Lecture 6: probabilistic models

wbg231

December 2022

1 overview

1.1 why probabilistic models

- gives us a unified framework for many models
- allows us to take tools form probability theory and use them to solve problems it also gives us a principled way to insert previous info into our model

1.2 today's lecture

- there are two ways to model data generation.
- 1. conditional $P(y|x)$
- generative models we want to learn the joint $P(x, y)$ and use that to make predictions
- we want to understand how can we build these models and estimate there parameters
- in both cases we use mle
- and we can compare generative and conditional models

2 conditional models

2.1 linear regression

- lets start with an example
- our goal is to predicted $y \in \mathbb{R}$ using our feature vector x from our data.
- could be house pricing, medical prediction, age of person based on photo etc

2.2 problem set up

- data training examples $D = \{(x_i, y_i)\}_{i=1}^n$ where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$
- model: a linear function$h$ with parameter θ to predict y from x

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

  where $\theta \in \mathbb{R}^d$ are our weights we predict the label as a linear function of our inputs
- we Incorporated a bias term (also called interpret) into x

2.3 parameter estimation

- loss function:we estimate θ by minimizing the square loss (the least squared method)

  $$j(\theta) = \frac{1}{n}\Sigma_{i=1}^n (y^i - \theta^t x^i)^2$$

  (empirical risk)
- let $X \in \mathbb{R}^{nd}$ be the design matrix whose rows are input features
- let $y \in \mathbb{R}^n$ by the vector of all targets
- we want to find

  $$\hat{\theta} = agmin_\theta J(\theta) = argmin_\theta (X\theta - y)^t (X\theta - y)$$

- we can see that $\nabla j(\theta) = 2X^t X\theta - 2X^T y$ which yields
- closed form solution $\hat{\theta} = (X^T X)^{-1} X^t y$
- note here that $(X^T X) \in \mathbb{R}^{d \times d}$ may not be invertable.
- recall that for a matrix $A \in \mathbb{R}^{n \times d}, rank(A)$ is the minimum number of linearly independent rows and columns in that matrix, and further $rank(A^T) = rank(A)$ and for any two matrices $rank(AB) \leq min(rank(a), rank(b))$
- so in this case $rank(A^T A) \leq min(rank(a), rank(a^T)) \leq min(n, d)$
- so in other words we must have at least d linearly independent examples, as well as all linearly independent features for our closed form solution to exist and be unique.
- we know that $X^T X$ is positive semi-definite so if it not invertable then there is a null space, meaning we can add any vector in the null space to our solution and thus the solution is not unique.

2.4 review

- so far we have seen linear regression, we assume that there is a linear function of our input and we take the best model based on the square loss.
- but why do we use the square loss?
- what are the assumptions we are making about our model and data that bring us to the square loss
- how can we look at it from a different perspective, and what do we need to assume about our data in terms of probability to derive this objective?

2.5 assumptions in linear regression

- how can we derive the objective form the data
- we assume that there is a linear function with some noise (ϵ) mapping x to y. that is
$$y = \theta^t x + \epsilon$$
- where we call ϵ the residual error capture all unmodeld effects eg noise
- further we assume t=hat the errors are distributed iid according to
$$\epsilon \sim N(0, \sigma^2)$$
- so what is the conditional distribution $Y|X = x$ (here capitals are rv)
  - well we know $Y|x = x$ is given by $y = \theta^t x + \epsilon$
  - where x and θ are fixed
  - thus $E[y] = E[\theta^t x + \epsilon] = E[\theta^t x] + E[\epsilon] = E[\theta^t x] + 0 = \theta^t x$
  - and variance is $var(y) = var(\theta^T x + \epsilon) = var(\theta^t x) + var(\epsilon) + 2cov(\theta^t x, y) = 0 + \sigma^2 + 0 = \sigma^2$
  - thus $Y|X = x \sim \mathcal{N}(\theta^t x, \sigma^2)$
- so in other words we are assuming that $Y|X = x$ is just a normal Gaussian shifted. this can be thought of as putting a Gaussian bump around the output of the linear predictor.
- so this is a linear map with some Gaussian noise around the results.
- this is an assumption this may be true, it may be untrue.

2.6 Maximum likelihood estimator

- so given a probabilistic model and dataset $\mathcal{D}$ how do we estimate θ? Assuming that our examples are iid
- the maximum likelihood principle stats that we should max the conditional likelihood of our data that is

$$\mathcal{L}(\theta) = P(\mathcal{D}|\theta) = P(y_1...y_n|\theta x_1...\theta x_n) = \Pi_{i=1}^{n} P(y^i|x^i, \theta)$$

- in practice we use the log log likelihood since it makes is easier to work with really small values, and sometimes try to minimize the negative log likelihood.

