MLE and a little optimization

Rob McCulloch

1. Introduction

When we did Naive Bayes we had to estimate

$$p(X_i = x_i \mid Y = y) \text{ (or } p(x_i \mid y) \text{).}$$

How did we do it?

We simply used *the observed frequency*:

To estimate $p(X_i = x_i \mid Y = y)$:
in the training data, out of the times $Y = y$,
what fraction of observations have $X_i = x_i$.

If $X_i \sim \text{Bern}(p)$, we estimate p with the observed fraction of times $x_i = 1$.

We call p the *parameter* of the *statistical model* $X \sim \text{Bern}(p)$.

We consider a variety of statistical models and need to estimated the associated parameters.

For example, if we assume $Y_i \sim N(\mu, \sigma^2)$ then we have to estimate (μ, σ^2).

While the observed conditional frequency seems very reasonable for esitmating probabilities, we want a general approach to estimating the parameters of a statisical model.

Maximimum likelihood is a very general appproach which we will review.

Along the way, we will also review some very basic ideas from optimization.

2. Finding a Minimum, one variable

Let f be a function of a single variable, so that $f(x)$ is an number for $x \in C \subset R$.

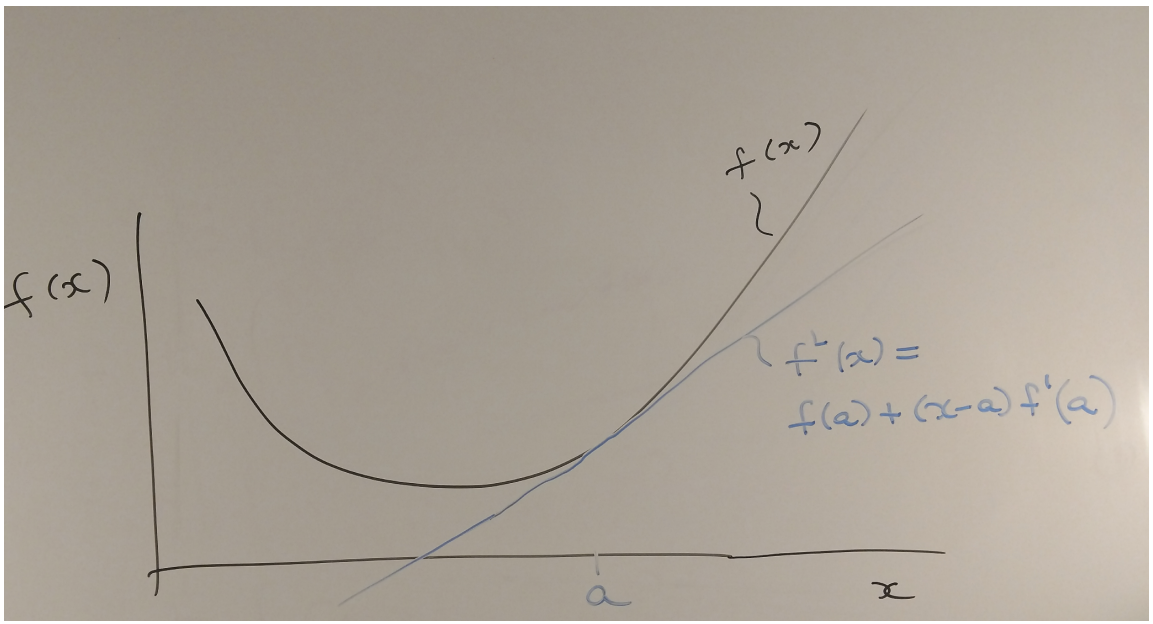
x_0 is a local minimum if $f(x) \geq f(x_0)$ for all x close to x_0.

x_0 is a global minimum if $f(x) \geq f(x_0)$ for all $x \in C$.

Recall:

The derivative gives you a linear approximation to the function:

$$f(x) - f(a) \approx (x - a)f'(a).$$



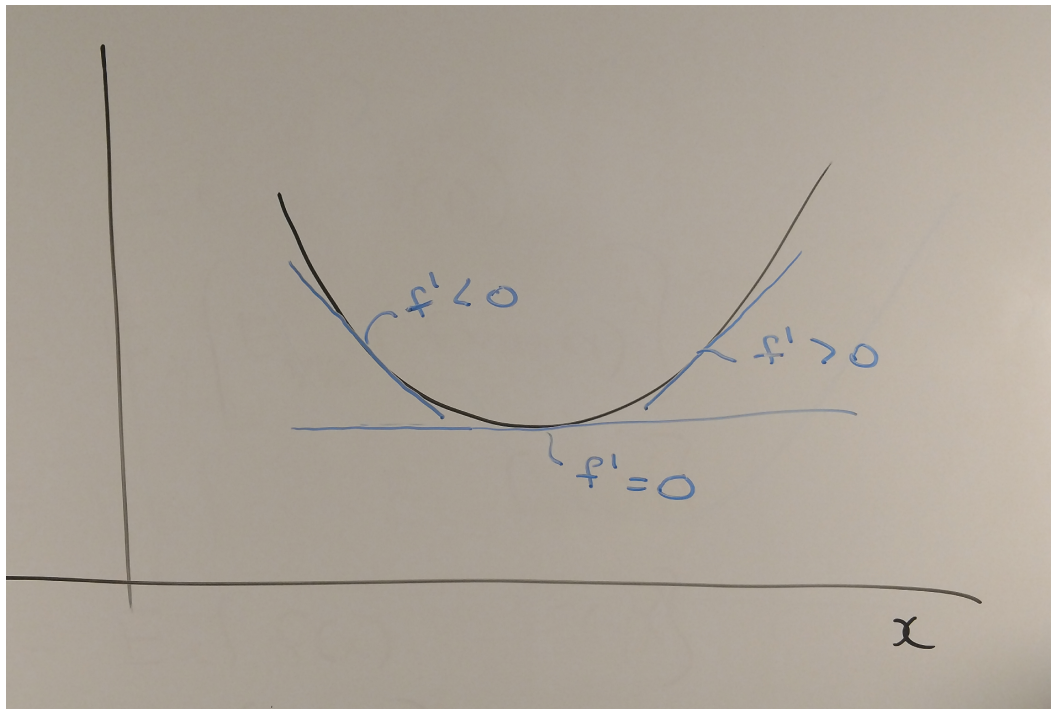
For x close to a, $f(x) \approx f^L(x)$.

Neccessary Condition:

If x_0 is a local min (or max) then $f'(x_0) = 0$.

Sufficient Condition:

If $f'(x_0) = 0$ and $f''(x_0) >$, then x_0 is a local minimum.

