# Rec 7 Notes

Overview:

1. Probability, Odds, Logit — 3 units of uncertainty
2. Logistic Regression
   3. Fitting a Logistic Regression Model
   4. Evaluating a Logistic Regression Model → skipped for time
5. Creating a Binary Classifier from a Logistic Regression Model
6. Carrying out a Logistic Regression in Practice

## ① Probability, Odds, Logit — 3 units of uncertainty

Suppose we have a random variable $Y$ that can take value 0 or 1, but its value is uncertain. We define the probability that $Y=1$ using a hypothetical large number of identical r.v.s $Y^1, \dots, Y^N$ all independent from one another. Then we say that probability that $Y=1$, denoted $\mathbb{P}[Y=1]$ as the portion of the $N$ that would take value 1 as $N$ gets larger and larger:

$$p = \mathbb{P}[Y=1] = \frac{\sum_{n=1}^{N} \mathbb{1}[Y=1]}{N} \qquad * \ \mathbb{1}[Y=1] = \begin{cases} 1 & \text{if } Y=1 \\ 0 & \text{if } Y=0 \end{cases}$$

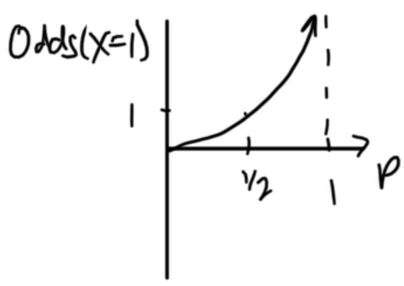
From this defn we can see probabilities take values in $[0,1]$.

Probability is just one way of measuring uncertainty.
In logistic regression, we use another representation or scale of uncertainty called the "log-odds" or "logit", which is a transformation of a third representation called "odds".

$$\text{odds}(Y=1) = \frac{\mathbb{P}[Y=1]}{\mathbb{P}[Y=0]} = \frac{p}{1-p} \qquad [\text{prob} \to \text{odds}]$$

* value in $[0,\infty]$

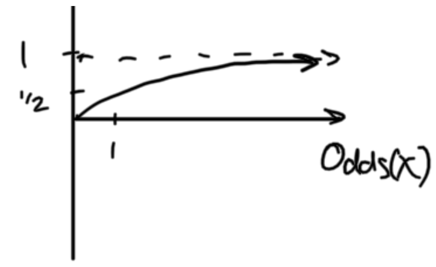
| $p$ | odds |
|---|---|
| .01 | .01 |
| .1 | .11 |
| .25 | .33 |
| .5 | 1 |
| .75 | 3 |
| .9 | 9 |
| .99 | 99 |
| $[0,1]$ | $[0,\infty]$ |




or equivalently, $[\text{odds} \to \text{prob}]$

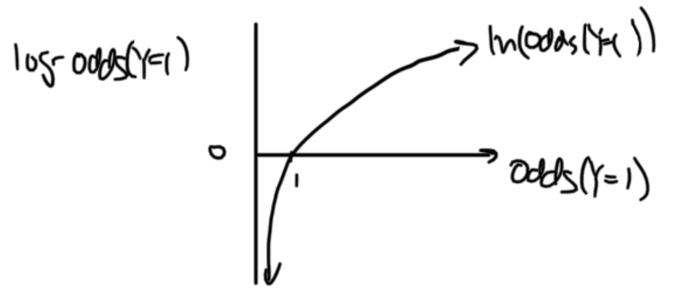
$$p = \frac{\text{Odds}(X=1)}{}$$

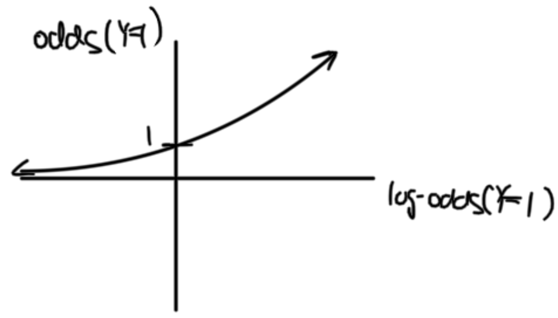
1+ Odds(X=1)