2.7 mle for linear regression

- so we can write
  $\ell(\theta) =$
  $\log(L(\theta) =$
  $log\Pi_{i=1}^{n} P(y^i|x^i, \theta)$
  $\Sigma_{i=1}^{n} log(p(y^i|x_i, \theta)) =$
  $\Sigma_{i=1}^{n} log(\frac{1}{\sqrt{w\pi\sigma}} e^{-\frac{1}{2}(\frac{y^i - \theta^t x_i}{\sigma})^2}) =$
  $\Sigma_{i=1}^{n} log(\frac{1}{\sqrt{w\pi\sigma}} e^{-(\frac{(y^i - \theta^t x^i)^2}{2(\sigma)^2}})$
  $= \Sigma_{i=1}^{n} (log(\frac{1}{\sqrt{w\pi\sigma}}) - \frac{(y^i - \theta^t x^i)^2}{2(\sigma)^2}) =$
  $n(log(\frac{1}{\sqrt{w\pi\sigma}}) - \frac{1}{2\sigma^2} \Sigma_{i=1}^{n} (y^i - \theta^t x^i)^2$
- our goal is to maximise this function,so we can take the gradient
- so first off note that we can write

$$\ell(\theta) = n(log(\frac{1}{\sqrt{w\pi\sigma}}) - \frac{1}{2\sigma^2} \Sigma_{i=1}^{n} (y^i - \theta^t x^i)^2 = n(log(\frac{1}{\sqrt{w\pi\sigma}}) - \frac{1}{2\sigma^2} \Sigma_{i=1}^{n} (y^i)^2 - 2\theta^t x^i y^i + (\theta^t x^i)^t (\theta^t x^i)$$

- then we can see that $\frac{\partial \ell}{\partial \theta_i} = -\frac{1}{2\sigma^2} \Sigma_{j=1}^{n} (2y^j x_i^j - 2\theta x^j x_i^j) = -\frac{1}{\sigma^2} \Sigma_{j=1}^{n} (y^j - \theta x^j) x_i^j$
- than we can set that two zero and see that $\theta^* = \Sigma_{j=1}^{n} \frac{y^j x_i^j}{x^j x_i^j}$
- then if we take the derivative with rep sect to θ as a vector we get $\theta^* = \Sigma_{i=1}^{n} \frac{x^i y^y}{(x^i)^t x_i} = (X^T X)^{-1} (X^T y)$ which is the linear regression solution

2.8 review

- so if we assume that our data generating process is Gaussian it is the same as doing least squares.
- but what if our data is not Gaussian

3 logistic regression

- consider binary classification $y \in \{0, 1\}$ that should the conditional distribution be? we model $P(Y|X = x)$ as a Bernoulli such that

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

- how should be parameterize y?
  - what is $P(y = 1|x)$ and $P(y = p|x$ where $h(x) \in (0, 1)$? these are basically transformed linear regressions models
  - what is the mean $Y|X = x$? we need a function f to map the linear predictor $\theta^t x \in \mathbb{R}$ to (0,1)? $E[Y|X = x] = \Sigma_{y \in y} yP(y = y|x = x) = 1P(y = 1|x = x) + 0P(y = 0|x = x) = P(y = 1|x = x)$
  - we define the logistic function as

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

- the logistic function looks like this

lecture_notes/lecture_6/immages/lecture_6_1.jpg

- $P(y|x)$ =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

- how can we extend this to multi-class classification?

3.1 mle for logistic regression

- we would again like to max our conditional log likely hood
- $\ell(\theta) = log(\mathcal{L}(\theta)) = \Sigma_{i=1}^n log(P(y^i|x^i, \theta)) = \Sigma_{i=1}^n log((f(\theta^t x^i)^{yi}(1-f(\theta^t x^i))^{1-y^i}) = \Sigma_{i=1}^n y^i log(f(\theta^t x^i) + (1-y^i) log(1-f(\theta^t x^i))$
- this does not have a closed form solution, but the likelihood is concave so we can use gradient ascent to get a unique optimal solution