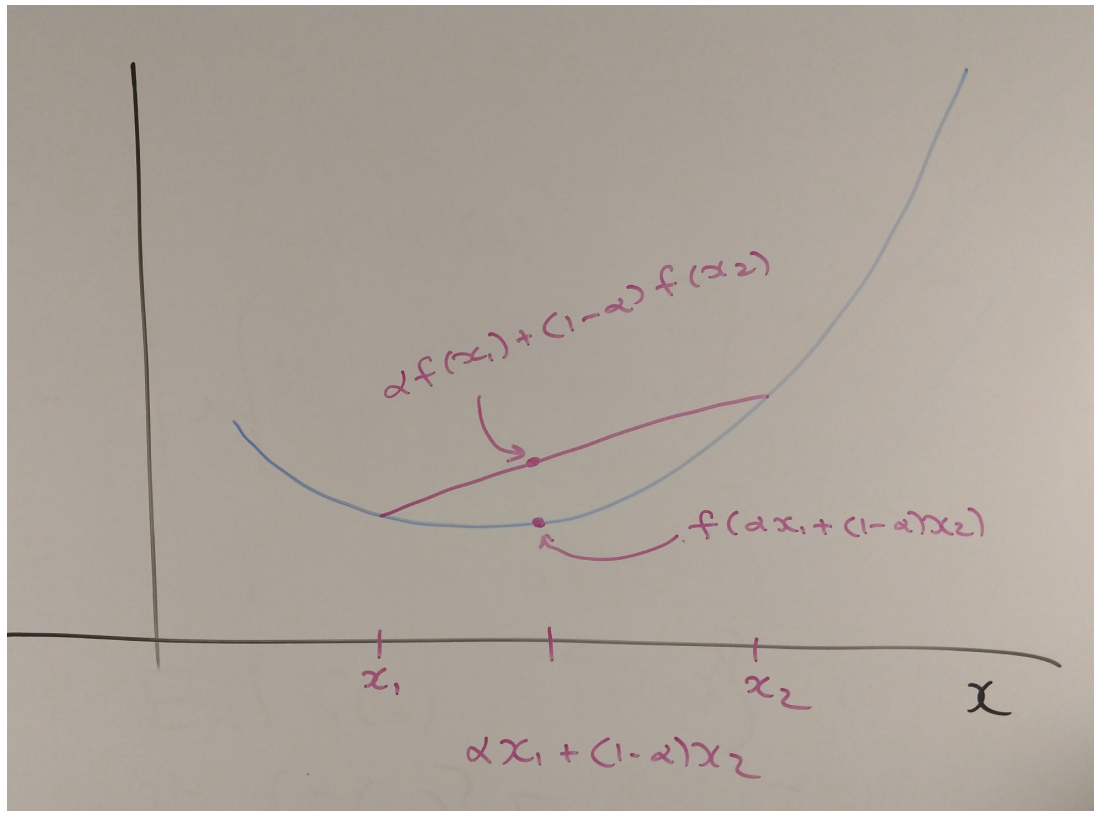


At a local minimum, the derivative is increasing.

Global Sufficient Condition

f is convex if

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2), \alpha \in [0,1].$$



If f is convex and $f'(x_0) = 0$, then x_0 is a global minimum.

We us optimization *a lot* in Machine Learning.

In particular, learning on the training data is often done by some kind of optimization.

For example, in the model $y_i \approx \beta' x_i$ we learn (*estimate*) β by solving

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} (y_i - \beta' x_i)^2$$

We will spend a chunk of time on versions of this problem.

3. Maximum Likelihood, the Bernoulli

Suppose we have a statistical model

$$Y \sim f(y \mid \theta)$$

where θ is the parameter (possibly a vector).

Given data $Y = y$ how can we estimate θ?

Maximum Likelihood:

Choose the θ that makes what you have seen most likely:

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, f(y \mid \theta)$$

In the iid case, we have $Y = (Y_1, Y_2, \ldots, Y_n)$ with

$$Y_i \sim f(y \mid \theta) \;\; iid,$$

so

$$f(y \mid \theta) = \prod_{i=1}^{n} f(y_i \mid \theta),$$

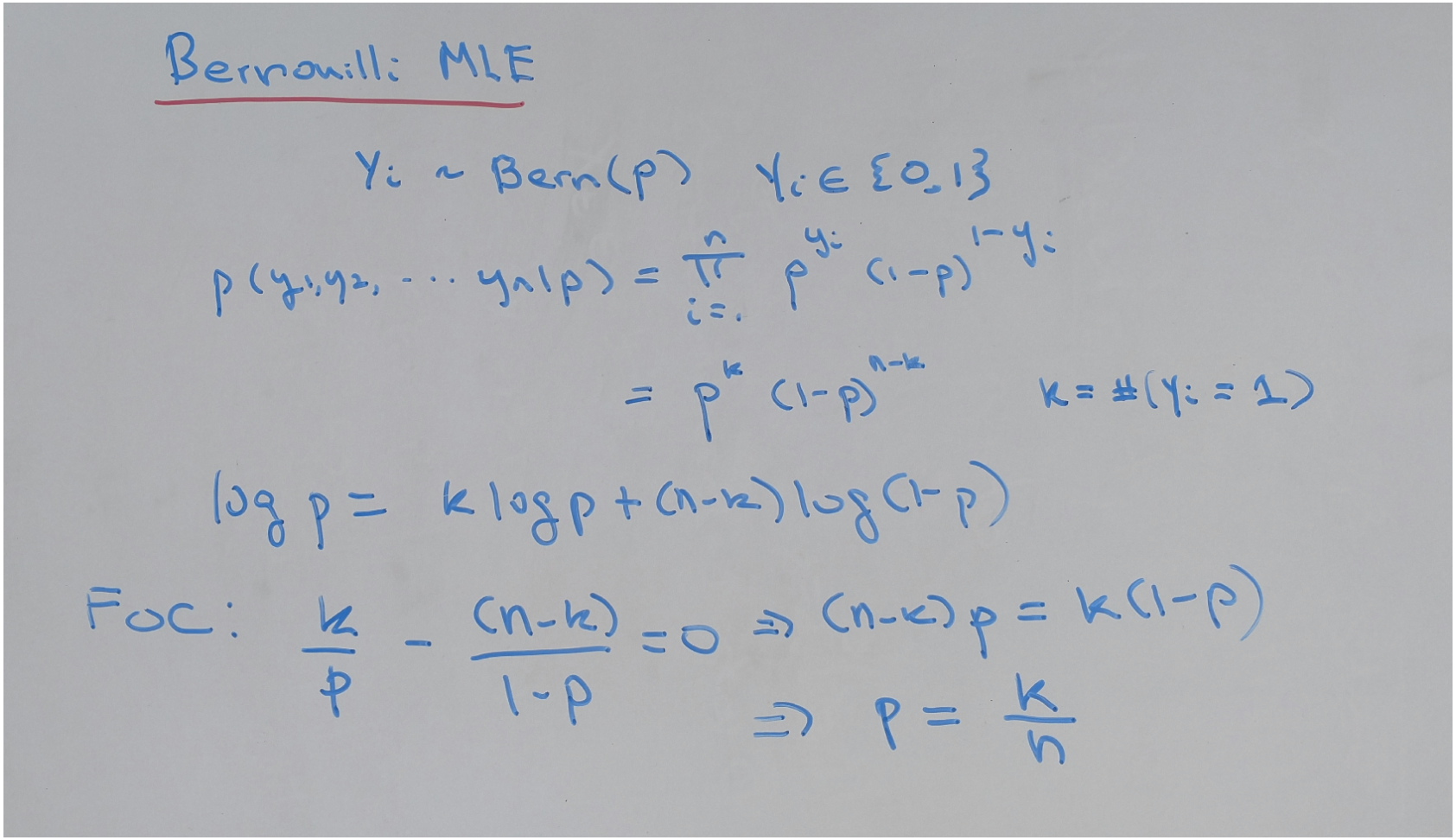
and the MLE is

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \prod_{i=1}^{n} f(y_i \mid \theta).$$

Note:

$f(y \mid \theta)$ viewed as a function of θ for a fixed y is called the likelihood function.