The "log-odds" are defined as you'd expect:

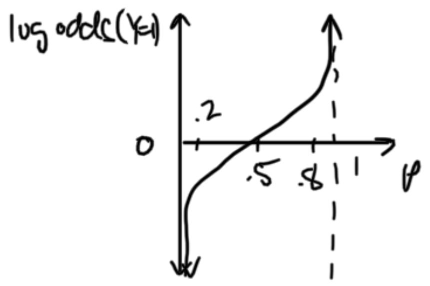
$$\text{log-odds}(Y=1) = \ln(\text{odds}(Y=1)) \quad \underline{\text{and}} \quad \text{odds}(Y=1) = e^{\text{log-odds}(Y=1)}$$

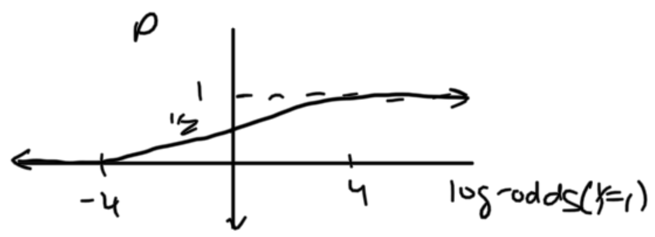



If you look at the relationship between logit(Y=1) and $\mathbb{P}[Y=1]$ you get equations:

$$\text{logit}(Y=1) = \ln(\text{odds}(Y=1)) = \ln\left(\frac{p}{1-p}\right) \quad \underline{\text{and}} \quad \mathbb{P}[Y=1] = \frac{e^{\text{logit}(Y=1)}}{1+e^{\text{logit}(Y=1)}} = \frac{1}{1+e^{-\text{logit}(Y=1)}}$$




| prob | odds | log-odds |
|---|---|---|
| .01 | .01 | -4.6 |
| .1 | .11 | -2.2 |
| .25 | .33 | -1.09 |
| .5 | 1 | 0 |
| .75 | 3 | 1.09 |
| .9 | 9 | 2.2 |
| .99 | 99 | 4.6 |
| [0,1] | [0,∞) | (-∞,∞) |

The convenient aspect of the logit(X=1) is that it takes values over $[-\infty, \infty]$ not just $[0,1]$ or $[0,\infty]$.

## ② Logistic Regression

In logistic regression, we have some predictors of the value of Y which we call $X_1, \dots, X_p$ and we suppose that

$$\text{logit}(Y=1 \mid X=x) = c_0 + c_1 x_1 + \dots + c_p x_p \quad (\text{letting } X = [X_1, \dots, X_p]')$$

and our goal is to find the "best" values for $c_1, \dots, c_p$.

In probabilities this assumption looks like:

$$\mathbb{P}[Y=1 \mid X=x] = \frac{1}{1+e^{-\text{logit}(Y=1 \mid X=x)}} = \frac{1}{1+e^{-(c_0+c_1x_1+\dots+c_px_p)}}$$

We define "best" values for $c_1,\dots,c_p$ based on what values make an observed data set <u>most likely</u>. So suppose we have data

$$(x^1, y^1), \dots, (x^n, y^n)$$

Our logistic model estimates the probability of $Y=1$ given $X=x$:

$$\hat{p}(x) = \hat{\mathbb{P}}[Y=1 \mid X=x]$$

which we can use to compute the probability of observing the data we observed:

$$\hat{\mathbb{P}}[\text{Observed Dataset}] = \prod_{i=1}^{n} \hat{\mathbb{P}}[Y=y^i \mid X=x^i]$$

$$= \left(\prod_{i:\,y^i=1} \underbrace{\frac{1}{1+e^{-(c_0+c_1x_1^i+\dots+c_px_p^i)}}}_{p_i(c)}\right)\left(\prod_{i:\,y^i=0} 1-\overbrace{\frac{1}{1+e^{-(c_0+c_1x_1^i+\dots+c_px_p^i)}}}^{p_i(c)}\right)$$

$$= f(c_0, c_1, \dots, c_p) \quad \leftarrow \text{notice this is a func of } c_1,\dots,c_p$$