3.2 gradient ascent for logistic regression

- so call gradient ascent $\theta = \theta + \nabla\ell(\theta)$
- so call the liklyhood of a single example $\ell_n = r^n log(f(\theta^t x^n) + (1 - y^n) log(1 - f(\theta^T x^n))$
- then we can see that $\frac{\partial \ell^n}{\partial \theta_i} = \frac{\partial \ell^n}{\partial f^n}\frac{\partial f^n}{\partial \theta_i} = (\frac{y^n}{f^n} - \frac{1-y^n}{1-f^n})\frac{\partial f^n}{\partial \theta_i}$
- recall that $f^n(\theta^t x^i) = \frac{1}{1+e^{-\theta^t x_i}}$
- thus we have $\frac{\partial f^n}{\theta_i} = \frac{\partial f^n}{e^{-\theta x_i}}\frac{\partial e^{-\theta x_i}}{\partial -\theta^t x_i}\frac{\theta^t x_i}{\theta_i} = \frac{-1}{(1+e^{-\theta^t x_i})^2} - e^{\theta^t x_i} x_{ij}$
- thus we have $\frac{\partial \ell^n}{\partial \theta_i} = \frac{\partial \ell^n}{\partial f^n}\frac{\partial f^n}{\partial \theta_i} = (\frac{y^n}{f^n} - \frac{1-y^n}{1-f^n})\frac{\partial f^n}{\partial \theta_i} = \frac{\partial \ell^n}{\partial f^n}\frac{\partial f^n}{\partial \theta_i} = (\frac{y^n}{f^n} - \frac{1-y^n}{1-f^n})(f^n(1-f^n)x_i^n) = (y^n - f^n)x_i^n$
- and then if we take the full partial we get $\frac{\partial \ell}{\partial \theta_i} = \Sigma_{j=1}^n (y^j - f(\theta^t x^j))x_i^j$

3.3 compare linear and logistic regression

- linear
  - it is linear
  - outputs real numbers
  - we assume that conditional distribution $Y|X = x$ is Gaussian
  - the transfer function $F(\theta^t x_i)$ is identity
  - and the mean $E[Y|X = x, \theta] = \theta^t x$ since we assumed that $Y|x = x \sim N(\theta^t x, \sigma^2)$
- logistic
  - it is linear
  - outputs categorical variables
  - we assume that condital distrobution $Y|X = x$ is bernuli
  - the transfer function is logistic
  - is also $f(\theta^t x)$ (i think this has to do with the log likelihood being that.
- the main difference between the two how we model the conditional distribution

4 generalized regression model

- suppose we have x and we want to predict $P(y|x)$
- for modeling we Can chose a parametric family of distributions $P(y, \theta)$ with parameters $\theta \in \Theta$
- chose a transfer function that maps a linear predictor in $\mathbb{R}$ to Θ

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

- so we can train this model using MLE

$$\hat{\theta} \in argmax_{\theta} logp(\mathcal{D}, \hat{\theta})$$

- and make predictions as $x \rightarrow f(w^t x)$
- so the transfer function in linear regression is identity
- the transfer function in logistic regression is the logistic function
- these algorithms only deffer based on the transfer function
- so the idea is we can do kind of simple regression, then use a transfer function to transom that model into the context we are working in (that is into our parametric family of distributions.)

4.1 example Poisson regression

- say we want to predict the number of people enter a restaurant in NYC during lurch time
- what features can be use dull, what i a good model for the number of visitors
- recall that a passion random variable Y with parameter λ has conditional

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

  this is a good model for the number of events that occur with in a period of time
- so suppose we say that our conditional distribution $Y|X = x \sim$Poisson(λ)
- x enters linearly that is

  $$x \to w^T x \to \lambda = f(w^t x)$$

  so that is we are using standard linear regression to learn a parameter λ that we then use as the parameter in our Poisson model
- standard approach is to use the exponential distribution ie $f(w^t x) = exp(w^t x)$
- the log likelihood of the full dataset $\mathcal{D}$ under our assumptions is

  $$log(p(\mathcal{D}, \lambda_i) = \Sigma_{i=1}^{n} log(\frac{\lambda^{y_i} e - \lambda}{y_i!}) = \Sigma_{i=1}^{n}[y_i log(\lambda_i) - \lambda_i - log(y_i!)] = log(p, \mathcal{D}, w)$$

  $$= \Sigma_{i=1}^{n}[y_i log(exp(w^t x_i)) - exp(w^t x_i) - log(y_i!)] = \Sigma_{i=1}^{n}[y_i w^t x_i - exp(w^t x_i) - log(y_i!)]$$
- we can take derivatives of this and try to solve it.
- this is an example of how we can model a problem by specifying our assumptions about the model.