In practice we often maximize the log of the likelihood or minimize the negative of the log likelihood.



FOC: "first order condition", $f' = 0$.

So, *the observed sample frequency is the MLE!*

4. Projecting onto a vector

Let x and $y \in R^n$.

So, for example, $x = (x_1, x_2, \ldots, x_n)'$.

We will find the solution to the following problem very useful:

$$\min_{\beta \in R} ||y - \beta x||^2$$

where $||x||^2 = \sum x_i^2$.

Recall:

$x, y \in R^n$,

The inner product is

$$< x, y >= x'y = y'x = \sum x_i y_i.$$

The L^2 or Euclidean norm (squared) is

$$||x||^2 =< x, x >= x'x = \sum x_i^2$$

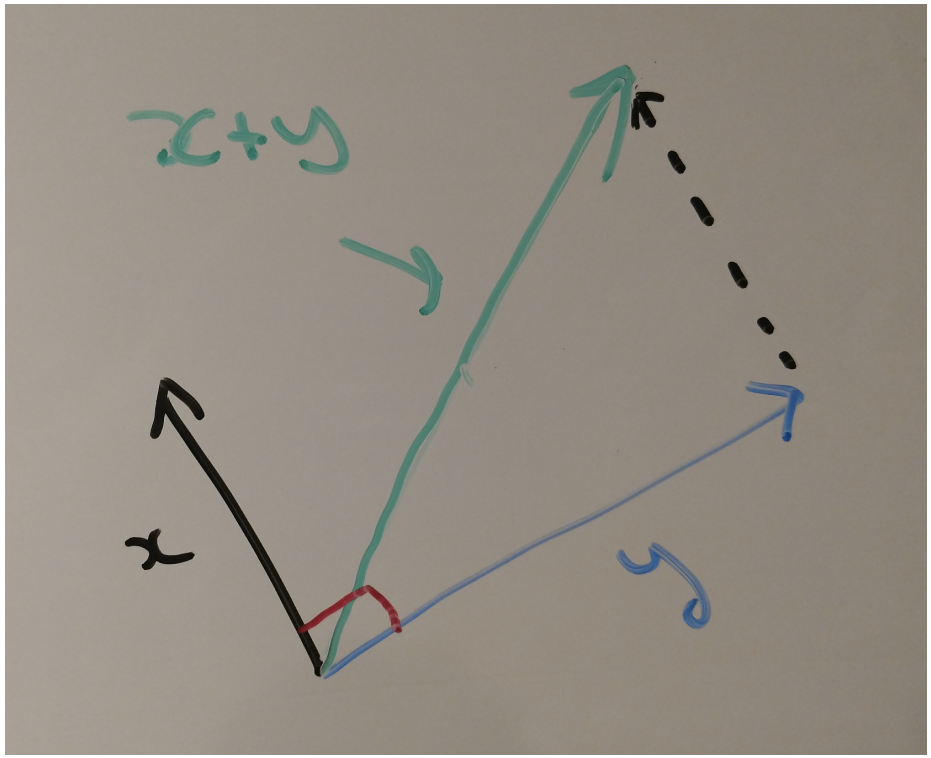
x and y are orthogonal if

$$< x, y >= 0$$

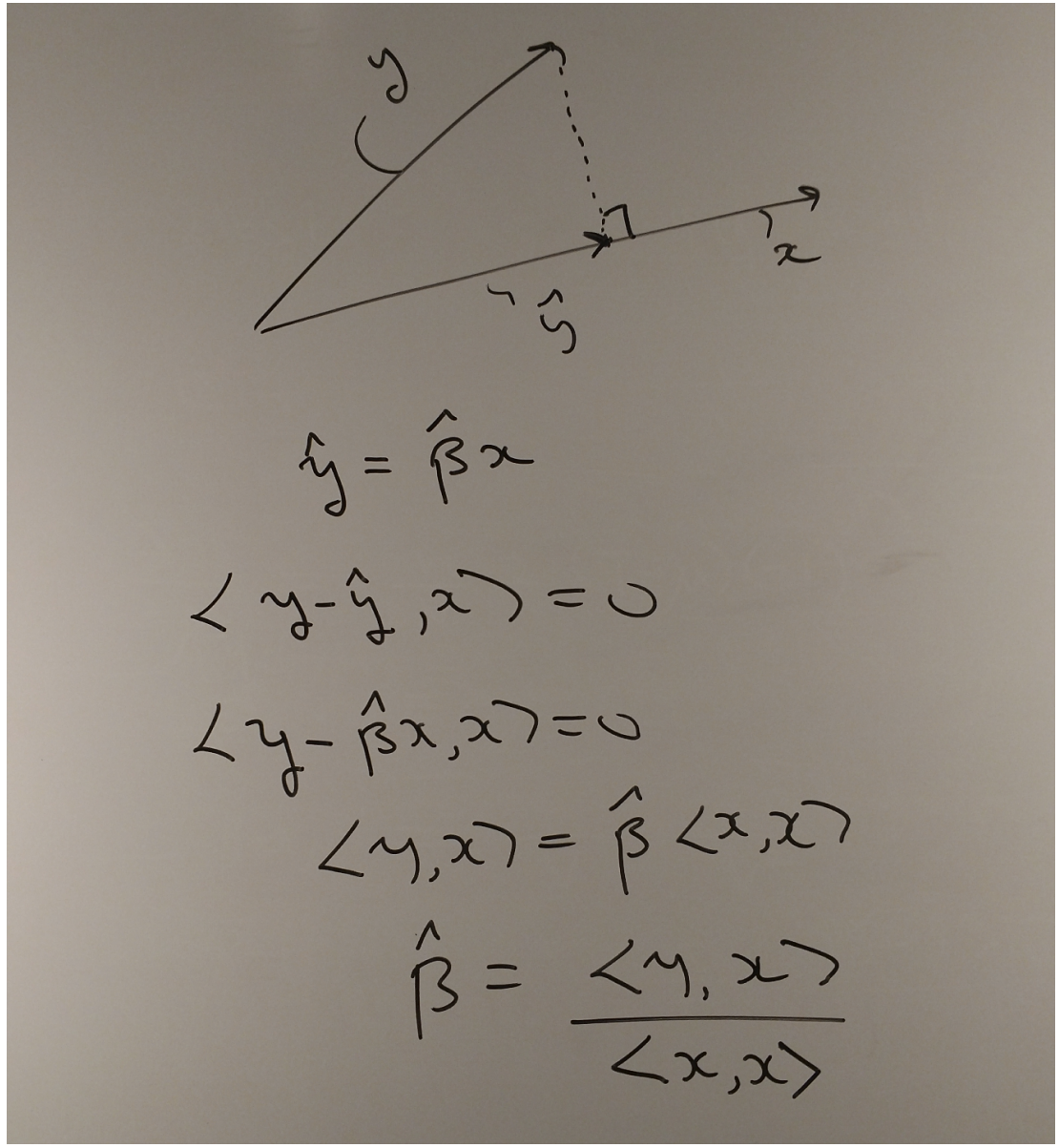
Note:

If x and y are orthogonal:

