. the mean of our conditional distribution is $E[\tilde{y}|\tilde{x} = x, \theta] = 1(P(y = 1|\theta^t x)) + (0)(1 - P(Y = 1|\theta^t x)) = P(Y = 1|x, \theta) = f(\theta^t x)$

- in both cases x enters through a linear function
- the mean difference between the two is due to there conditional distributions
- can we generalize this?

generalized regression model

- our task is given some x find the distribution of y conditional on that ie $P(\tilde{y}|\tilde{x} = x)$
- to model this,
  1. chose a parametric family of distributions $p(y|x, \theta)$ with a parameter $\theta \in \Theta$
  2. chose a transfer function that maps a linear predictor in $\mathbb{R}$ to Θ ie

$$x \in \mathbb{R}^d \to w^t x \in \mathbb{R} \to f(w^t x) = \theta \in \Theta$$

- the finally we learn $\hat{\theta} \in argmax_\theta log(P(\mathcal{D}|\theta))$

poisson regression example

- the Poisson distribution is a discrete probability distribution used to model the number of events during a fixed time period has parameter λ and pdf

$$P(\tilde{y} = k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

  where $\lambda < 0, E[Y] = \lambda$

- suppose that we chose our parametric distribution of families to be poisson that is assume $\tilde{y}|\tilde{x} = x, \eta \sim poisson(\eta)$
- so how can we think about our transfer function?

$$x \to w^t x \in \mathbb{R} \to f(w^t x) = \lambda \in (0, \infty)$$

- if we are mapping from $\mathbb{R} \to (0, \infty)$ it is a common choice to let $f(x) = e^x$

- all right so lets derive the mle for this type of variable

$$\mathcal{L}(D|\theta) = \Pi_{n=1}^{n} P(Y^n|\lambda^n) = \Pi_{i=1}^{n} \frac{(\lambda^n)^{y^n} e^{-\lambda^n}}{(y^n)!}$$

- meaning that

$$\ell = \sum_{n=1}^{n} y^n log(\lambda) - \lambda^n - log((y^n)!) = \sum_{n=1}^{n} y^n log(e^{w^t x^n}) e^{-w^t x^n} - log((y^n)!)$$

where $\lambda^n = e^{w^t x^n}$

- then we can find our gradient as

$$\nabla \ell_w = \sum (\frac{y^n}{e^{w^t x^n}} - 1)(e^{w^t x^n} x^n)$$

multinomial logistic regression

- we are going from a bernoulli distribution to a categorical in that case
- so we can say

$$\tilde{y} = y|\tilde{x} = x, \theta \sim categorical(\theta) : \quad \theta \in \mathbb{R}^d, \sum \theta = 1, \theta_i \leq 0 \quad \forall i \in [1, k]$$

and we can think of $\theta_i = P(y = i|x, \theta)$ that is each element in θ is the likelihood an example given it's inputs is class that class

- for each x we compute a linear score function for each class that is

$$x \rightarrow (w_1^t x, \cdots, w_k^t x) \in \mathbb{R}^k$$

that is for a given x we can compute a dot product between that input and each classes weight vector to get (what equates to the similarly between the two)

- the soft max function is the our transfer function mapping our k scores to a probability vector $\theta \in \mathbb{R}^d$ which sums to 1.

$$(s_1 \cdots s_k) \rightarrow \theta = (\frac{e^{s_1}}{\sum_{i=1}^{k} e^{s_i}} \cdots \frac{e^{s_k}}{\sum_{i=1}^{k} e^{s_i}})$$

- so further

$$p(y = c|x, w) = \frac{e^{w_y^t x}}{\sum_{i=1}^{k} e^{w_i^t x}}$$

- this can be thought of as learning k linear regression models, then passing them through this transfer function to normalize each model and predict based on which is most likely

review

- Recipe for conditional distribution for prediction
  1. define input and outputs space
  2. chose the output distribution $P(y|x, \theta)$ could be conditional, could be bernoulli, could be gaussian etc
  3. chose a transfer function that maps $w^t x$ to the parameter space of that parametric distribution Θ
- then to learn the model fit a maximum likelihood estimator to the data

generative models

bayes rule

- our goal is to learn the joint distribution

$$P(x, y|\theta)$$

- then predict the label for x as

$$argmax_{y \in Y} P(x, y|\theta)$$

- so in conditional models we learn $P(y|x, \theta)$ that what is the distribution of y given we hold x and θ constant where as in generative models we are learning $P(x, y|\theta)$ so we are learning how x and y are disputed together under the assumption of there being some true parameter θ
- we train as

$$P(x, y) = P(x|y)P(y)$$

  that is we learn the joint distribution of x and y by modeling as the product of two distributions
- then we test by writing

$$argmax_y P(y|x) = argmax_y \frac{p(x|y)P(y)}{P(x)} = argmax_y P(x|y)P(y)$$

  so that is we predict using the most likely class according to out model.

