Stanford CS229 Intro to ML(2018)

Ian Poon

November 2024

"The course says that its prerequisites include only linear algebra, basic probability and statistics, but my friend Debnil told me that it's a lie. You need to be really good with linear algebra since the class is very mathy. If you take the course as your first introduction into machine learning, you're in for a rough time.", graduated Stanford CS ex-student on CS229.

Contents

1 Appendix: Hoeffding lemma and inequality

Recall from undergraduate probability `MIT 18.600` intro to probability, basic inequalities like **markov inequality** and **chebyshev inequality** which we state without proof as a refresher here.

Theorem 1 (Markov Inequality)

Let $Z \geq 0$ be a non negative random variable then for all $t \geq 0$

$$\mathbb{P}(Z \geq t) \leq \frac{\mathbb{E}[Z]}{t}$$

Theorem 2 (Chebyshev's inequality)

Let Z be any random variable with $\text{Var}(Z) < \infty$ then

$$\mathbb{P}(Z \geq \mathbb{E}[Z] + t \text{ or } Z \leq \mathbb{E}[Z] - t) \leq \frac{\text{Var}(Z)}{t^2}$$

for $t \geq 0$.

Note that in 18.600 we expressed the same thing as

$$P\{|X - \mu| \geq k\} \leq \frac{\sigma^2}{k^2}$$

and then after it can be used to derive the law of weak numbers of course

Definition 3

The **moment generating function** of Z is

$$M_Z(\lambda) = \mathbb{E}[\exp(\lambda Z)]$$

Note the choice of $\lambda > 0$ is arbitrary

Proposition 4 (Chernoff bounds)

Let Z be any random variable. Then for any $t \geq 0$

$$\mathbb{P}(Z \geq \mathbb{E}[Z] + t) \leq \min_{\lambda \geq 0} \mathbb{E}[e^{\lambda(Z - \mathbb{E}[Z])}]e^{-\lambda t} = \min_{\lambda \geq 0} M_{Z - \mathbb{E}[Z]}(\lambda)e^{-\lambda t}$$

and

$$\mathbb{P}(Z \leq \mathbb{E}[Z] - t) \leq \min_{\lambda \geq 0} \mathbb{E}[e^{\lambda(\mathbb{E}[Z] - Z)}]e^{-\lambda t} = \min_{\lambda \geq 0} M_{\mathbb{E}[Z] - Z}(\lambda)e^{-\lambda t}$$

Proof. In 18.600 you have already proven(just let $X = Z - \mathbb{E}[Z]$)

$$P\{X \geq a\} \leq e^{-ta}M(t) \quad \forall t > 0$$
$$P\{X \leq a\} \leq e^{-ta}M(t) \quad \forall t < 0$$

and because this applies for all $\lambda > 0$ taking the minimum on the RHS will not affect the inequality. □

Next recall the moment generating function for an arbitrary normal random variable from 18.600. Given $Z \sim \mathcal{N}(0, \sigma^2)$ then we know

$$\mathbb{E}(\exp(\lambda Z)) = \exp\left(\frac{\lambda^2\sigma^2}{2}\right)$$

Now that we are done with our refresher of undergrad probability theory consider some applications related to our purposes

Proposition 5

A **Rademacher random variable** or **random sign variable** is defined such that $S = 1$ with probability $\frac{1}{2}$ and $S = -1$ also with probability $\frac{1}{2}$. Then

$$\mathbb{E}_S[e^{\lambda S}] \leq \exp\left(\frac{\lambda^2}{2}\right)$$

Proof. Consider the taylor expansion of the exponential function($e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$

$$\begin{aligned}\mathbb{E}[e^{\lambda S}] &= \sum^{\infty} \frac{\lambda^k \mathbb{E}[S^k]}{k!} \\ &= \sum_{k=0,2,4\ldots} \frac{\lambda^k}{k!} = \sum_{k=0}^{\infty} \frac{\lambda^{2k}}{(2k)!}\end{aligned}$$

where the second equality follows because $E[S^k] = (1)^k 0.5 + (-1)^k 0.5$ which is clearly 1 when k even 0 when k odd. Therefore all the odd k terms disappear. Now using the fact that $(2k)! \geq 2^k \cdot k!$. To see this simply take $(2k)!/k! \geq 2^k \Rightarrow \underbrace{(2k)(2k-1)\ldots(k)}_{\text{k terms}} \geq (2)^k$. Then it is clear to see that every of the k LHS terms is bigger than equal to the RHS terms for $k = 2$ and above. You may verify that we have equality for $k = 1, 0$.So we have

$$\mathbb{E}[e^{\lambda S}] \leq \sum_{k=0}^{\infty} \frac{(\lambda^2)^k}{2^k \cdot k!} = \sum_{k=0}^{\infty} \left(\frac{\lambda^2}{2}\right)^k \frac{1}{k!} = \exp\left(\frac{\lambda^2}{2}\right)$$

□

Lemma 6

Let Z be a bounded random variable with $Z \in [a, b]$ then

$$\mathbb{E}[\exp(\lambda(Z - \mathbb{E}[Z]))] \leq \exp\left(\frac{\lambda^2(b-a)^2}{8}\right)$$

for all $\lambda \in \mathbb{R}$

Proof. Recall jensen inequality also from 18.600 that states for convex function $f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$. This is applicable in our context because $f(t) = \exp(t)$ and $f(t) = \exp(-t)$ are convex functions. Well thats because their second derivatives are both bigger than zero. Recall $\exp x, \exp -x > 0$ as they have an asymptote $y = 0$. Now we use a

technique apparently common in probability theory known as **symmetrization**(not that I heard off). First let Z' be an independent copy of Z with the same distribution so that $Z' \in [a, b]$ and $\mathbb{E}[Z'] = \mathbb{E}[Z]$. Then we have

$$\mathbb{E}_Z[\exp(\lambda(Z - \mathbb{E}_Z[Z]))] = \mathbb{E}_Z[\exp(\lambda(Z - \mathbb{E}_{Z'}[Z']))] \leq \mathbb{E}_Z[\mathbb{E}_{Z'} \exp(\lambda(Z - Z'))]$$

where the last equality follows because

$$\mathbb{E}_Z[\exp(\lambda(Z - \mathbb{E}_{Z'}[Z']))] = \mathbb{E}_Z[\exp(\mathbb{E}_{Z'}[\lambda(Z - Z')])]$$

But we also know that $\mathbb{E}[Z - Z'] = 0$ therefore the normal distribution is symmetric about the mean 0. So $S(Z - Z')$ has the same distribution as $Z - Z'$ where $S \in \{-1, 1\}$ is the rademacher variable. So we have

$$\mathbb{E}_{Z,Z'}[\exp(\lambda(Z - Z'))] = \mathbb{E}_{Z,Z',S}[\exp(\lambda S(Z - Z'))] = \mathbb{E}_{Z,Z'}[\mathbb{E}_S[\exp(\lambda S(Z - Z'))|Z, Z']]$$

recall from `MIT 18.600` this follows by

$$\mathbb{E}[X] = \mathbb{E}_Y[\mathbb{E}_X[X|Y]] = \sum_y \mathbb{E}_X[X|Y = y]p_Y(y)$$

(conditional expectation) but we also know that

$$\mathbb{E}_S[\exp(\lambda S(Z - Z'))|Z, Z'] \leq \exp\left(\frac{\lambda^2(Z - Z')^2}{2}\right)$$

from 5. Notice that this makes sense since $\mathbb{E}_S$(is with respect to S) and $[\ldots|Z, Z']$ indicate that Z, Z' is treated as some constant. This is analagous to $\mathbb{E}_X[X|Y = y]$ in our conditional expectation above. Moreover we know that $Z, Z' \in [a, b]$ so $|Z - Z'| \leq (b - a)$ hence

$$\mathbb{E}_{Z,Z'}[\exp(\lambda(Z - Z'))] = \mathbb{E}_S[\exp(\lambda S(Z - Z'))|Z, Z'] \leq \exp\left(\frac{\lambda^2(b - a)^2}{2}\right)$$

where the first equality follows by the terms in red above □

Remark 7. *Note that we only proved for a factor of* 2 *instead of* 8 *but that will suffice for our purposes apparently. From now on we will just assume the stronger case of* 8 *is true by a similar proof*