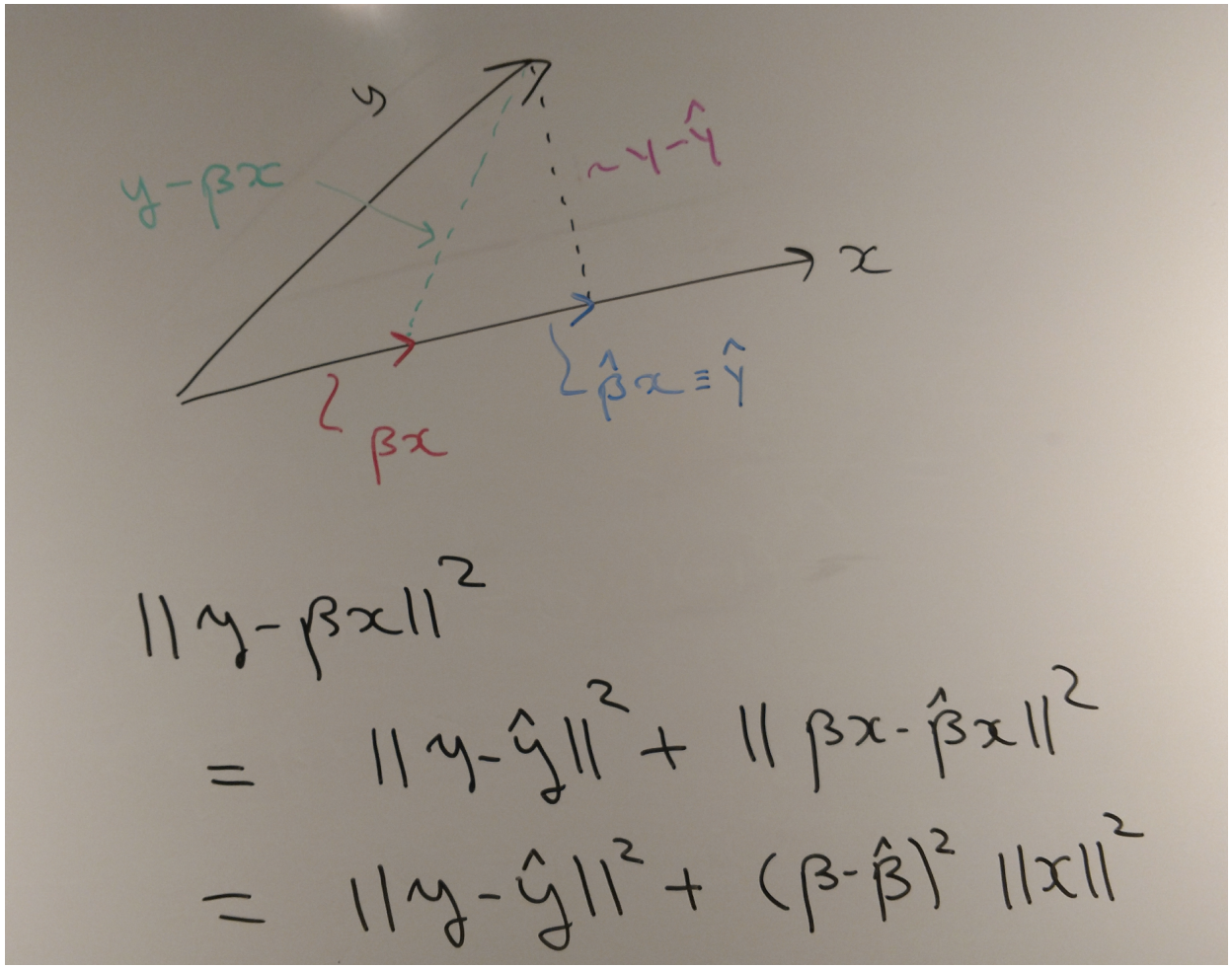
$$\begin{aligned} ||x+y||^2 &= < x+y, x+y > \\ &= < x, x > +2 < x, y > + < y, y > \\ &= ||x||^2 + ||y||^2 \end{aligned}$$



$\hat{y}$ is the orthogonal projection of y onto x.



To solve our problem we have



So that obviously the min is obtained at $\beta^* = \hat{\beta}$.

5. Finding a Minimum, Several Variables

Now suppose $x = (x_1, x_2, \ldots, x_p)'$

and $f(x) = f(x_1, x_2, \ldots, x_p) \in R$.

How do we solve:

$$\min_x f(x)$$

The Gradient:

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_p} \right]$$

where

$$\frac{\partial f(x)}{\partial x_i}$$

is what you get by holding all the x_j, $j \neq i$ fixed, and then differentiating with respect to x_i.

The gradient is a multivariate derivative in that (skipping some technical details):

$$f(x) \approx f(a) + \nabla f(a)(x - a)$$

Note that $\nabla f(x)$ is a row vector so that the product above makes sense with x a column vector.

An alternative notation is:

$$f(x) \approx f(a) + < \nabla f(a), (x - a) >$$

In R^2, this looks like:

$$x = (x_1, x_2)$$
$$a = (a_1, a_2)$$

$$f(x) - f(a) = f(x_1, x_2) - f(a_1, a_2)$$

$$\simeq \frac{\partial f}{\partial x_1}(a_1, a_2)(x_1 - a_1) + \frac{\partial f}{\partial x_2}(a_1, a_2)(x_2 - a_2)$$

$$= \nabla f(a) \begin{bmatrix} x_1 - a_1 \\ x_2 - a_2 \end{bmatrix} = \nabla f(a)\,(x - a)$$

$$= \langle \nabla f(a), (x - a) \rangle$$

$$f(x) \simeq f(a) - \langle \nabla f(a), a \rangle + \langle \nabla f(a), x \rangle$$

$$\equiv C + \frac{\partial f}{\partial x_1}(a)\, x_1 + \frac{\partial f}{\partial x_2}(a)\, x_2$$

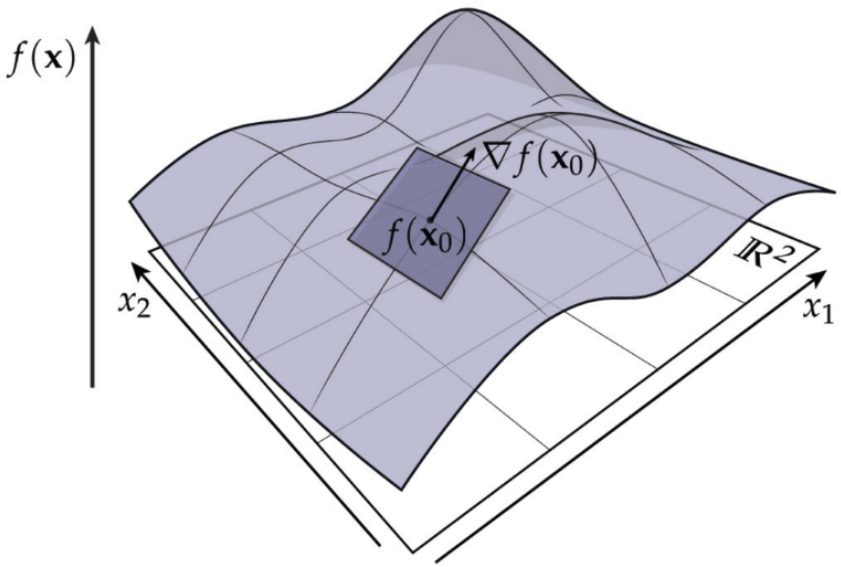
linear approximation to f at a!!!