naive bayes

- suppose we want to label an email as real or spam
- let our input space be all possible emails and let $x \in X$ be an email where $x_i \in [0,1]$ represents if the ith word in some dictionary is in that email
- so what is the probability of a given document x?

$$P(x) = \prod_{y \in Y} P(x,y) = \prod_{y \in Y} P(x|y)P(y)$$

- then what is the likelihood of of one document given a class

$$P(x|y) = P(x_1 \cdots x_d|y) = P(x_1|y)P(x_2|y,x_1) \cdots P(x_d|y,x_1 \cdots x_{d-1}) = \prod_{i=1}^{d} P(x_i|y,x_{<i})$$

- this problem has a tone of dependencies but
- to deal with this we have the naive bayes assumption which says that features are conditionally independint of one another given the label and thus

$$P(x|y) = \prod_{i=1}^{d} P(x_i|y)$$

- assume that $P(x_i = 1|y = 1) = \theta_{i,1}, P(x_i = 1|y = 0) = \theta_{i,0}$ so for each example we are learning 2 (but really one parameter)
- and $P(y = 1) = \theta_1$
- so we can write

$$P(x,y) = P(x|y)P(y) = p(y)\prod_{i=1}^{d} P(x_i|y) = p(y)\prod_{i=1}^{d}(\theta_{i,y}\mathbb{I}(x_i = 1)+(1-\theta_{i,y}\mathbb{I}(x_i = 0)))$$

- so here we max the licklyhood of the data $\prod_{i=1}^{n} p_\theta(x^n, y^n)$ so we are maximizing the licklyhood of our overall data not just of the conditional likelihood of seeing y

ikelihood we've seen before).

$$\frac{\partial}{\partial\theta_{j,1}}\ell = \frac{\partial}{\partial\theta_{j,1}}\sum_{n=1}^{N}\sum_{i=1}^{d}\log\left(\theta_{i,y^{(n)}}\mathbb{I}\left\{x_i^{(n)}=1\right\}+\left(1-\theta_{i,y^{(n)}}\right)\mathbb{I}\left\{x_i^{(n)}=0\right\}\right)+\log p_{\theta_0}(y^{(n)}) \quad (4$$

$$= \frac{\partial}{\partial\theta_{j,1}}\sum_{n=1}^{N}\log\left(\theta_{j,y^{(n)}}\mathbb{I}\left\{x_j^{(n)}=1\right\}+\left(1-\theta_{j,y^{(n)}}\right)\mathbb{I}\left\{x_j^{(n)}=0\right\}\right) \qquad \text{ignore } i \neq j \quad (4$$

$$= \sum_{n=1}^{N}\mathbb{I}\left\{y^{(n)}=1 \wedge x_j^{(n)}=1\right\}\frac{1}{\theta_{j,1}}+\mathbb{I}\left\{y^{(n)}=1 \wedge x_j^{(n)}=0\right\}\frac{1}{1-\theta_{j,1}} \qquad \text{ignore } y^{(n)} \quad (4$$

-

- so i think the thing to keep in mind here is that $y_i = 1, x_i = 1$ mean difference things. $y_i = 1$ means that the email is span $x_i = 1$ means the word is present
- this is actually a pretty simple weighted sum of number of observations as our derivative
- solving this out we see

$$P(x_i|y) = \theta_{j,1} = \sum_{i=1}^{n} \frac{\mathbb{I}(y^n = 1 \wedge x_j^n = 1)}{\mathbb{I}(y^n = 1)} = \frac{\text{number of spam reviews with the word}}{\text{number of spam reviews}}$$

- we can expand the naive bayes model to continuous outputs by setting our conditional probability of x given y as $P(x_i|y = k) \sim \mathcal{N}(\mu_{i,k}, \sigma_{i,k}^2)$
- the math and interperation are largely the same though