Theorem 8 (Heoffding's inequality)

Let $Z_1, \ldots, Z_n$ be independent bounded random variables with $Z_i \in [a, b]$ for all i where $-\infty < a \leq b < \infty$. Then

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}[Z_i]) \geq t\right\} \leq \exp\left(-\frac{2nt^2}{(b - a)^2}\right)$$

and

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mathbb{E}[Z_i]) \leq -t\right\} \leq \exp\left(-\frac{2nt^2}{(b - a)^2}\right)$$

for all $t \geq 0$

Proof. By 6 we and chernoff bounds we have

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i-\mathbb{E}[Z_i])\geq t\right) &= \mathbb{P}\left(\sum_{i=1}^{n}(Z_i-\mathbb{E}[Z_i])\geq nt\right)\\ &\leq \mathbb{E}\left[\exp\left(\lambda\sum_{i=1}^{n}(Z_i-\mathbb{E}[Z_i])\right)\right]e^{-\lambda nt}\\ &= \left(\prod_{i=1}^{n}\mathbb{E}[e^{\lambda(Z_i-\mathbb{E}[Z_i])}]\right)e^{-\lambda nt} \leq \left(\prod_{i=1}^{n}e^{\frac{\lambda^2(b-a)^2}{8}}\right)e^{-\lambda nt}\end{aligned}$$

Now like how in our discussion regarding chernoff bounds earlier, the choice of $\lambda > 0$ is arbitrary so we may minimize over $\lambda > 0$ to obtain

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i-\mathbb{E}[Z_i])\geq t\right) \leq \min_{\lambda\geq 0}\exp\left(\frac{n\lambda^2(b-a)^2}{8}-\lambda nt\right) = \exp(-\frac{2nt^2}{(b-a)^2})$$

where the last equality is found by taking the 1st derivative of $\frac{n\lambda^2(b-a)^2}{8}-\lambda nt$ and setting it to zero, where we will find that $\lambda = \frac{4t}{(b-a)^2}$. Then substituting this value of λ back into $\frac{n\lambda^2(b-a)^2}{8}-\lambda nt$ yields the desired relation

2 Appendix: Gaussians

Definition 9

A vector valued random variable $X = [X_1 \dots X_n]^T$ is said to have a **multivariate normal(or Gaussian distribution)** with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\sum \in S_{++}^n$ if its probability desnity function is given by

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}\left|\Sigma\right|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)\right)$$

We write this as $X \simeq \mathcal{N}(\mu,\Sigma)$. Note that

$$S_{++}^n = \left\{A \in \mathbb{R}^{n\times n} : A = A^T \text{ and } x^TAx > 0 \text{ for all } x \in \mathbb{R}^n \text{ such that } x \neq 0\right\}$$

that is the set of positive definite matrices(we will verify this below)

To make sense of this first recall the **univariate gaussian distribution**

$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

where we must have

$$\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}\exp\left(-\frac{1}{2\sigma}(x-\mu)^2\right) = 1$$

for this to be a well defined probability density function.we say $\frac{1}{\sqrt{2\pi\sigma^2}}$ is the **normalization factor**

Hence analagously $\frac{1}{(2\pi)^{n/2}|\sum|^{1/2}}$ is the normalization factor for the multivariate case that is

$$\frac{1}{(2\pi)^{n/2}\left|\Sigma\right|^{1/2}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)dx_1dx_2\dots dx_n = 1$$

Proposition 10

For any random vector X with mean μ and covariance matrix Σ

$$\Sigma = E[(X-\mu)(X-\mu)^T] = E[XX^T] - \mu\mu^T$$

Proposition 11

Suppose that Σ is the covariance matrix corresponding to some random vector X. Then Σ is symmetric positive semidefinite(as stated in the defintion above)

Example 12

Consider the case of the **bivariate gaussian**(that is where n=2). So we have

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

notice that Σ is diagonal since recall for independent variables, $\text{Cov}(X,Y) = 0$ if $X \neq Y$ else we have $\text{Cov}(X,X) = \text{Var}(X)$(which are precisely the elements on the diagonal). Then we have

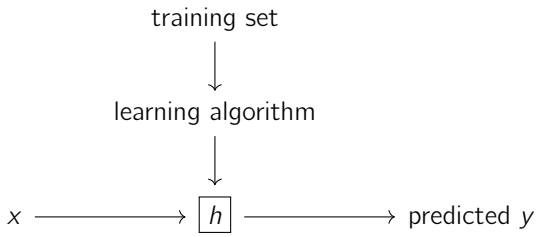
$$\begin{aligned} p(x;\mu,\Sigma) &= \frac{1}{2\pi \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^T \begin{bmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2 - \frac{1}{2\sigma_2^2}(x_2-\mu_2)^2 \right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2 \right) \cdot \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2\sigma_2^2}(x_2-\mu_2)^2 \right) \end{aligned}$$

And this is to be expected! Recall from `MIT 18.600 Intro To Prob` a for a joint probability distribution function $p_{X\cap Y} = p_{X,Y}(x,y) = p_X(x)p_Y(y)$ if X, Y are independent

Fact 13 (Notation)

Consider

- $x^{(i)}$ for input variables(**input features**)
- $y^{(i)}$ for output variables(**target variables**)
- $(x^{(i)}, y^{(i)})$ called a **training example**
- the data set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ is called the **training set**
- let $\mathcal{X}$ denote the space of input values and $\mathcal{Y}$ denote the space of output values
- the function $h : \mathcal{X} \to \mathcal{Y}$ is called a **hypothesis**



Fact 14

When the target variable that we're trying to predict is **continuous** we call the learning problem a **regression** problem. If instead y is **discrete** we say it is a **classification** problem

3 Linear Regression

Example 15

Suppose we let our hypothesis be an approximation of y as a linear function of x. Say $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots$. To simply this into a simple sum we assume $x_0 = 1$ which corresponds to the **intercept term** and write

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

where we are viewing θ(the **parameters/weights** and x hs n dimensional vectors.)

Given a training set we define the cost function to help us pick/learn the best paramters θ. One reasonable way is to define the cost function by how close $h(x)$ is to y

Definition 16

The **cost function**

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)})^2$$

3.1 LMS Algorithm

We want to choose θ to minimize $J(\theta)$ to do so we consider **gradient descent** which repeatedly performs

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

starting at some initial θ where α is known as the **learning rate**. Essentially it repeatedly takes a step in the direction of the steepest decrease of J. Now let us try to work out the expression for this algorithm

$$\begin{aligned}\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2 \\ &= (h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\ &= (h_\theta(x)-y)x_j\end{aligned}$$

So for a single training example we have the **update rule**

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

In particular this what we call the **LMS**(least mean squares) update rule. So far we have considered the LMS rule for when there is only a single training example. Let us extend this to more than one via 2 ways.