Stolen off the web, changed a to x_0:

Gradient as Best Linear Approximation

Another way to think about it: at each point x0, gradient is the vector $\nabla f(\mathbf{x}_0)$ that leads to the best possible approximation

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle$$

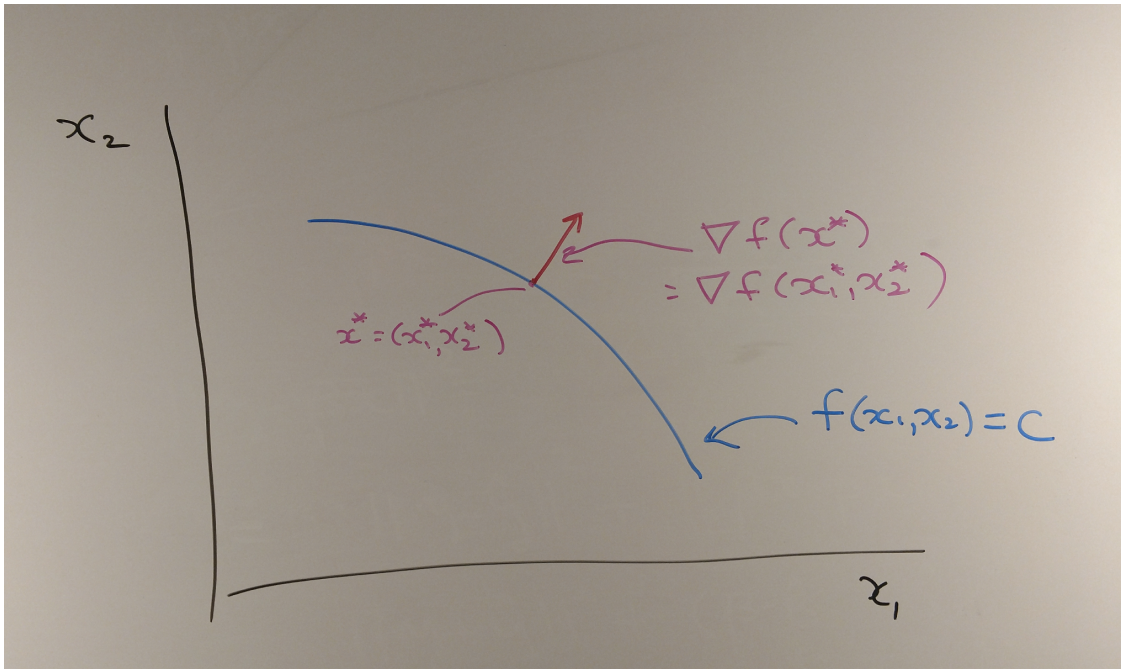


Starting at x_0, this term gets:

- **bigger if we move in the direction of the gradient,**
- **smaller if we move in the opposite direction, and**
- **doesn't change if we move orthogonal to gradient.**

CMU 15-462/662

We can visualize the gradient using the *contours* of f.
A *contour* is the set $\{x : f(x) = c\}$.



- If you want to increase f as fast as possible, go in the direction of the gradient ∇f.
- If you want to decrease f as fast as possible, go in the direction of the negative gradient $-\nabla f$.
- If you want to move without changing f go in a direction orthogonal to the gradient. The gradient is orthogonal (perpendicular) to the contour.

Neccessary Condition for a local min/max:

If x^* is a local min (or max) then we must have

$$\nabla f(x^*) = 0$$

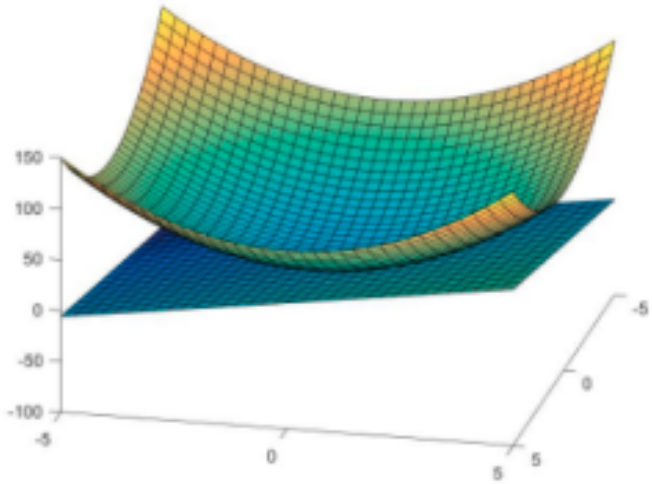
Again f is convex if,

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2), \alpha \in [0,1].$$

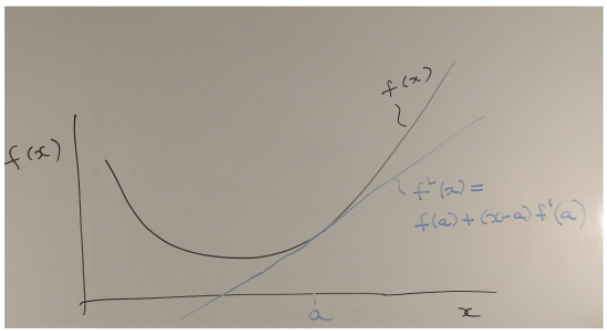
exactly as before except that now x denotes a vector $\in R^p$.

As before, if f is convex, then a local minimum is a global minimum.

Convex function with linear approximation, 2 dimensional:



Convex function with linear approximation, 1 dimensional:



6. Maximum Likelihood, the normal

Suppose

$$Y_i \sim N(\mu, \sigma^2), \quad \text{iid}$$

what is the MLE of $\theta = (\mu, \sigma^2)$?

$$f(y|\mu,\sigma^2) = \prod f(y_i|\mu,\sigma^2)$$

$$= \prod \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{-\frac{1}{2\sigma^2}(y_i-\mu)^2}$$

$$= (2\pi)^{-\frac{n}{2}} \sigma^{-n} e^{-\frac{1}{2\sigma^2}\sum(y_i-\mu)^2}$$

$$-\log L(\mu,\sigma^2) = \frac{n}{2}\log(2\pi) + n\log\sigma + \frac{1}{2\sigma^2}\sum(y_i-\mu)^2$$

Let $v = \sigma^2$

$$= \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(v) + \frac{1}{2v}\sum(y_i-\mu)^2$$

$$-2\log L(\mu,v) = n\log(2\pi) + n\log(v) + \frac{1}{v}\sum(y_i-\mu)^2$$

We want to simplify $\sum(y_i - \mu)^2$.

$$\sum_{i=1}^{n}(y_i - \bar{y}) = \sum y_i - n\bar{y} = n\frac{\sum y_i}{n} - n\bar{y} = 0$$

$$\sum_{i=1}^{n}(y_i - \mu)^2 = \sum\left((y_i - \bar{y}) + (\bar{y} - \mu)\right)^2$$

$$= \sum(y_i - \bar{y})^2 + 2\sum(y_i - \bar{y})(\bar{y} - \mu) + \sum(\bar{y} - \mu)^2$$

$$= \sum(y_i - \bar{y})^2 + 2(\bar{y} - \mu)\sum(y_i - \bar{y}) + n(\bar{y} - \mu)^2$$

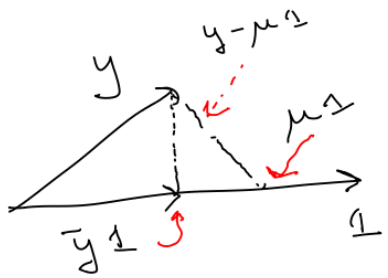
$$= \sum(y_i - \bar{y})^2 + n(\bar{y} - \mu)^2$$