We want to make the observed data as likely as possible, so maximize:

$$\max \hat{\mathbb{P}}[\text{observed dataset}] = \max_{c_0,c_1,\dots,c_p} f(c_0, c_1, \dots, c_p)$$

If you compose a function w/ an increasing function, the maximizer doesn't change. So I can alternatively maximize something that doesn't involve so many products:

$$\max \ln\left(\hat{\mathbb{P}}[\text{observed dataset}]\right)$$

$$= \max_{c} \sum_{i:\,y^i=1} \ln(p_i(c)) + \sum_{i:\,y^i=0} \ln(1-p_i(c))$$

$$= \max_{c} \sum_{i=1}^{n} y^i \ln(p_i(c)) + (1-y^i)\ln(1-p_i(c)) = \max_{c} \ell(c)$$

** note: If curious, a formal proof of the fact that composing a function w/ a monotonic func doesn't change local minima/maxima of the original func. Let $g(x)$ be a monotonic func and $f(x)$ be the original function. Then defining $h(x) := (g \circ f)(x) = g(f(x))$, we are saying $x^*$ local extrema of $f(x)$ iff $x^*$ is local extrema of $h(x)$.
Consider $x^*$ a local extrema $f(x)$. WLOG let $x^*$ be a local maximizer, so $f(x) \leq f(x^*)$ in some nbhd of $x^*$. By saying $g(x)$ is monotonic

we mean if $x_1 \le x_2$ then $g(x_1) \le g(x_2)$. Using this, $x^*$ is a local maximizer of $h(x)$ since

$$h(x) = g(f(x))$$
$$\le g(f(x^*)) \quad \text{*since } f(x) \le f(x^*) \ \forall x \text{ in nbhd of } x^* \text{ and } g \text{ is monotonic}$$
$$= h(x^*)$$

Next consider $x^*$ a local maximizer of $h(x)$. Then $h(x) \le h(x^*)$ $\forall x$ in a nbhd of $x^*$. By def$^n$ of $h$ we have

$$g(f(x)) \le g(f(x^*))$$

If $f(x) > f(x^*)$ then, by monotonicity of $g$, the above inequality must not hold. So it must be that $f(x) \le f(x^*)$ $\forall x$ in nbhd of $x^*$.

③ Fitting a Logistic Regression Model

As a sum of concave fncs, this is a concave function so there is a guaranteed unique maximizer.

** note: Showing the sum of concave fncs is concave...

Let $f_1(x)$, $f_2(x)$ be 2 convex functions. One def$^n$ of this is that for any $x_1, x_2$, $f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall \lambda \in [0,1]$

If I create a new function $g(x) = f_1(x) + f_2(x)$, it is also convex.

WTS for any $x_1, x_2$

$$g(\lambda x_1 + (1-\lambda)x_2) \le \lambda g(x_1) + (1-\lambda) g(x_2) \quad \forall \lambda \in [0,1]$$

$$\begin{aligned}
g(\lambda x_1 + (1-\lambda)x_2) &= f_1(\lambda x_1 + (1-\lambda)x_2) + f_2(\lambda x_1 + (1-\lambda)x_2) && \forall \lambda \in [0,1] \\
&\le \lambda f_1(x_1) + (1-\lambda) f_1(x_2) + \lambda f_2(x_1) + (1-\lambda) f_2(x_2) && \forall \lambda \in [0,1] \\
&= \lambda(f_1(x_1) + f_2(x_1)) + (1-\lambda)(f_1(x_2) + f_2(x_2)) && \forall \lambda \in [0,1] \\
&= \lambda g(x_1) + (1-\lambda) g(x_2) && \forall \lambda \in [0,1]
\end{aligned}$$

If $f_1(x)$ and $f_2(x)$ are concave, then $-f_1(x)$ and $-f_2(x)$ are convex. By above, $(-f_1(x)) + (-f_2(x))$ is convex. And finally,

$$-\left((-f_1(x)) + (-f_2(x))\right) = f_1(x) + f_2(x) \text{ is concave.}$$

By induction this holds for the sum of any finite number of concave functions. In particular, $f_1(x) + f_2(x) + \dots + f_n(x)$ is just a series of the sum of 2 concave functions

$$((f_1(x) + f_2(x)) + f_3(x)) + \dots + f_n(x).$$

Finally, in this case we just need to show $\ln\left(\frac{1}{1+e^{-x}}\right)$ is concave. We can do this w/ the 2$^{nd}$ derivative test.