(i) Firstly we can replace it with

```
Repeat until convergence{
    θ_j = θ_j + α Σ_{i=1}^{m} (y^(i) − h_θ(x^(i))) x_j^(i)  (for every j)
}
```

Which basically what it does for a single training example to the whole training set. This is what we call **batch gradient descent**

(ii) Alternatively we can replace it with

```
Loop{
    for i = 1 to m,{
        θ_j = θ_j + α(y^(i) − h_θ(x^(i))) x_j^(i)  (for every j)
    }
}
```

This is what we call **stochastic(incremental) gradient descent**. Notice how every update is only according to the gradient of the error with respect to that single training example only(notice there is no $\sum_{i=1}^{m}$)

With this we should

Fact 17

Observe that the difference between batch and stochastic gradient descent is that batch batch scans through the entire training set before making a step while the stochastic gradient descent starts making progress right away. For the reason due to time contraints, stochastic gradient descent is often preferred over batch for large datasets

To learn more about the theory behind gradient descent and other optimization methods relevant in these notes like newton method and coordinate ascent please refer to your `MIT 6.7220 Non Linear Analysis(2024)` notes

3.2 normal equations

Gradient descent is one way of minimizing J. We now discuss a second way of doing so, this time performing the minimization explicitly and without resorting to an iterative algorithm(i.e we want a closed form solution instead of

some recursively defined one). First consider some relevant matrix calculus concepts

Definition 18

For a function $f : \mathbb{R}^{m\times n} \mapsto \mathbb{R}$ mapping from $m \times n$ matrices to the real numbers we define the derivative of f with respect to A to be

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

Proposition 19

let A be an $n \times m$ while B be an $m \times n$ matrix so AB is a $n \times n$ square matrix and BA is a $m \times m$ square matrix

$$\operatorname{tr} AB = \operatorname{tr} BA$$

Proof. Consider

$$\operatorname{tr} AB = \sum_i^n (AB)_{ii} = \sum_i^n \sum_j^m (A)_{ij}(B)_{ji}$$

and

$$\operatorname{tr} AB = \sum_j^m (BA)_{jj} = \sum_j^m \sum_i^n (B)_{ji}(A)_{ij}$$

clearly these 2 sums are the same

Corollary 20

We have

1. $\operatorname{tr} ABC = \operatorname{tr} CAB = \operatorname{tr} BCA$
2. $\operatorname{tr} A = \operatorname{tr} A^T$
3. $\operatorname{tr}(A + B) = \operatorname{tr} A + \operatorname{tr} B$
4. $\operatorname{tr} aA = a \operatorname{tr} A$

Proof. For (1) consider that this is an extension of the previous proposition and note that only cyclic permutations will be guaranteed to result in a square matrix. Just consider if $A = n \times m$, $B = m \times l$, $C = l \times n$ matrices. The rest are very trivial(the diagonal is unchanged or linear in those operations).

Proposition 21

We have

$$\nabla_A \operatorname{tr} AB = B^T \tag{1}$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T \tag{2}$$

$$\nabla_A \operatorname{tr} ABA^T C = CAB + C^T AB^T \tag{3}$$

$$\nabla_A |A| = |A| (A^{-1})^T \tag{4}$$

where $|A|$ denotes the determinant of A

Proof. Consider that for 1 we have

$$\operatorname{tr}(AB) = \sum_{i,j} A_{ij} B_{ji}$$

and so

$$\frac{\partial}{\partial A_{ij}} \operatorname{tr}(AB) = B_{ji}$$

For 2 consider

$$(\nabla_{A^T} f(A))_{ij} = \frac{\partial f}{\partial (A^T)_{ij}} = \frac{\partial f}{\partial A_{ji}} = (\nabla_A f(A))_{ji} .$$

For 3 consider that by product rule, treating $\bullet$ as a variable and the rest as constants we have

$$\begin{aligned}
\nabla_A \operatorname{tr} ABA^T C &= \nabla_\bullet \operatorname{tr} \bullet BA^T C + \nabla_\bullet \operatorname{tr} AB \bullet^T C \\
&= (BA^T C)^T + \nabla_\bullet (CAB\bullet^T) \\
&= (BA^T C)^T + \nabla_\bullet (CAB\bullet^T)^T \\
&= (BA^T C)^T + \nabla_\bullet (\bullet B^T A^T C^T) \\
&= (BA^T C)^T + (B^T A^T C^T)^T \\
&= C^T AB^T + CAB
\end{aligned}$$

For 4 from `linear algebra` that the formula for inverse is given

$$A^{-1} = (A')^T / |A|$$

so

$$A' = |A| \, (A^{-1})^T$$

where A' is the cofactor matrix. But see that since $|A| = \sum_j A_{ij} A'_{ij}$, we then have

$$\frac{\partial}{\partial A_{ij}} |A| = A'_{ij}$$

then clearly the proposition follows

3.3 least squares revisited

armed with the tools of matrix derivatives let us now proceed to find in closed form the value of θ that minimizes $J(\theta)$ as promised. First we define the **design matrix** X that contains the *training example*s' input values in its row

$$X = \begin{bmatrix} - & (x^{(1)})^T & - \\ - & \vdots & - \\ - & (x^{(m)})^T & - \end{bmatrix}$$

Let $\overrightarrow{y}$ be the m dimensional vector containing all the target values from the training set

$$\overrightarrow{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Now since $h(x^{(i)}) = (x^{(i)})^T\theta$ we see that

$$X\theta - \vec{y} = \begin{bmatrix} (x^{(1)})^T\theta - y^{(1)} \\ \vdots \\ (x^{(m)})^T\theta - y^{(m)} \end{bmatrix} = \begin{bmatrix} h(x^{(1)}) - y^{(1)} \\ \vdots \\ h(x^{(m)}) - y^{(m)} \end{bmatrix}$$

So it can be seen that

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y})$$

combining (2) and (3) in 21 we have

$$\nabla_{A^T} \operatorname{tr} ABA^TC = B^TA^TC^T - BA^TC$$

hence

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) && (1)\\ &= \frac{1}{2}\nabla_\theta(\theta^TX^TX\theta - \theta^TX^T\vec{y} - \vec{y}^TX\theta + \vec{y}^T\vec{y}) && (2)\\ &= \frac{1}{2}\nabla_\theta \operatorname{tr}(\theta^TX^TX\theta - \theta^TX^T\vec{y} - \vec{y}^TX\theta + \vec{y}^T\vec{y}) && (3)\\ &= \frac{1}{2}(X^TX\theta + X^TX\theta - 2X^T\vec{y}) && (4)\\ &= X^TX\theta - X^T\vec{y} && (5) \end{aligned}$$

where (3) follows because the trace of a real number is just the real number. For (4) $\vec{y}^T\vec{y}$ disappears because it does not θ so it just treated as a constant. The blue terms is an application of 3.3. While the black term follows because $\operatorname{tr} A = \operatorname{tr} A^T$. To minimize $J(\theta)$ we now equate the above with 0 where we finally find the so called **normal equations**

$$X^TX\theta = X^T\vec{y}$$

where

$$\boxed{\theta = (X^TX)^{-1}X^T\vec{y}}$$

is the value of θ that minimizes $J(\theta)$

3.4 probabilistic interpretations

One might question now,

Question 22. *So when faced with a regression problem, when might linear regression specifically the least squares cost function J be a reasonable choice?*

In this section we discussion certain probabilistic assumptions that justify such an approach. Assume that the target variables and inputs are related via

$$y^{(i)} = \theta^Tx^{(i)} + \varepsilon^{(i)}$$

where $\varepsilon^{(i)}$ is an error term. we also further assume that $\varepsilon^{(i)}$ is IID(independently and identically distributed) according to a Gaussian distribution(also called a normal distribution) with mean zero and some variance σ^2. That is to say

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

Therefore by definition of normal random variable

Definition 23

the probability density of $\varepsilon^{(i)}$ is given by

$$p(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2}\right)$$

.

We know that

$$\mathbb{E}(y^{(i)}) = \mathbb{E}(\varepsilon^{(i)}) + \theta^T x^{(i)} = \theta^T x^{(i)}$$

recall properties of expectation where $\mathbb{E}(ax + b) = a\mathbb{E}(x) + b$. Similarly for variance we have

$$\text{var}(y^{(i)}) = \text{var}(\varepsilon^{(i)}) = 0$$

therefore the probability density of $y^{(i)}$ is

$$p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

the notation $p(y^{(i)}|x^{(i)}; \theta)$ indicates that this is the distribution of $y^{(i)}$ given $x^{(i)}$ and paramterized by θ. So we may also write the distribution of $y^{(i)}$ as

$$y^{(i)}|x^{(i)}; \theta \simeq \mathcal{N}(\theta^T x^{(i)}, \sigma^2)$$

(see this as a translation from of $\theta^T x^{(i)}$ from the distribution above hence the new value of the mean).