Here is another way.

$$\sum_{i=1}^{n}(y_i-\mu)^2 = \|y-\mu\mathbf{1}\|^2$$

$$y=(y_1,y_2,\cdots y_n)' \qquad \mu\mathbf{1}=\begin{bmatrix}\mu\\ \mu\\ \vdots\\ \mu\end{bmatrix}$$

$$\mathbf{1}=(1,1,\cdots 1)'$$



$$\|y-\mu\mathbf{1}\|^2 = \|y-\bar{y}\mathbf{1}\|^2 + \|\bar{y}\mathbf{1}-\mu\mathbf{1}\|^2$$

$$= \sum(y_i-\bar{y})^2 + n(\bar{y}-\mu)^2$$

$$\bar{y}=\frac{\langle y,\mathbf{1}\rangle}{\langle\mathbf{1},\mathbf{1}\rangle}=\frac{\sum y_i}{n}$$

$S = \sum (y_i - \bar{y})^2.$

$$-2\log L =$$

$$c + n\log(v) + \frac{1}{v}\left[S + n(\bar{y} - \mu)^2\right]$$

$$\frac{\partial}{\partial \mu} = \frac{n}{v}\, 2(\bar{y} - \mu)(-1)$$

$$\Rightarrow \mu^* = \bar{y}$$

$$\frac{\partial}{\partial v}(\text{at } \mu^*) = \frac{n}{v} - \frac{S}{v^2}$$

$$v^* = \frac{S}{n} = \frac{\sum (y_i - \bar{y})^2}{n}$$

7. The Multinoulli MLE

The fundamental Bernoulli random variable considers the case where something is about to happen or not and we code one possibility up as a 1 and the other as a 0.

The Multinoulli distribution consider the more general case where there is a a set of k possible outcomes.

For example, if we survey a customer and ask them to rate our product on a 1-5 scale then there are 5 possible outcomes.

Let $\{1, 2, \ldots, k\}$ denote the possible outcomes for Y.

Let

$$p = (p_1, p_2, \ldots, p_k)$$

with

$$P(Y = j \mid p) = p_j$$

Then

$$Y \sim \text{Multinoulli}(p)$$

Given $Y_i \sim \text{Multinoulli}(p)$ we want to compute the MLE of p.

$$Y_{ij} = \begin{cases} 1 & \text{if } Y_i = j \\ 0 & \text{else} \end{cases} \qquad \begin{array}{l} i = 1, 2, \cdots n \\ j = 1, 2, \cdots k \end{array}$$

$$p(y_1, y_2, \cdots y_n | p) = \prod_i p_1^{y_{i1}} p_2^{y_{i2}} \cdots p_k^{y_{ik}}$$

$$= p_1^{m_1} p_2^{m_2} \cdots p_k^{m_k}$$

$$m_j = \sum_i Y_{ij} = \left[\#\text{ of times } Y_i = j \right]$$

How do we maximize this likelihood?

With just two possible outcomes we had one variable,
$p = P(Y = 1)$.

Now we have $p_j, j = 1, 2, \ldots, k$ with the constraint $\sum p_j = 1$.

We also have $0 \leq p_j \leq 1$, but we won't have to worry about this.

We could let $p_k = 1 - \sum_{j=1}^{k-1} p_j$ and then optimize over
$(p_1, p_2, \ldots, p_{k-1})$.

But, it is easier to use *lagrange multipliers*.

8. Lagrange Multiplier

Let $x \in R^p$.

We want to solve:

$$\min_x f(x), \quad \text{subject to} \quad g(x) = 0$$

Let

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

and then minimize $\mathcal{L}$ unconstrained over (x, λ).

Differentiating $\mathcal{L}$ with respect to λ gives:

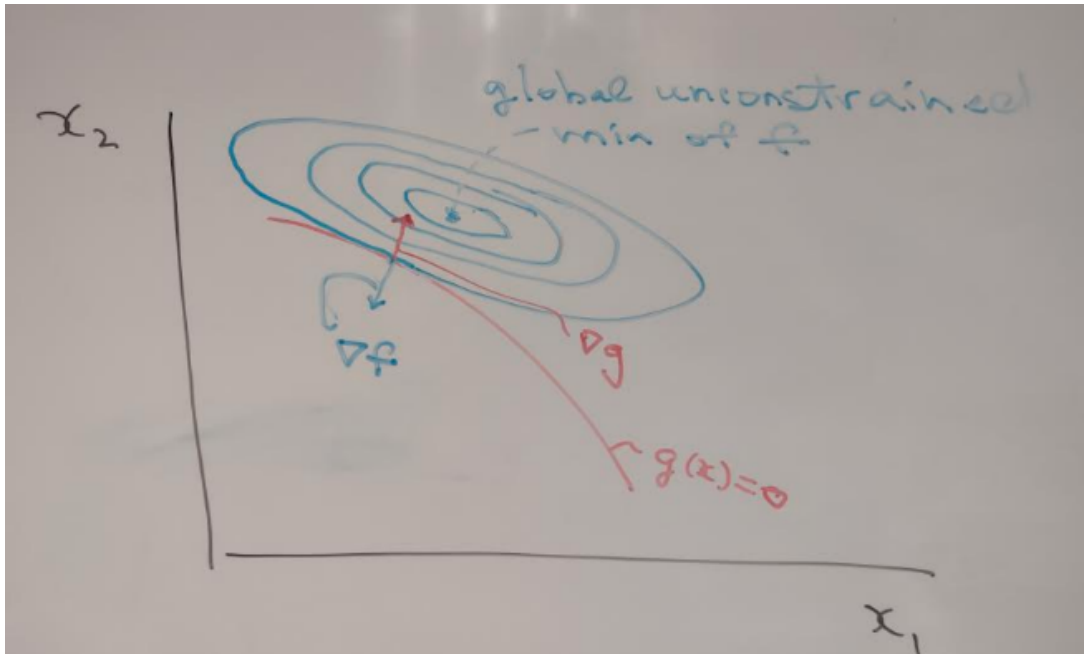
$$g(x) = 0$$

at the min/max.

Differentiating $\mathcal{L}$ with respect to x give:

$$\nabla f(x) + \lambda \nabla g(x) = 0$$

at a local min (or max).
Because of the constraint $g(x) = 0$ you can only move orthogonal to ∇g.



But, $\nabla f \propto \nabla g$, tells you that "small" moves orthogonal to ∇g will not change f so it is a local minimum or maximum.

9. The Multinoulli MLE again

To obtain the Multinoulli MLE we will have

$$L(p) = \prod p_j^{m_j}$$

and we maximize this subject to

$$\sum p_j = 1.$$

We will max the log likelihood:

$$\mathcal{L}(p, \lambda) = \sum_j m_j log(p_j) + \lambda(\sum_j p_j - 1)$$

$$L = \sum m_k \log p_k + \lambda\left(\sum p_k - 1\right)$$

$$\frac{\partial L}{\partial p_k} = \frac{m_k}{p_k} + \lambda$$

$$\Rightarrow \quad p_k \propto m_k$$

$$\Rightarrow \quad \hat{p}_k = \frac{m_k}{\sum m_k} = \frac{m_k}{n}.$$

The MLE is the observed sample frequency.