$$\frac{d}{dx}\left(\ln\left(\frac{1}{1+e^{-x}}\right)\right) = \frac{1}{\frac{1}{1+e^{-x}}} \cdot \frac{1}{(1+e^{-x})^2} e^{-x} = \frac{\cancel{(1+e^{-x})}}{(1+e^{-x})^{2}} e^{-x}$$

$$\frac{e}{1+e^{-x}} = \frac{1}{1+e^{x}} = (1+e^{\cdot})$$

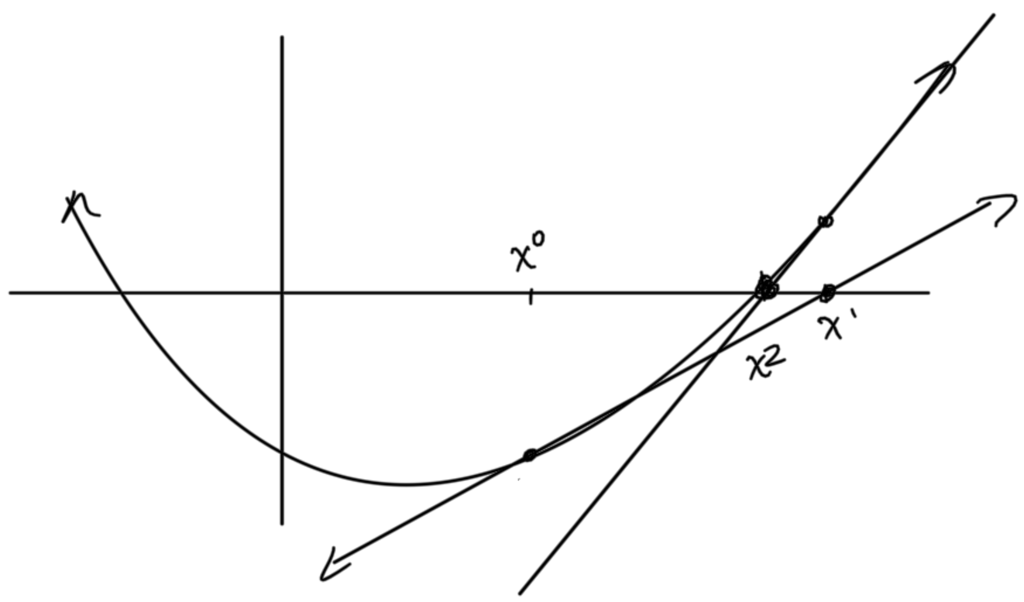
$$\frac{d^2}{dx^2}\left(\ln\left(\frac{1}{1+e^{x}}\right)\right) = \frac{1}{(1+e^{x})^2}(-1)e^{x} = -\frac{e^{x}}{(1+e^{x})^2}$$

$$\leq 0 \quad \forall x \quad \Rightarrow \quad \underline{\text{Concave}} \ \forall x$$

We find the optimum using numerical methods like Newton's Method to find zeros of the derivative.

Newton's Method: Given a function $f(x)$ we compute an approximate zero of the function by

1. Choose some initial point $x^* = x^0$
2. Compute the best linear approximation of $f(x)$ at $x^*$:
$$\tilde{f}(x) = f(x^*) + f'(x^*)(x - x^*)$$
3. Compute the zero of $\tilde{f}(x)$:
$$x^+ = \frac{-f(x^*)}{f'(x^*)} + x^*$$
4. Repeat step 2 and 3 w/ $x^* = x^+$ until $|f(x^0)| < \varepsilon$.