We would now like to express this probability density function explicitly as a function of θ instead

Definition 24

We define the **likelihood function**

$$L(\theta) = L(\theta; X, \overrightarrow{y}) = p(\overrightarrow{y}|X; \theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta)$$

The last term makes sense because by assumption the $\varepsilon^{(i)}$s are independent(and hence also the $y^{(i)}$s given the $x^{(i)}$s). Recall the definition of independent joint probability density functions from `MIT 18.600`.

Fact 25

Now consider that essentially we have a design matrix X which contains both $x^{(i)}$'s and θ's. So clearly we want to find the θ's to maximize the probability of $y^{(i)}$'s which represents the correct output corresponding to the inputs $x^{(i)}$. From the above we see that maximizing $p(\vec{y}|X; \theta)$ is equivalent to maximizing $L(\theta)$ which is the likelihood function.

To maximize $L(\theta)$, we could maximize its logarithm instead to make calculations easier. To that end we define the

Definition 26

the **log likelihood** $\ell(\theta)$:

$$\begin{aligned}\ell(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2\end{aligned}$$

in which case the task of maximizing $\ell(\theta)$ is the same as maximizing

$$\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2$$

which we recognize to be $J(\theta)$ in original least squares cost function therefore justifying our use of $J(\theta)$ for our regression problem.

Remark 27. *Note that the use of such a probabilistic assumptions makes the task of least squares regression equivalent to finding the maximum likelihood estimate of θ. This is just one possible assumption that justifies this and is no means a necessary condition(as in the only)*

4 classification and logistic regression

Having talked about regression(i.e for continuous target variables if you recall) we move on to classification(i.e for discrete target variables if you recall)

For now we focus our discussion specifically on **binary classification** problems in which y only takes the values 0 or 1

Definition 28

Some terminology matters. We say 1 is the **positive class** while 0 is the **negative class**. Given $x^{(i)}$, the corresponding $y^{(i)}$ is also called the **label** of the training example

First we justify why we even need to to have 2 separate problem solving paradigms, classification for discrete and regression for continuous.

Question 29. *Can we just let our hypothesis be the same linear function we used in our old linear regression model? i.e treat the discrete case in the same way as continuous?*

Suppose we do so, that is we approach the classification problem ignoring the fact that y is discrete valued. However our old definition of $h_\theta(x)$ in 15 may not make sense as it may take values larger than 1 or smaller than 0 so is not appropriate for our *binary classification* problem where $y^{(i)}$ is 0 or 1. To fix this we need to change the form of our hypothesis.

Definition 30

Define

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

is called the **logistic function** or the **sigmoid function**

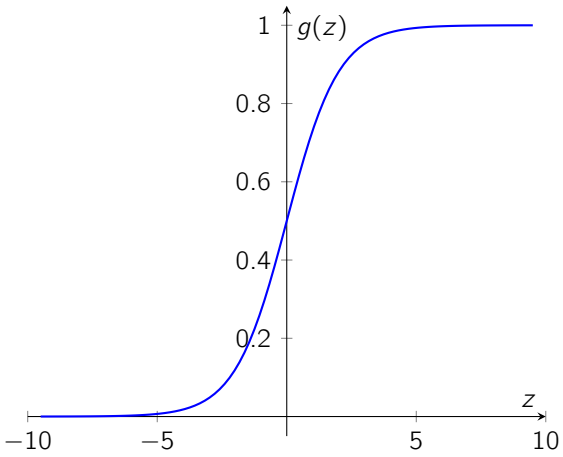


Figure 1: Sigmoid function

Notice that $g(z)$ tends towards 1 as $z \to \infty$ and towards 0 as $z \to -\infty$. That is to say that $h(x)$ is always bounded between 0 and 1 as desired.

Fact 31

Note the following usefu property of the sigmoid function

$$\begin{aligned} g'(z) &= \frac{d}{dz}\frac{1}{1 + e^{-z}} \\ &= \frac{1}{(1 + e^{-z})^2}(e^{-z}) \\ &= \frac{1}{(1 + e^{-z})} \cdot \underbrace{\left(1 - \frac{1}{(1 + e^{-z})}\right)}_{\frac{e^{-z}}{1+e^{-z}}} \\ &= g(z)(1 - g(z)) \end{aligned}$$

So given the logistic regression model how do we fit θ for it? Just like how we did for the least squares let us make a few probabilistic assumptions that make justifies its use and then we fit the paramters via a method known as *maximum likelihood*(essentially . To this end, first assume that

$$\begin{aligned} P(y = 1|x; \theta) &= h_\theta(x) \\ P(y = 0|x; \theta) &= 1 - h_\theta(x) \end{aligned}$$

which can be written more compactly as

$$p(y|x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

to see this just simply sub in the values when $y=0$ and $y=1$ respectively to see the equivalence. In other words for our binary classification case here, we are assuming $y|x;\theta$ follows a **Bernoulli** distribution(in contrast to normal in the previous least squares regression case)

Now assuming the m training examples were generated *independently* just like before we have

$$L(\theta) = p(\overrightarrow{y}|X;\theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m}(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}$$

and again like before it would be easier to maximize the log likihood

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \log h(x^{(i)}) + (1-y^{(i)})\log(1-h(x^{(i)}))$$

Recall when we minimized the log likihood for the least squares regression case, we found it to be equivalent to minimizing the least mean squares cost function in which case we got the LMS update rule. It turns out in this case the *stochastic gradient ascent* update rule looks similar

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

but it is not since now $h_\theta(x^{(i)})$ is a non linear function as opposed to $\theta^T x^{(i)}$ from before. Recall that our hypothesis now uses the sigmoid function now. To see how the above was derived consider

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\ell(\theta) &= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right)\frac{\partial}{\partial\theta_j}g(\theta^T x) \\
&= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right)g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial\theta_j}\theta^T x \\
&= (y(1-g(\theta^T x)) - (1-y)g(\theta^T x))x_j \\
&= (y-h_\theta(x))x_j
\end{aligned}$$

where the second line follows by 31

5 generalized linear models

We will learn soon that in fact the 2 examples, one regression and classification, that we have considered so far in fact belong to a broader family of models called **generalized linear models**(GLM)s

5.1 the exponential family

To work our way up to GLMs(generalized linear models) we first define

Definition 32

we first define the class of distributions that is the **exponential family** which are in the form

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

here η is called the **natural paramter**(or the **canonical paramter**) of the distribution. $T(y)$ is the **sufficient statistic** and $a(\eta)$ is the **log partition function**

That is to say a fixed choice of T, a, b defines a *family* of distributions parametrized by η. As we vary η we get different distributions within this family.

Example 33 (Bernoulli distribution)

The **bernoulli distribution** is in fact an example of the exponential family. That is to say there exists a choice of T, a, b It is defined by the equations above. First see that

$$\begin{aligned} p(y;\phi) = p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\ &= \exp(y\log\phi + (1-y)\log(1-\phi)) \\ &= \exp\left(\left(\log\left(\frac{\phi}{1-\phi}\right)\right)y + \log(1-\phi)\right) \end{aligned}$$

I think there is a typo in the notes log clearly refers to ln but the same concept applies. From which we see that

$$\eta = \log(\phi/(1-\phi))$$

where we get

$$\phi = \frac{1}{1+\varepsilon^{-\eta}}$$

(this happens to be sigmoid function!). Then we also see that

$$\begin{aligned} T(y) &= y \\ a(\eta) &= -\log(1-\phi) = \log(1+\varepsilon^{\eta}) \\ b(y) &= 1 \end{aligned}$$

Example 34 (Gaussian distribution)

The **Gaussian distribution** is in fact an example of the exponential family. That is to say there exists a choice of T, a, b It is defined by the equations above.

$$p(y;\mu) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(y-\mu)^2\right) = \frac{1}{\sqrt{2\pi}\exp\left(-\frac{1}{2}y^2\right)}\cdot\exp\left(\mu y - \frac{1}{2}\mu^2\right)$$

From which we may similary further simplify and compare with the exponential family and see that

$$\begin{aligned} \eta &= \mu \\ T(y) &= y \\ a(\eta = \mu^2/2 &= \eta^2/2 \\ b(y) &= (1/\sqrt{2\pi})\exp(-y^2/2) \end{aligned}$$