10. KKT

We will have occasion to consider constraint sets of the form

$$g(x) \leq 0$$

rather than just

$$g(x) = 0$$

The Karush-Kuhn-Tucker conditions cover both inequality and equality constraints.

We'll see how things change with one inequality constraint and then state the general result.

KKT:

To minimize $f(x)$ subject to $g(x) \leq 0$, form

$$L(x, \alpha) = f(x) + \alpha g(x)$$

and then solve

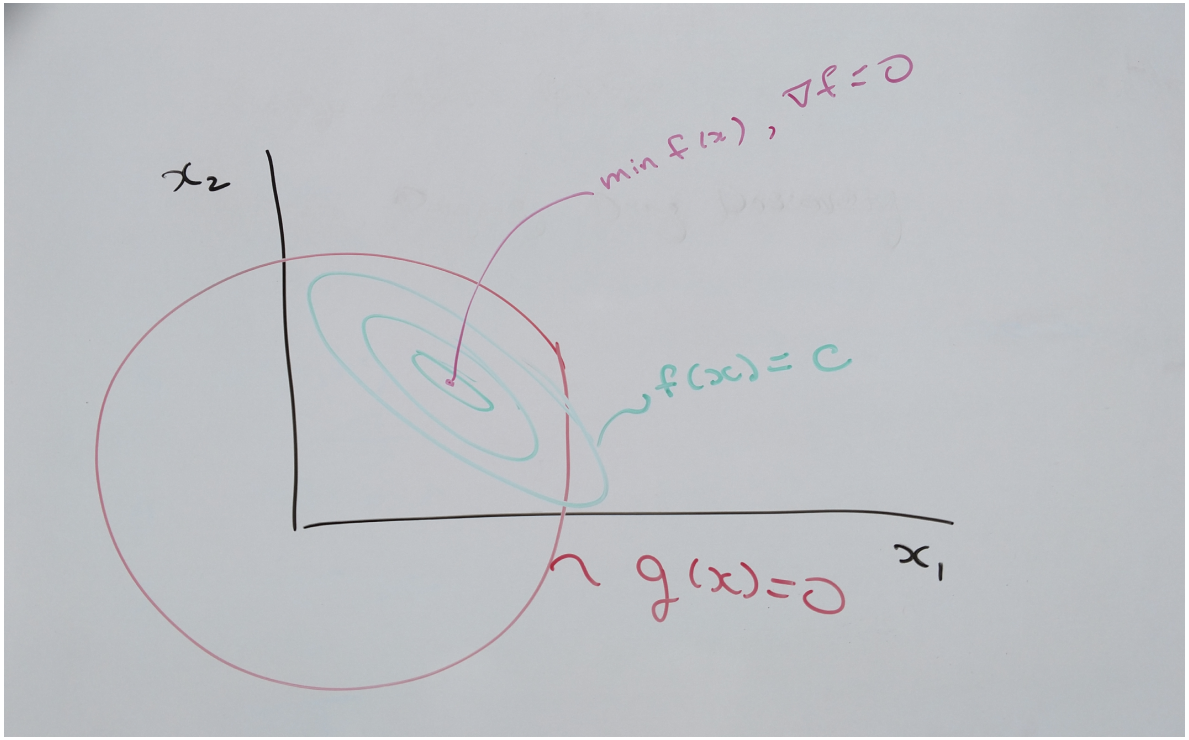
$$min_x max_{\alpha, \alpha \geq 0} \; L(x, \alpha).$$

With $\alpha \geq 0$ we must have $g(x) \leq 0$, since otherwise we could get a max of infinity.

This looks hard to understand, but it just boils down to there are two case depending on whether the constraint is "binding" or not.

Here is the case where the constraint is not binding.

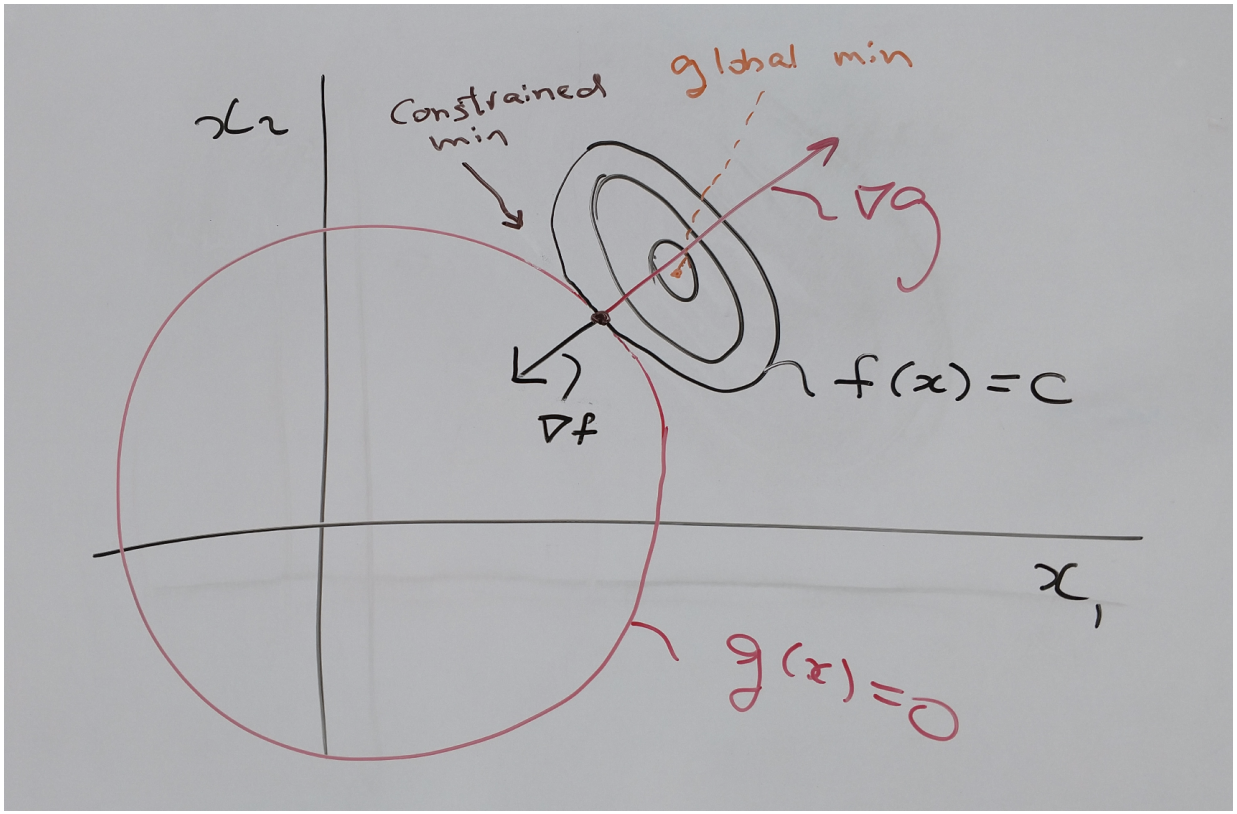
The global min is in the interior of the set $g(x) \leq 0$.



The neccessary condition is just $\nabla f = 0$.

Here is the key picture for the case where the constraint is binding.

Remember, ∇f is the direction in which f goes up the fastest!!
∇f points perpendicularly to the contour of f.



It is intuitive that $\nabla f + \alpha \nabla g = 0$ with $\alpha > 0$.