<u>Computing the Derivative</u>

$$\frac{\partial p_i(c)}{\partial c_k} = \frac{\partial}{\partial c_k}\left(1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)^{-1} = -\left(1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)^{-2}\left(e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)(-x_k^i)$$

$$\frac{\partial \ell}{\partial c_k} = \sum_{i=1}^n y^i \frac{1}{p_i(c)} \cdot \frac{\partial p_i(c)}{\partial c_k} + (1-y^i)\frac{1}{1-p_i(c)} \cdot \frac{\partial}{\partial c_k}(1-p_i(c))$$

$$= \sum_{i=1}^n y^i \frac{\cancel{1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}}}{1} \cdot -\left(1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)^{\cancel{-2}}\left(e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)(-x_k^i)$$

$$+ (1-y^i)\frac{\cancel{1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}}}{\cancel{e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}}}\left(1+e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)^{\cancel{-2}}\cancel{\left(e^{-(c_0+c_1x_1^i+\ldots+c_px_p^i)}\right)}(-x_k^i)$$

$$= \sum_{i=1}^n y^i(1-p_i(c))x_k^i \qquad -(1-y^i)p_i(c)x_k^i$$

$$= \sum_{i=1}^{n} \quad y^i x_k^i - \cancel{y^i p_i(c) x_k^i} - p_i(c) x_k^i + \cancel{y^i p_i(c) x_k^i}$$

$$= \sum_{i=1}^{n} (y^i - p_i(c)) x_k^i \qquad \forall k = 1, \dots, p$$

$$\frac{\partial \ell}{\partial c_0} = \sum_{i=1}^{n} y^i - p_i(c)$$

$$\frac{\partial^2 \ell}{\partial c_k c_j} = \frac{\partial \ell}{\partial c_j}\left(\frac{\partial \ell}{\partial c_k}\right) = \frac{\partial \ell}{\partial c_j} \sum_{i=1}^{n} (y^i - p_i(c)) x_k^i$$

$$= \sum_{i=1}^{n} \frac{\partial \ell}{\partial c_j} (y^i - p_i(c)) x_k^i$$

$$= \sum_{i=1}^{n} \frac{\partial \ell}{\partial c_j} - p_i(c) x_k^i$$

$$= \sum_{i=1}^{n} - (1 - p_i(c))(x_j^i)(x_k^i)$$

$$c^* = c^0$$

$$\downarrow$$

$$\tilde{\ell}'(c) = \ell'(c^*) - \ell''(c^*)(c^* - c)$$

$$\downarrow$$

if $|\ell'(c^+)| > \varepsilon$

$$c^+ = \frac{-\ell'(c^*)}{\ell''(c^*)} + c^*$$

$$\downarrow$$

return $c^+$

this works for one coefficient, for more you do:

$$c^+ = c^* - H^{-1}(c^*) \nabla \ell(c^*)$$

ex) $(x^1, y^1) = (2, 1)$
$(x^2, y^2) = (-1, 1)$
$(x^3, y^3) = (-2, 0)$
$(x^4, y^4) = (1, 0)$

single-variable model: $\mathbb{P}(Y=1 \mid X=x) = \frac{1}{1+e^{cx}}$

$$\ell(c) = \sum_{i=1}^{n} y^i \ln(p_i(c)) + (1-y^i)\ln(1-p_i(c))$$

$$= \ln \frac{1}{1+e^{-2c}} + \ln \frac{1}{1+e^{c}} + \ln 1 - \frac{1}{1+e^{2c}} + \ln 1 - \frac{1}{1+e^{-c}}$$

$$\ell'(c) = \sum_{i=1}^{n} (y^i - p_i(c)) x_k^i$$

$$= \left(1 - \frac{1}{1+e^{-2c}}\right)(2) + \left(1 - \frac{1}{1+e^{c}}\right)(-1) + \left(-\frac{1}{1+e^{2c}}\right)(-2) + \left(-\frac{1}{1+e^{-c}}\right)(1)$$

$$\ell''(c) = \sum_{i=1}^{n} -(1-p_i(c))(x^i)^2$$

$$= -\frac{e^{-2c}}{(1+e^{-2c})^2} 2^2 - \frac{e^{+c}}{(1+e^{+c})^2}(-1)^2 - \frac{e^{+2c}}{(1+e^{+2c})^2}(-2)^2 - \frac{e^{-c}}{(1+e^{-c})^2}(1)$$

Suppose we start w/ $c^* = 0$. Then Newton's Method would give

1) $c^* = 0 - \frac{\ell'(0)}{\ell''(0)} = 0.4$

2) $c^* = 0.4 - \frac{\ell'(0.4)}{\ell''(0.4)} = 0.419$

$\vdots$

This is confirmed in the desmos plot.