5.2 Constructing GLMs

Fact 35

Consider a classification or regression problem where we would like to predict the value of some random variable y as a function of x. To derive the GLM for this problem we first make three assumptions

1. $y|x;\theta$ is a member of the exponential family(η). That is given x, θ the distribution of y follows some exponential family distribution with parameter η.
2. Given x our goals is to predict the **expected value** of $T(y)$ *given* x.For instance, in most of our examples we will have $T(y) = y$. In that case means our hypothesis will then be $h(x) = E[y|x]$
3. The natural parameter η and the inputs x are related linearly $\eta = \theta^T x$(Or, if η is vector valued then $\eta_i = \theta_i^T x$)

5.3 Ordinary Least Sqaures

We will illustrate how a GLM is contructed by considering the example of *ordinary least squares*

Consider the scenario where the target variable y(called the **response variable** in GLM terminology) is continuous and we model the conditional distribution of y given x as a gaussian $\mathcal{N}(\mu, \sigma^2)$(which recall is a part of the exponential family). Hence

$$h_\theta(x) = E[y|x;\theta] = \mu = \eta = \theta^T x$$

where the first equality follows from assumption 2, the 2nd equality from the fact that $y|x;\theta \sim \mathcal{N}(\mu, \sigma^2)$, the 3rd equality from assumption 1 and our earlier discussions that $\mu = \eta$ in the formulation of Gaussian as an exponential family distribution and the last equality from assumption 3.

5.4 Softmax Regression

Consider a classification problem in which the reponse variable y can take on any one of k values so $y \in \{1, 2 \ldots, k\}$. For example classifying email into spam,personal and work-related. The response variable is still discrete but now takes on more than two values so we model it according to multinomial distribution.

Now we define k parameters $\phi_1, \ldots, \phi_k$ which specificies the probabilities of each of the k outcomes But we only require $k-1$ of the ϕ_i's knowing that $\sum_{i=1}^{k} \phi_i = 1$ for paramters to be independent.So we have

$$\phi_i = p(y = i, \phi), i = 1, \ldots, k-1 \quad \text{and} \quad p(y = k, \phi) = 1 - \sum_{i=1}^{k-1} \phi_i$$

Next we define $T(y) \in \mathbb{R}^{k-1}$ as follows:

$$T(1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(2) = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \ldots T(k-1) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, T(k) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Now notice that $(T(y))_i = \mathbf{1}\{y = i\}$. To see this just consider what the value of $(T(y))_i$(i.e the ith element of the vector $T(y)$) if $i \neq y$ and if $i == y$

Remark 36. *Unlike our previous examples we do not have $T(y) = y$!! Also note that $\mathbf{1}\{\ldots\}$ is the indicator function*

Problem 37

Show that the multinomial is a member of the exponental family. That is to say there exists a choice of T, a, b It is defined by the equations above.

we first consider that we have

$$\begin{aligned}
p(y;\phi) &= \phi_1^{\mathbf{1}(y=1)}\phi_2^{\mathbf{1}(y=2)}\ldots\phi_k^{\mathbf{1}(y=k)} \\
&= \phi_1^{\mathbf{1}(y=1)}\phi_2^{\mathbf{1}(y=2)}\ldots\phi_k^{1-\sum_{i=1}^{k-1}\mathbf{1}(y=i)} \\
&= \phi_1^{(T(y))_1}\phi_2^{(T(y))_2}\ldots\phi_k^{1-\sum_{i=1}^{k-1}(T(y))_i} \\
&= \exp\left((T(y))_1\log(\phi_1) + (T(y))_2\log(\phi_2) + \ldots + \left(1 - \sum_{i=1}^{k-1}((T(y))_i\right)\log(\phi_1)\right) \\
&= \exp\left((T(y))_1\log(\phi_1/\phi_k) + (T(y))_2\log(\phi_2/\phi_k) + \ldots + (T(y))_{k-1}\log(\phi_{k-1}/\phi_k) + \log(\phi_k)\right) \\
&= b(y)\exp(\eta^T T(y) - a(\eta))
\end{aligned}$$

where the forth line follows since $x^y = e^{y\ln x}$ and recall Andrew uses ln as log.Therefore we have

$$\eta = \begin{bmatrix} \log(\phi_1/\phi_k) \\ \log(\phi_2/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{bmatrix}$$

and

$$a(\eta) = -\log(\phi_k)$$

and

$$b(y) = 1$$

□

Definition 38

From above, we may define link function is given for $i = 1, \ldots, k$ by

$$\eta_i = \log\frac{\phi_i}{\phi_k}$$

For convenience we have also defined $\eta_k = \log(\phi_k/\phi_k) = 0$ To invert the link function and derive the response function, we therefore have that

$$e^{\eta_i} = \frac{\phi_i}{\phi_k} \tag{1}$$

$$\phi_k e^{\eta_i} = \phi_i \tag{2}$$

$$\phi_k\sum_{i=1}^{k} e^{\eta_i} = \sum_{i=1}^{k}\phi_i = 1 \tag{3}$$

But this implies that

$$\phi_k = 1/\sum_{i=1}^{k} e^{\eta_i}$$

which upon substitution into (2) we obtain

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$

This mapping from η's to ϕ's is known as the **softmax** function

To complete the model, by assumption 3 of 35 we know that η_i's are linearly related to x's that is we have $\eta_i = \phi_i^T x$ for $i = 1, \ldots, k-1$ where $\theta_1, \ldots, \theta_{k-1} \in \mathbb{R}^{n+1}$ are the paramters of our model. Therefore since $\eta_k = 0$(recall above) it implies

$$\eta_k = \theta_k^T x = 0$$

Therefore we have

$$\begin{aligned} p(y = i|x;\theta) &= \phi_i \\ &= \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} \\ &= \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} \end{aligned}$$

6 Generative Learning Algorithms

So far we've mainly been talking about learning algorithms that model $p(y|x;\theta)$, the conditional distribution of y given x. We now consider a different type of learning algorithm.

Definition 39

Algorithms that try to learn $p(y|x)$ directly(such as logistic regression) or algorithms that try to learn mappings directly from the space of inputs η to the labels $\{0, 1\}$(such as the perceptron algorithm) are called **discriminative** learning algorithms.

In contrast algorithms that instead try to *model* $p(x|y)$ are called **generative** learning algorithms.

We look at some examples of generative leanring algorithms to make sense of this

6.1 Gaussian discriminant analysis

The first generative algorithm we'll look at is the **Gaussian discriminant analysis**(GDA). In this model as the name suggests we'll assume that $p(x|y)$ is distributed according to the multivariate normal distribution.Refer to the appendix for a quick recap on gaussian distribution properties if necessary...to be continued(blah blah ill just see the problem sets)

7 support vector machines

Fact 40

Notation matters: we now introduce some new notation for talking about classification. We now write our linear classifier with paramters w, b as

$$h_{w,b}(x) = g(w^T x + b)$$

where here $g(z) = 1$ if $z \geq 0$ and $g(z) = -1$ otherwise. Also note that b now takes the role that was once θ_0 in 15 while w takes the role of $[\theta_1 \dots \theta_n]^T$

Next we assume for our binary classification problem $y \in \{-1, 1\}$ instead of $\{0, 1\}$ for convenience in later calculations. So essentially note that these are still binary values with the negative value known as the negative class and the postive value as the positive class just that the two values have changed

Definition 41

Given a training example $(x^{(i)}, y^{(i)})$ we define the **functional margin** of (w, b) with respect to the training example

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

again noting that $y \in \{-1, 1\}$ and the **geometric margin** is given by

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{||w||} \right)^T x^{(i)} + \frac{b}{||w||} \right)$$

It is clear that if $||w|| = 1$ the geometric margin equals the functional margin.

The above is for a single training example. If given a training set, the functional and geometric margin is given by

$$\hat{\gamma} = \min_i \hat{\gamma}^{(i)} \quad \text{and} \quad \gamma = \min_i \gamma^{(i)}$$

respectively.