4.2 multinomal logistic regression example

- we are going to extend the logistic regression to multiple classes
- so we parameterize this with a probability vector $\theta \in \mathbb{R}^k$ where there are k classes.
- a probability vector θ has the following properties
  - $\sigma_{i=1}^{n} \theta_i = 1$ and $\theta_i \geq 0$

  - and $\forall y \in [1,k] P(Y=y) = \theta_y$

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

$$x \rightarrow [< w_1, x >, , , < w_k, x >] \in R^d$$

- and we use the soft max function as our transfer function

$$(s_1..s_k) \rightarrow \theta = (\frac{e^{s_1}}{\Sigma_{i=1}^{n} e^{s_i}} ... \frac{e^{s_k}}{\Sigma_{i=1}^{n} e^{s_i}})$$

- it is kind of a way to normalize everything so the likelihood over all classes sum to one and all values are positive
- how to predict class of a new entry $x \in \mathbb{R}^d$
  - first we compute the scores $x \rightarrow [< w_1, x >, , , < w_k, x >] \in R^d \rightarrow \theta$
  - then we take the soft max of $\theta, f(\theta)$
  - we can write the conditional probability for class y as

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

  - so we would take the arg max over y as the final prediction

4.3 review

- we have so far seen how to construct a model, so that our problem can be solved in a more principled probabilistic point of view
- the most important thing is to chose the output distribution bass ed on the input data, and takes that we have
- as well as the transfer function which maps a linear combination of the input to the parameter, then we cal learn it using mle and gradient descent
- this family of models are called generalized linear models
- the parameters of the function are linear , but the distribution themselves may not be linear.

5 generative models

5.1 review

- so far we have dealt with conditional models that is finding $P(y|x)$
- now we want to find the joint of y,x that is $P(x, y, \theta)$
- and we predict the labels as $argmax_y P(x, y, \theta)$

5.2 generative models through Bayes rule

- the joint distribution $P(x,y) = P(x|y)P(y)$ based on the observed labels we are generating our model around the input. we are generating probabilities about the input.
- often we can estimate the marginal $P(y)$
- and for testing proposes we go the other way ie $P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)}$ and then we can predict our label as $y = argmax_y P(x|y)P(y)$

5.3 naive Bayes model

lets consider the task of binary text classification of spam versus not spam

- it we take the text as a bag of words, that is we take each word put it it in a dictionary's, this dictionary says if we have this word or not, and the size of the dictionary is the number of words we have in total.
- so our dictionary is x of length d and $x_i = \{0,1\}$ indicates whether or not a word is in the text.
- so we can find $P(x|y) = P(x_1...x_d|y) = P(x_1|y)P(x_2|y,x_1)....P(x_d|y,x_1...x_d) = \Pi_{i=1}^{d} P(x_i|y,x_{<i})$
- the order of the x's does not matter we just need to have one
- but the issue, is that is very hard to model, because there are many levels of conditional assumptions meaning we need estimate many parameters

5.4 naive Bayes

- the naive Bayes assumption solved this by saying that features are conditionally independent given the label that is $P(x|y) = \Pi_{i=1}^{d} P(x_i|y)$
- this is a strong assumption, but it works well in practice.
- the $x's$ are only conditionally independent.

5.5 parameterize our model

- for a binary x_i (that is if a word is in the document or not) assume $P(x_i|y)$ is Bernoulli such that

$$P(x_i = 1|y = 1) = \theta_{i,1}, P(x_i = 1|y = 0) = \theta_{i,0}$$

and

$$P(y = 1) = \theta_0$$

- this allows us to write $P(y,x) = P(x|y)P(y) = P(y)\Pi_{i=1}^{d}P(x_i|y) = P(y)\Pi_{i=1}^{d}\theta_{i,y}\mathbb{I}\{x_i = 1\} + (1 - \theta_{i,y}\mathbb{I}\{x_i = 0\})$
- now we can write out likelihood of the dataset as

$$\mathcal{L}(\theta, d) = \Pi_{n=1}^{N}P(y^n)\Pi_{i=1}^{d}[\theta_{i,y^n}\mathbb{I}(x_i^n = 1) + (1 - \theta_{i,y^n}\mathbb{I}(x_i^n = 0)])$$