④ Evaluating a Logistic Regression Model:

a) How good is my model?

b) How meaningful are each individual parameter of the model?

⑤ Creating a Binary Classifier using a Logistic Regression

a) A binary classifier is a function $f: \mathbb{R}^p \to \{0,1\}$ whose input is the values of our predictors and output is our prediction.

ex) $f(x_1, \dots, x_p) = 1$ *works great if you're predicting something that always happens.

ex) $f(x_1, \dots, x_p) = \begin{cases} 1, & \text{if } x_1 \geq 0 \\ 0, & \text{o.w.} \end{cases}$ * works great if $Y=1$ when $x_1$ is non-negative

ex) $f(x_1, \dots, x_p) = \begin{cases} 1, & \tilde{\mathbb{P}}(Y=1 \mid X_1 = x_1, \dots, X_p = x_p) = \dfrac{1}{1+e^{-(c_0 + c_1 X_1 + \dots + c_p X_p)}} \geq T \\ 0, & \text{o.w.} \end{cases}$

This is a binary classifier made using logistic regression.

<u>Geometry of a Logistic Regression based classifier:</u>

Take a look at our logistic regression fnc:

$$\tilde{\mathbb{P}}(Y=1 \mid X_1=x_1, \dots, X_p=x_p) = \frac{1}{1+e^{-(c_0+c_1x_1+\dots+c_px_p)}} \geq T$$

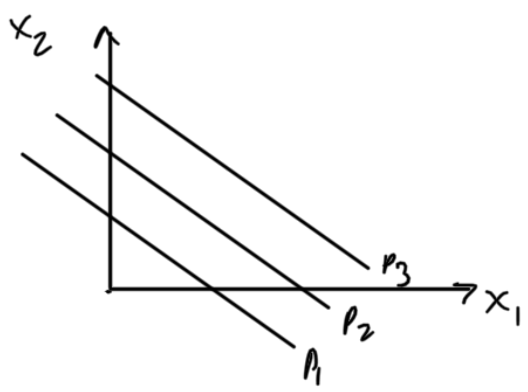
What does the set of predictors $(x_1, \dots, x_p)$ that all give the same probability look like? Aka what is a contour line of this function? Suppose the probability is $p$, so

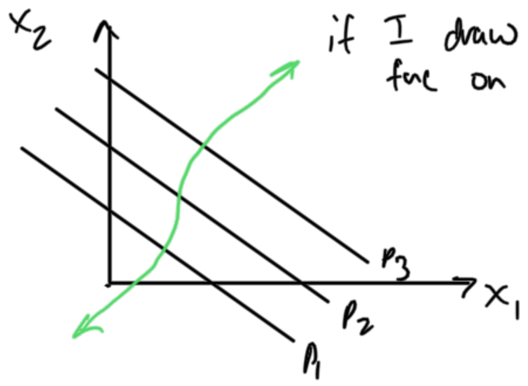
$$p = \frac{1}{1+e^{-(c_0+c_1x_1+\dots+c_px_p)}}$$

The fnc $x \to \frac{1}{1+e^{x}}$ is 1-to-1 so all that matters is that

$$c_0 + c_1x_1 + \dots + c_px_p = \text{constant}$$

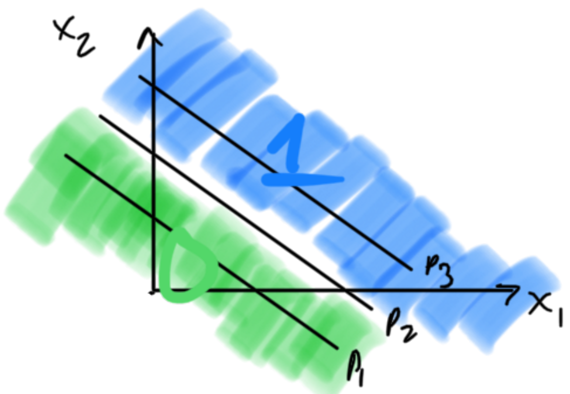
This means a contour line of the linear fnc is a contour line of our logistic regression fnc. These are hyperplanes! So we can draw contour lines for our logistic regression like this:




if I draw the logistic fnc on top.