Fact 42

A functional/geometric margin is defined such that:

1. a large margin represents a correct prediction
2. a large margin represents a confident prediction

For (1): Note that the classifier classifies point $x^{(i)}$ in the positive class if $w^T x^{(i)} + b > 0$ and in the negative class otherwise. Now, suppose $y^{(i)} = 1$(positive class). Then for the functional margin to be large we need $w^T x^{(i)} + b$ to be large and positive. On the other hand if $y^{(i)} = -1$ then we need $w^T x^{(i)} + b$ to be large and negative. In this way maximizing the functional margin indeed works towards getting a correct prediction.

For(2) consider that given a training set the geometric margin represents the distance from the decision boundary which indicates a very confident set of predictions it would seem natural that it is good fit to training data is one that maximizes the *geometric margin*.

Remark 43. *We will formalize this concepts which fall under* **margin theory** *in CS229M ML theory. Currently as you can tell this is just an informal description of margins work in classification.*

So we now have a an optimization problem of maximing the geometric margin

Remark 44. *We use geometric margin because clearly unlike the functional margin its value is invariant to scaling of parameters. This will come in handy as we will see later*

Question 45. *How best should we pose our optimization problem?*

(Attempt 1) We first try

$$\max_{\gamma,w,b} \gamma$$

where $\gamma^{(i)} = y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad i = 1, \ldots, m$ and $||w|| = 1$(which ensures we have the geometric margin). However this is not easily solved because the condition $||w|| = 1$ makes the feasible set non convex. This is because $||w|| = 1$ essentially constricts the set to the surface of the unit sphere. But if you take 2 two points on the surface, the line connecting them passes through the interior of the sphere and not through the surface and hence the surface of a sphere is not convex.

(Attempt 2) Instead, recalling that $\gamma = \frac{\hat{\gamma}}{||w||}$ we try to reformulate the problem equivalently as

$$\max_{\hat{\gamma},w,b} \frac{\hat{\gamma}}{||w||}$$

where $\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, i = 1, \ldots, m$. But then now instead of a non convex set we have a non convex function(for the same reasoning as the above case).

(Attempt 3) Finally we could try

$$\min_{w,b} \frac{1}{2} ||w||^2$$

where $\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b) \geq 1 = \hat{\gamma}, i = 1, \ldots, m$. We achieve this by scaling w, b such that $\hat{\gamma} = 1$. In which case we have the task of maximizing $\hat{\gamma}/||w|| = 1/||w||$ which is equivalent to minimizing $||w||^2$ and thus why we format like so.Note that such a form can be solved effectively with commercial quadratic programming code. This is valid because scaling w, b does not change any important properties we are concerned about when we use g as a classifier. That is

$$g(w^T x + b) = g(2w^T x + 2b)$$

refer above for how g was defined then see that scaling will not change the polarity of $w^T x + b$

7.1 Lagrange duality

Consider a problem in the following form

$$\min_{w} f(w)$$

where $h_i(w) = 0, i = 1, \ldots, l$.

Definition 46

The **Langrangian** is defined to be

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

where β_i's are called the **Lagrange multipliers**.

We would then find and set $\mathcal{L}$'s partial derivatives to zero

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0; \frac{\partial \mathcal{L}}{\partial \beta_i} = 0$$

and solve for w and β

We now try to generalize this to constrained optimization problems in which we may have inequity as well as equality constraints.

Definition 47

Consider what we call a **primal** optimization problem

$$\min_w f(w)$$

where

$$g_i(w) \leq 0, i = 1, \ldots, k$$
$$h_i(w) = 0, i = 1 \ldots, l$$

Definition 48

We define the **generalized langrangian** to be

$$\mathcal{L}(w, a, b) = f(w) + \sum_{i=1}^{k} a_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

where a_i's and β_i's ar the lagrange multipliers

Now consider the quantity

$$\theta_{\mathcal{P}}(w) = \max_{a,\beta: a_i \geq 0} \mathcal{L}(w, a, \beta)$$

where here $\mathcal{P}$ stands for "primal". Noting that the above condition essentially says for *all* possible $\alpha, \beta, \alpha_i \geq 0$(the reason for $a_i \geq 0$ is to ensure the langarian is the lower bound of p^* recall `convex analysis stanford notes`) what the max value of $\mathcal{L}$? Next you should be able to verify that

$$\theta_{\mathcal{P}} = \begin{cases} f(w) & \text{if w satisfies primal constraints} \\ \infty & \text{otherwise} \end{cases}$$

This implies

$$\min_w f(w) = \min_w \theta_{\mathcal{P}}(w) = \min_w \max_{a,\beta: a_i \geq 0} \mathcal{L}(w, a, \beta)$$

when primal constraints are met.

Definition 49

We now pose the **dual** optimization problem

$$\max_{a,\beta: a_i \geq 0} \theta_{\mathcal{D}}(a, \beta) = \max_{a,\beta: a_i \geq 0} \min \mathcal{L}(w, a, \beta)$$

Question 50. *How are the primal and dual problems related?*

It can be shown that

$$d^* = \max_{a,\beta: a_i \geq 0} \min_w \mathcal{L}(W, a, \beta) \leq \min_w \max_{a,\beta: a_i \geq 0} \mathcal{L}(w, a, \beta) = p^*$$

refer to your `Rudin` notes 1st few sections where suprenum and infinum and its properties were first introduced.

Remark 51. *In optimization theory this is called weak duality*

Now consider:

Question 52. *Under what conditions will $d^* = p^*$ (or what we call strong duality in optimization theory terms)?*

Definition 53

Consider a mapping between vector spaces $f : V \to W$

An **affine** function is such that

$$f(v) = Av + b$$

for some matrix A and vector b.

Note that in contrast a **linear** function is such that

$$f(v) = Av$$

First suppose that

1. f and the g_i's are convex
2. the h_i's are affine
3. Suppose further that g_i's are strictly feasible meaning that there exists some w so that $g_i(w) < 0$ for all i

If you recall from your `stanford convex analysis notes` we essentially have slater conditions for a convex optimization problem, which means we have strong duality($p^* = d^*$). This why such conditions work.

Remark 54. *Recall the equivalent definitions of convexity of function from `MIT 6.7220 Non Linear Optimization`. In our context we will use the definition for convexity that states if a function has a hessian, its positive semi-definite for practical purposes.*

Now under these assumptions it can be shown that there exists w^*, a^*, β^* such that

- w^* is the solution to the primal problem
- a^*, β^* are the solutions to the dual problem
- $p^* = d^* = \mathcal{L}(w^*, a^*, \beta^*)$
- w^*, a^*, β^* satisfy the **Karush-Kuhn-Tucker**(KKT) conditions(see below)

Also recall from `stanford convex analysis notes` KKT conditions naturally follow under applying optimal constraints that ensure strong duality. In our case:

Fact 55

The **Karush-Kuhn-Tucker**(KKT) conditions are defined by

$$\frac{\partial}{\partial w_i}\mathcal{L}(w^*, a^*, \beta^*) = 0, i = 1, \ldots, n \tag{1}$$

$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, a^*, \beta^*) = 0, i = 1, \ldots, l \tag{2}$$

$$a_i^* g_i(w^*) = 0, i = 1, \ldots, k \tag{3}$$

$$g_i(w^*) \leq 0, i = 1, \ldots, k \tag{4}$$

$$a^* \geq 0, i = 1, \ldots, k \tag{5}$$

We say equation 5 is the **dual complentarity**(or *complementary slackness* - again recall `stanford convex analysis notes`) condition. Specifically it implies that if $a_i^* > 0$ then $g_i(w^*) = 0$. That is to say the $g_i(w) \leq 0$ constraint is **active** meaning it holds with equality rather than inequality.