Example:

What happens when we do

$$\min_{x:||x||\leq c} a'x$$

What happens when we do

$$\max_{x:||x||\leq c} a'x$$

With $c = 1$.

$$g(x) = x_1^2 + x_2^2 - 1$$

$$f(x) = a^T x = [a_1, a_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = a_1 x_1 + a_2 x_2$$

$$\nabla g(x) = [2x_1, 2x_2]$$

$$\nabla f(x) = [a_1, a_2] = a^T$$

At min:

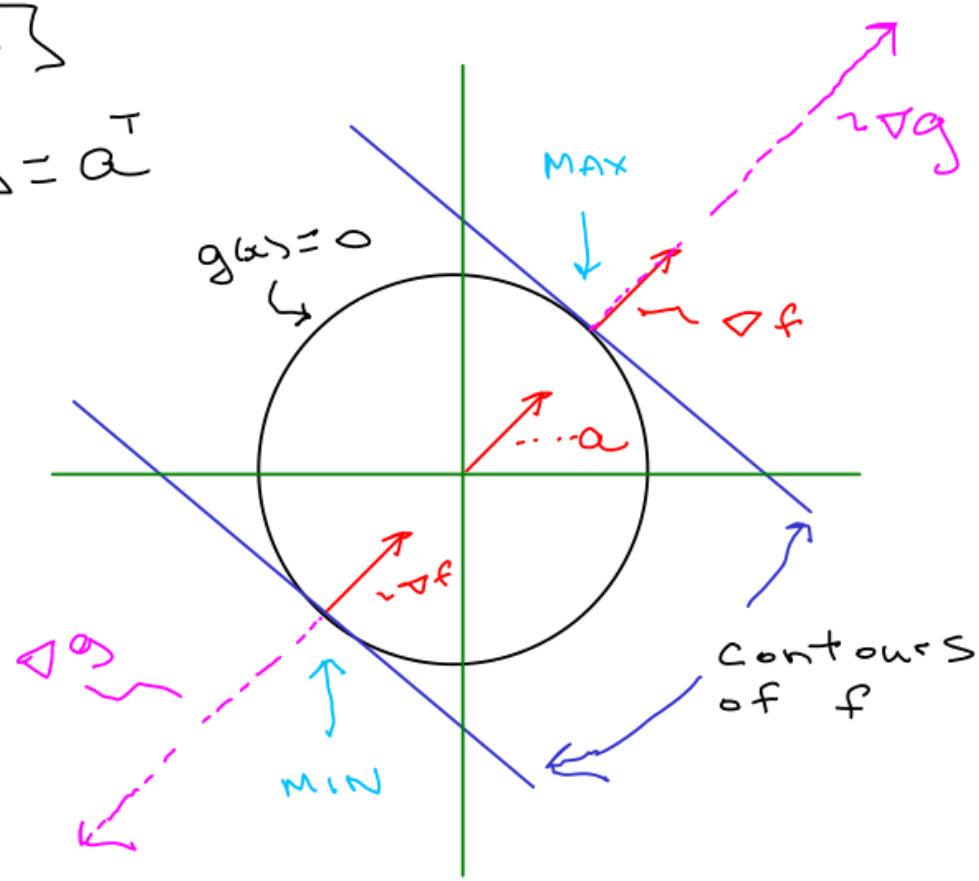
$$\nabla f + \alpha \nabla g = 0$$

$$\alpha > 0$$

At max

$$\nabla f + \alpha \nabla g = 0$$

$$\alpha < 0$$



Solve for MIN

$\nabla f + \alpha \nabla g \quad \alpha > 0$

$$[a_1, a_2] + 2\alpha [x_1, x_2] = 0$$

$$a_i + 2\alpha x_i = 0$$

$$x_i = -\frac{a_i}{2\alpha}$$

$$x_1^2 + x_2^2 = 1 \Rightarrow x_i = \frac{-a_i}{\sqrt{a_1^2 + a_2^2}}$$

$$x^* = -\frac{a}{\|a\|}$$

Solve for MAX

$$[a_1, a_2] - 2\alpha [x_1, x_2] = 0 \quad \alpha > 0$$

$$x_i = \frac{a_i}{2\alpha} \Rightarrow x_i = \frac{a_i}{\sqrt{a_1^2 + a_2^2}}$$

$$x^* = \frac{a}{\|a\|}$$

11. Gradient Descent

To find the MLE for the Bernoulli parameter p we solved

$$\frac{dl(p; y)}{dp} = 0$$

where

$$l(p; y) = \log(p(y \mid p)), \quad y = (y_1, y_2, \ldots, y_n).$$

To find the MLE for the normal (μ, v) parameter (with $v = \sigma^2$) we solved

$$\nabla M(\mu, v) = [\frac{\partial M}{\partial \mu}(\mu, v), \frac{\partial M}{\partial v}(\mu, v)] = 0$$

where

$$M(\mu, v) = -\log(f(y \mid \mu, v)), \quad y = (y_1, y_2, \ldots, y_n).$$

When optimizing more complex functions $f(\theta)$ it may not be easy to solve

$$\nabla f(\theta) = 0.$$

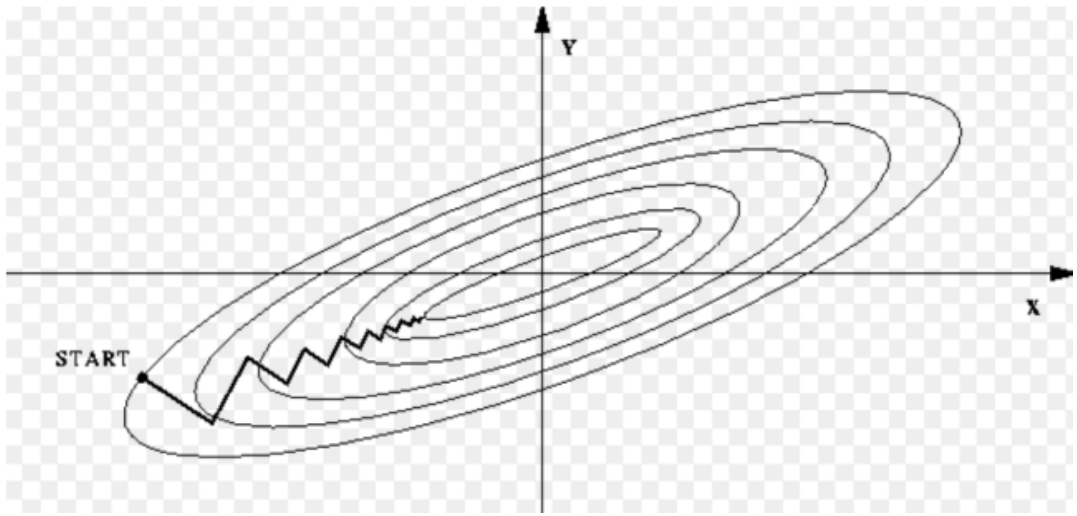
It is very common in Machine Learning to use an iterative approach where at each iteration we change the current θ my moving in the direction indicated by the gradient:

$$\theta \to \theta - \epsilon \nabla f(\theta).$$

ϵ is called the learning rate.

A lot of the "optimization" in Machine Learning is about schemes for choosing the learning rate and letting it change as we iterate.

Learning rate: you want to go in the direction of $\nabla f(\theta)$, but how far should you go?



This is the basic idea, but in practice it is more complicated:

- stochastic gradient descent
- learning rate strategies