After applying our threshold we choose a contour line that divides the space into 2 regions. For instance, in the drawing above if $p_1 \leq p_2 \leq p_3$ and I choose $T = p_2$ then my classifier looks like:




When the "decision boundary" for a classifier is one line, it is called a "<u>linear classifier</u>".

Notice that any logistic regression based classifier is equivalent to a classifier of the form

$$f(x_1, \dots, x_p) = \begin{cases} 1, & a'x \geq b \\ 0, & \text{o.w.} \end{cases}$$

Given a logistic regression how do you a and b?

b) Often you have a binary classifier that involves a threshold parameter. To choose the value of the threshold we can use a ROC curve to balance the tradeoff between

"True Positive Rate" - proportion of 1's that are labeled 1's

and

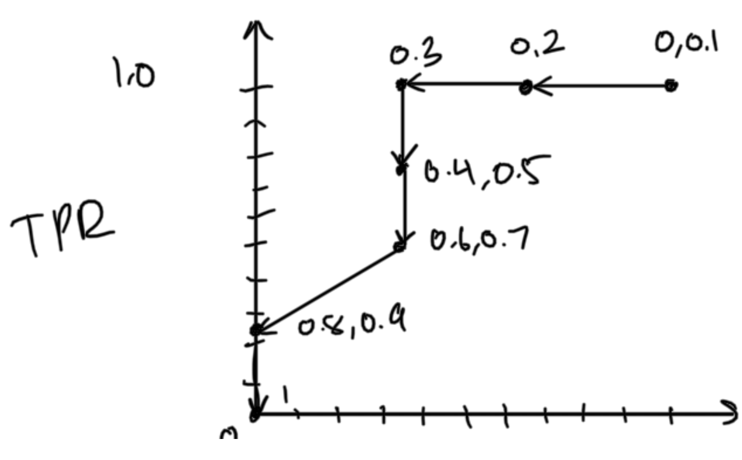
"False Positive Rate" - proportion of 0's that are errantly labeled a 1

Lets work out an example. Suppose you use logistic regression to estimate the probability that an outcome is 1 for 7 different samples and get the following:

| $P[y^i = 1 \mid x^i]$ | $y^i$ |
|---|---|
| 0.1 | 0 |
| 0.2 | 0 |
| 0.3 | 1 |
| 0.5 | 1 |
| 0.7 | 1 |
| 0.75 | 0 |
| 0.95 | 1 |

As you change the labeling threshold you have a trade off between TPR and FPR:

| Threshold | False Positive Rate | True Positive Rate |
|---|---|---|
| 0 | $3/3 = 1$ | $4/4 = 1$ |
| 0.1 | $3/3 = 1$ | $4/4 = 1$ |
| 0.2 | $2/3 \approx 0.66$ | $4/4 = 1$ |
| 0.3 | $1/3 \approx 0.33$ | $4/4 = 1$ |
| 0.4 | $1/3 \approx 0.33$ | $3/4 = .75$ |
| 0.5 | $1/3 \approx 0.33$ | $3/4 = .75$ |
| 0.6 | $1/3 \approx 0.33$ | $2/4 = .5$ |
| 0.7 | $1/3 \approx 0.33$ | $2/4 = .5$ |
| 0.8 | $0/3 = 0$ | $1/4 = .25$ |
| 0.9 | $0/3 = 0$ | $1/4 = .25$ |
| 1 | $0/3 = 0$ | $0/4 = 0$ |



0 FPR 1.0

To evaluate how good the thing underlying your classifier is at predicting the outcome we can use the AUC.

In this case, the AUC is:

$$1 - (1/3)(1/2) - 1/2(1/3)(1/4) = 0.792$$

Which threshold would you choose?

## 6 Carrying out a Logistic Regression in Practice

Attached python notebook shows how to fit a logistic model using Sklearn. This package doesn't include all the detailed statistics on model fit and coefficient significance that statistical packages do.