7.2 optimal margin classifier

recall 45 that we posed the following optimization problem for fiding the optimial margin classifier

$$\min_{\gamma,w,b} = \frac{1}{2}||w||^2$$

where $y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, \ldots, m$. First we can make this such that it satisfies primal constraints by simply writing the constraints as

$$g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$$

Then by *dual complementary* condition above we have that $a_i > 0$ only for training examples that have functional margin($y^{(i)}(w^T x^{(i)} + b)$) equal to one since that will be when $g_i(w) = 0$. We now contruct the langarian for our problem

$$\mathcal{L}(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} a_i[y^{(i)}(w^T x^{(i)} + b) - 1]$$

note that we only have β_i langrange multipliers, no α_i since we only have inequality contraints, no equality ones.We now apply KKT conditions. Recall from `convex analysis stanford` that the dual optimal value is obtained by taking the infinum over x for a fixed set of langarian multipliers. But in our case our x is in terms variables w, b. Recall from `MIT 18.101` the multivariable minimum is when all 1st order partial derivatives of each variable are equal to zero. Therefore we have

$$\nabla_w \mathcal{L}(w, b, a) = w - \sum_{i=1}^{m} a_i y^{(i)} x^{(i)} = 0$$

The 2nd equality implies

$$w = \sum_{i=1}^{m} a_i y^{(i)} x^{(i)}$$

we do the same for b getting

$$\frac{\partial}{\partial b}\mathcal{L}(w, b, a) = \sum_{i=1}^{m} a_i y^{(i)} = 0$$

now substitue our expression for w back into the langarian to get

$$\mathcal{L}(w,b,a) = \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}a_i a_j (x^{(i)})^T x^{(j)} - b\sum_{i=1}^{m} a_i y^{(i)}$$

Remark 56. $\sum_{i,j=1}^{m}$ *sums over* $(i,j) : (1,1) \to (m,m)$ *for a total of* $m \times m$ *iterations*

now using substituting our expression for $\frac{\partial}{\partial b}\mathcal{L}(w,b,a)$ in too we have

$$\mathcal{L}(w,b,a) = \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}a_i a_j (x^{(i)})^T x^{(j)}$$

basically the last term has been removed since equal zero. This is precisely the value of our dual optimal. Now as you would recall from convex analysis, you know the next step is not optimize this over the set of langrage multipliers(aka our dual problem)

$$\max_a W(a) = \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}a_i a_j \langle x^{(i)}, x^{(j)}\rangle$$

where $a_i \geq 0, i = 1,\ldots,m$ and $\sum_{i=1}^{m} a_i y^{(i)} = 0$. You can now easily verify that we indeed have the KKT conditions as stated in 55 for our problem.

7.3 Kernels

That's enough of optimization theory for today.We now return to our discussion on linear regression(recall continuous input variables x) briefly.

Example 57

Suppose we now define

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

Se say our original input variable x is a the **input attributes** of our problem while the "new" input variables x^2, x^3 are the **input features** of our problem. We say $\phi(x)$ is a **feature mapping**, which maps attributes to the features.

Definition 58

We define the corresponding **kernel** to be

$$K(x,z) = \phi(x)^T\phi(z)$$

note that this is essentially $\langle x,z\rangle$ the inner product...to be be continued just a little more stuff that probably will be covered in psets

8 Learning Theory

8.1 Preliminaries

Lemma 59 (Boole's Inequality)

Let $A_1, \ldots, A_n$ denote events(may or may not be independent) then

$$P(A_1 \cup \ldots \cup A_k) \leq P(A_1) + \ldots + P(A_k)$$

Proof. Recall **Boole's Inequality** from `MIT 18.600`

Lemma 60 (Hoeffding inequality)

Let $Z_1, \ldots, Z_m$ be m iid random variables drawn from a bernoulli(which we denote by ϕ here) distribution. That is

$$P(Z_i = 1) = \phi \text{ and } P(Z_i = 0) = 1 - \phi$$

Let $\hat{\phi} = (1/m)\sum_{i=1}^{m} Z_i$ be the mean of these random variables and let $\gamma > 0$ be fixed then

$$P(\left|\phi - \hat{\phi}\right|) \leq 2\exp(-2\gamma^2 m)$$

Proof. See appendix above.

Now to begin our discussion we first restrict our attention to binary classifiers in which the labels are $y \in \{0, 1\}$. Now assume we are given a training set $S = \left\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\right\}$ of size m where the training examples $(x^{(i)}, y^{(i)})$ are drawn idd from some probability distribution $\mathcal{D}$.

Definition 61

For a hypothesis h we define the **training error**(also called the **empirical risk/error** in learning theory) to be

$$\hat{\varepsilon}(h) = \frac{1}{m}\sum_{i=1}^{m} 1\left\{h(x^{(i)}) \neq y^{(i)}\right\}$$

where $1\{\ldots\}$ is an indicator function that is 1 when the condition in its arguments is specified and zero otherwise.

Therefore this function simply represents the fraction of training examples that h misclassifies.

Definition 62

We also define the **generalization error** to be

$$\varepsilon(h) = P_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$$

which represents the probability that if we now draw a new example (x, y) from the distribution $\mathcal{D}$, h will misclassify it.

Like we have in the past we now ask:

Example 63

Suppose we have the setting where $h_\theta(x) = 1\left\{\theta^T x \geq 0\right\}$. What's a reasonable way of fitting the paramters θ?

To that end we might try to minimize the training error and pick

$$\hat{\theta} = \arg\min_\theta \hat{\varepsilon}(h_\theta)$$

we call this process **empirical risk minimization**(ERM)

So we can now think of ERM as

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\varepsilon}(h)$$

Definition 64

We define the **hypothesis class** $\mathcal{H}$ used by the learning algorithm to be the set of all classifiers considered by it.

8.2 the case of finite hypothesis class

Consider a finite hypothesis class $\mathcal{H} = \{h_1, \ldots, h_k\}$ consisting of k hypotheses. Therefore in our context of binary classification $\mathcal{H}$ is just a set of k functions mapping from $\mathcal{X}$ to $\{0, 1\}$. Next sample an arbiturary training example (x, y) from $\mathcal{D}$.Then we define bernoulli random variable Z whose distribution is defined by $Z = 1\{h_i(x) \neq y\}$ so our training error can be written as

$$\hat{\varepsilon}(h_i) = \frac{1}{m}\sum_{j=1}^{m} Z_j$$

where $Z_j = 1\left\{h_i(x^{(j)}) \neq y^{(j)}\right\}$. Now we might think

Question 65. *Can we give guarantees on the generalization error of $\hat{h}$? That is $\varepsilon(\hat{h}) < \{Bound?\}$ If so how?*

As a rough overview we will do so via

1. showing that $\hat{\varepsilon}(h)$ is a reliable estimate of $\varepsilon(h)$ for all h
2. show that this implies an upper bound on the generalization error of $\hat{h}$

First observe that $\mathbb{E}[Z_j] = \hat{\varepsilon}(h_i)$ so we can apply the **hoeffding inequality**(60) and obtain

$$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

where $\varepsilon(h_i) = P(Z_i = 1)$. Therefore we we have shown for a particular h_i the training error will be close to the generalization error with high probability assuming m is large.This makes sesnse intuitively actually. The greater the training set size,the more accurate the esimate of the generalization error the training error is

Let's do better by generalizing this to all $h \in \mathcal{H}$ not just one particular h_i. To this end let A_i denote the event that $|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma$. That is we restate the above results, saying that have just shown that $P(A_i) \leq 2\exp(-2\gamma^2 m)$.