- and then we have $\ell(\theta) = log\mathcal{L} = log\Pi_{n=1}^{N}P(y^n)\Pi_{i=1}^{d}[\theta_{i,y^n}\mathbb{I}(x_i^n = 1) + (1 - \theta_{i,y^n}\mathbb{I}(x_i^n = 0)]) = \Sigma_{n=1}^{n}\Sigma_{i=1}^{d}log(\theta_{i,y^n}\mathbb{I}(x_i^n = 1) + (1 - \theta_{i,y^n}\mathbb{I}(x_i^n = 0))) + log(P(y^n))$
- then taking the derivative we find $\frac{\partial\ell}{\partial\theta_{1,j}} = \Sigma_{i=1}^{d}\Sigma_{n=1}^{N}\frac{\mathbb{I}\{x_i^n=1\}-\mathbb{I}\{x_i^n=0\}}{\theta_{i,y^n}\mathbb{I}\{x_i^n=1\}+(1-\theta_{i,y^n})\mathbb{I}\{x_i^n=0\}} = \Sigma_{n=1}^{N}\frac{\mathbb{I}\{x_j^n=1\}-\mathbb{I}\{x_j^n=0\}}{\theta_{j,y^n}\mathbb{I}\{x_j^n=1\}+(1-\theta_{j,y^n})\mathbb{I}\{x_j^n=0\}} = \Sigma_{n=1}^{N}\frac{\mathbb{I}\{y^n=1\wedge x_j^n=1\}}{\theta_{j,1}} - \frac{\mathbb{I}\{y^n=1\wedge x_j^n=0\}}{1-\theta_{j,1}}$
- setting this equal to zero and solving yields $\theta_{j,1} = \frac{\Sigma_{n=n}^{n}I(y^n=1\wedge x_j^n=1}{\Sigma_{n=1}^{n}\mathbb{I}(y^n=1)}$
- whhcih can be practically thought of as $\frac{\text{number of fake reviews with a word}}{\text{number of fake reviews}}$

5.6 review

- we need to make assumptions for generative models
- in the naive byes we assume that the inputs are conditionally independent given the labels
- recipe for a naive Bayes model
  - chose a distribution for $P(x_i|y)$
  - chose $P(y)$ often a categorical distribution
  - estimate the parameters using mle

5.7 naive bays with continuous features

- let us consider a classification task with continuous inputs $P(x_i|y{\sim}\mathcal{N}(\mu_{i,y}, \sigma_{i,y}^2)$ and $P(y = k) = \theta_k$
- so here we are assuming that our data is normally distributed around each input (in the case word)
- the likelihood of the data is thus.

$$P(\mathcal{D}) = \Pi_{n=1}^{n}P(y^n)\Pi_{i=1}^{d}P(x_i^n|y^n) = \Pi_{n=1}^{n}P(y^n)\Pi_{i=1}^{d}\frac{1}{\sqrt{2\pi}\sigma_{i,y^n}}e^{-\frac{1}{2\sigma_{i,y_n}^2}(x_i^n-\mu_{i,y^n})^2}$$

- then we can take the log as $\ell = \Sigma_{n=1}^{n}log(\theta_{y^n} + \Sigma_{n=1}^{n}\Sigma_{i=1}^{d}log(\frac{1}{\sqrt{2\pi}\sigma_{i,y^n}} - \frac{1}{2\sigma_{i,y^n}^2})(x_i^n - \mu_{i,y^n})^2$

- then we can find the partial as $\frac{\partial \ell}{\partial \mu_{i,j}} = \Sigma^{n}_{n:y^n=k} \frac{1}{\sigma^2_{j,k}} (x^n_j - \mu_{j,k})$
- and solving this yields $\mu_{i,j} = \Sigma_{n:y^n=k} \frac{x{j}^n}{\mathbb{I}(y^n=k)}$ which is the sample mean of x_j in class k.

5.8 decision boundary of a Gaussian NB model

- so the decision boundary of a model is given by the $log(\frac{P(y=1|x)}{p(y=0|x)}$ that is the log odds
- in general this is quadratic for Gaussian naive Bayes models
- how ever if we assume that all classes have the same variance then the decision boundary becomes linear

5.9 Gaussian naive Bayes vs logistic regression

- logistic regression

  modal type: conditional

  - characterization $P(y|x)$
  - we assume that $Y|x = x$ is Bernoulli
  - we make no assumptions about y.
  - our design boundary is $\theta^t x$

  - is a generative model
  - we are trying to find $P(x|y), P(y)$ we assume that $Y|x = x$ is Bernoulli
  - we assume that X is Gaussian
  - our decision boundary is $\theta^t x$
- asymptotically logistic regression and Gaussian naive Bayes converge to the same solution, if our assumption about the generative model, holds
- what happens if the GNB assumptions are false.? the two models will converge to different solutions, but the GNB will converge faster but have a worse over all error, while the logistic regression will converge slower but have a lower overall error
- but once again this all depends on our assumptions

5.10 review

- today we saw different approaches on how to model our problems, each has there own strengths and weakness and which is right to sue really depends on how well your assumptions AR meet
- in principle it is ah rd to chose in advance what is the Best way to learn the problem.