Now by **boole's inequality**(or what Andrew calls the "union bound") we have

$$\begin{aligned} P(\exists h \in \mathcal{H} | \varepsilon(h_i) - \hat{\varepsilon}(h_i) > \gamma) &= P(A_1 \cup \ldots \cup A_k) \\ &\leq \sum_{i=1}^{k} P(A_i) \\ &\leq \sum_{i=1}^{k} 2\exp(-2\gamma^2 m) \\ &= 2k\exp(-2\gamma^2 m) \end{aligned}$$

then subtracting 1 from both sides we find that

$$P(\neg\exists h \in \mathcal{H} | \varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) = P(\forall h \in \mathcal{H} | \varepsilon(h_i) - \hat{\varepsilon}(h_i)| \leq \gamma) \geq 1 - 2k\exp(-2\gamma^2 m)$$

where the symbol $\neg$ means not.Nice! With the application of relatively simple inequalties we have arrived at a **uniform convergence** result(recall what it means in mathematical analysis...it makes sense in this context) since this bounds holds simultaneously for all $h \in \mathcal{H}$(as seen in the 2nd term)

Note there are 3 quantities of interest here m, γ and the probability error. What we did above we as to for a particular m, γ get a bound on the on probability difference between training and generalization error $|\varepsilon(h) - \hat{\varepsilon}(h)|$. In general easily do the same by fixing 2 variables to bound another variable.

Example 66

Let's suppose we fix the probability difference and γ. Now find a bound on m such probability at least $1 - \delta$ we have $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma$. First we set $\delta = 2k\exp(2\gamma^2 m)$. Then from the above we know

$$P(\forall h \in \mathcal{H} | \varepsilon(h_i) - \hat{\varepsilon}(h_i)| \leq \gamma) \geq 1 - 2k\exp(-2\gamma^2 m)$$

therefore our problem is to find m such that

$$1 - 2k\exp(-2\gamma^2 m) \geq 1 - \delta$$

then one can easily verify that

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

The training set size m that an algorithm requires to achieve a certain level of performance is called the algorithms **sample complexity**

Remark 67. *In this book apparently Andrew treats* ln *as a* log*. Please bear with this notation. I think its probably of big o notation that we will employ later. In which case* ln *is just treated as "logarithmic"*

Example 68

Let's suppose we fix the probability difference and m. Now find a bound on γ such that probability at least $1-\delta$ we have $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma$. First we set $\delta = 2k\exp(2\gamma^2 m)$. Then from the above we know

$$P(\forall h \in \mathcal{H} | \varepsilon(h_i) - \hat{\varepsilon}(h_i)| \leq \gamma) \geq 1 - 2k\exp(-2\gamma^2 m)$$

therefore our problem is to find m such that

$$1 - 2k\exp(-2\gamma^2 m) \geq 1 - \delta$$

then one can easily verify that

$$\gamma \geq \sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

Theorem 69

Let $|\mathcal{H}| = k$ and let any m, δ be fixed. Then with probability at least $1-\delta$ we have that

$$\varepsilon(\hat{h}) \leq \left(\min_{h\in\mathcal{H}} \varepsilon(h)\right) + 2\sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

assumining uniform convergence i.e $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma, \forall h \in \mathcal{H}$(recall earlier)

Proof. First define $h^* = \arg\min_{h\in\mathcal{H}} \varepsilon(h)$ and $\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{\varepsilon}(h)$(recall ERM). Then

$$\begin{aligned}
\varepsilon(\hat{h}) &\leq \hat{\varepsilon}(\hat{h}) + \gamma \\
&\leq \hat{\varepsilon}(h^*) + \gamma \\
&\leq \varepsilon(h^*) + 2\gamma
\end{aligned}$$

where the first line follows by uniform convergence assumption(compares ε with $\hat{\varepsilon}$ for the same arugment $\hat{h}$. The second line follows because by definition $\hat{h}$ is the minimum of $\hat{\varepsilon}$. Then the final line follows again by uniform convergence assumption. Now because uniform converence holds when the probability is at least $1 - 2k\exp(-2\gamma^2 m)$ and that we want to probability to be at least $1-\delta$ for a fixed m and probability by 68 we mus have that

$$\gamma \geq \sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

which implies

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*) + 2\gamma \quad \Rightarrow \quad \varepsilon(\hat{h}) \leq \min_{h\in\mathcal{H}} \varepsilon(h) + 2\sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

8.3 the case of infinite hypothesis class

Definition 70

Given a set $S = \{x^{(i)}, \ldots, x^{(d)}\}$ of points $x^{(i)} \in \mathcal{X}$ we say that $\mathcal{H}$ **shatters** S if $\mathcal{H}$ can realize any labelling on S. That is for any set of labels $\{y^{(1)}, \ldots, y^{(d)}\}$ there exists some $h \in \mathcal{H}$ so that $h(x^{(i)}) = y^{(i)}$ for all $i = 1, \ldots, d$.

Definition 71

Given a hypothesis class $\mathcal{H}$ we define its **Vapnik-Chervonenkis dimension** written $VC(\mathcal{H})$ to be the size of the largest set that is shattered by $\mathcal{H}$

Example 72 (H)

Consider

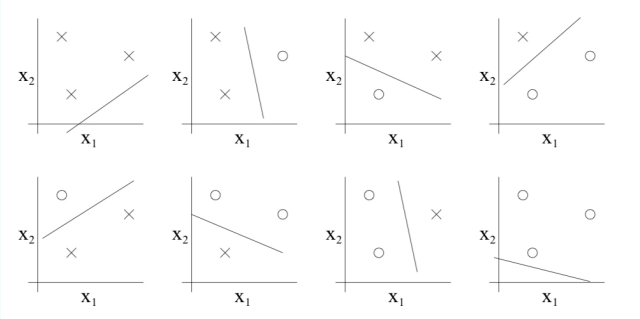


Figure 2: $VC(\mathcal{H}) = 3$

9 regularization and model selection

Suppose we are trying to select several different models for a learning problem. For instance we might be using a polynomial regression model $h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_k x^k$ and wish to select if k should be $0, 1 \ldots$ or 10.

Question 73. *How can we automatically select a model that represents a good tradeoff between the twin eveils of bias and variance?*

9.1 Cross Validation

...to be continued after psets

9.2 The K-means clustering algorithm

Definition 74

In the **clustering** problem we are given a training set $\{x^{(i)}, \ldots, x^{(m)}\}$ and want to group the data into a few cohesive "clusters". Here $x^{(i)} \in \mathbb{R}^n$ as usual but no labels $y^{(i)}$ are given so this is an unsupervised learning problem

The k means clustering algorithm is as follows

```
1    1. Initialize cluster centroid μ_1, μ_2, ..., μ_k ∈ R^n randomly
2    2. Repeat until convergence:
3        For every i set
4            c^(i) = arg min_j ||x^(i) − μ_j||^2
5        For each j set
```

6 $$\mu_j = \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$$

In the algorithm above k(a parameter of the algorithm) is the number of clusters we want to find. The cluster centroids μ_j represent our current guesses for the positions for the center of the clusters. The inner loop essentially assigns training example $x^{(i)}$ to the cloest centroid μ_j and then the new mean is calculated for the new shifted centroid cluster.

Definition 75

The **distortion function** is defined to be

$$J(c,\mu) = \sum_{i=1}^{m} \left|\left|x^{(i)} - \mu_{c^{(i)}}\right|\right|^2$$

it can be proven that the k means is gauranteed to converge as each inner loop repeated minimizes J with respect to c while holding μ fixed so it must monotonically decrease.

9.3 Mixtures of Gaussians and the EM algorithm

We now discuss what is known as the EM(expectation-maximization) for density estimation.

Suppose that we are given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$ as usual. As we are working in the unsupervised learning setting these points do not come with labels.

We wish to model the data by specifying a joint distribution

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$$

Here

- $z^{(i)} \sim$ Multinomial(ϕ) where $\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$
- the paramters ϕ_j gives $p(z^{(i)} = j)$
- $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$

we let k denote the number of values that the $z^{(i)}$s can take on. This is to say that each $x^{(i)}$ was generated

1. First randomly choosing $z^{(i)}$ from $\{1, \ldots, k\}$
2. Then drawing $x^{(i)}$ from one of the k gaussians i.e the $x^{(i)}|z^{(i)}$ depending on $z^{(i)}$

So our parameters for the model are ϕ, μ and $\sum$. First we write down the likelihood of our data

$$\ell(\phi,\mu,\Sigma) = \sum_{i=1}^{m} \log p(x^{(i)};\phi,\mu,\Sigma) = \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)};\mu,\Sigma)p(z^{(i)};\phi)$$