Today's Specials

- Detailed look at Lagrange Multipliers
- Forward-Backward and Viterbi algorithms for HMMs
- Intro to EM as a concept [Motivation, Insights]

Lagrange Multipliers

- Why is this used ?
- I am in NLP. Why do I care ?
- How do I use it ?
- Umm, I didn't get it. Show me an example.
- Prove the math.
- Hmm... Interesting !!

Constrained Optimization

- Given a metal wire, f(x,y) : $x^2+y^2=1$

  Its temperature $T(x,y) = x^2+2y^2-x$

  Find the hottest and coldest points on the wire.
- Basically, determine the optima of T subject to the constraint 'f'
- How do you solve this ?

Ha ... That's Easy !!

- Let $y = \sqrt{1-x^2}$ and substitute in T
- Solve T for x

How about this one?

- Same T
- But now,

$$f(x,y):(x^2+y^2)^2-x^2+y^2=0$$

- Still want to solve for y and substitute?
- Didn't think so !

All Hail Lagrange !

- Lagrange's Multipliers [LM] is a tool to solve such problems [& live through it]
- Intuition:
  - For each constraint 'i', introduce a new scalar variable – L_i (the Lagrange Multiplier)
  - Form a linear combination with these multipliers as coefficients
  - Problem is now unconstrained and can be solved easily

Use for NLP

- Think EM
  - The “M” step in the EM algorithm stands for “Maximization”
  - This maximization is also constrained
  - Substitution does not work here either

- If you are not sure how important EM is, stick around, we'll tell you !

Vector Calculus 101

- A gradient of a function is a vector :
  - Direction : direction of the steepest slope uphill
  - Magnitude : a measure of steepness of this slope

- Mathematically, the gradient of f(x,y) is:

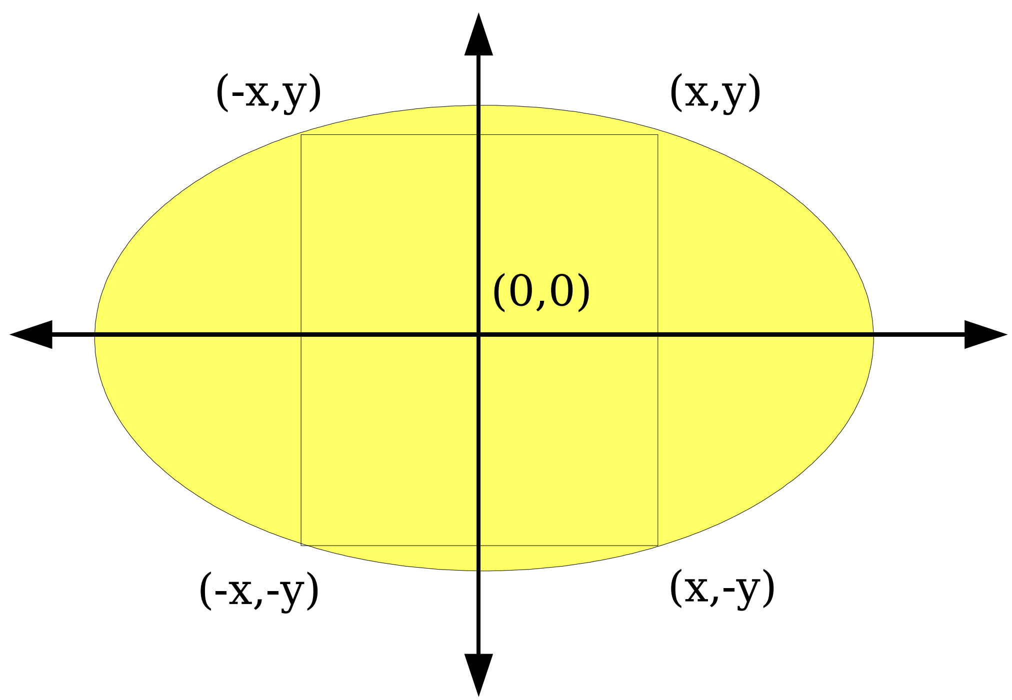
$$\mathrm{grad}(f(x,y)) = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

How do I use LM ?

- Follow these steps:
  - Optimize f, given constraint: g = 0
  - Find gradients of 'f' & 'g', grad(f) & grad(g)
  - Under given conditions, grad(f) = L * grad(g) [proof coming]
  - This will give 3 equations (one each for x, y and z)
  - Fourth equation : g = 0
  - You now have 4 eqns & 4 variables [x,y,z,L]
  - Feed this system into a numerical solver
  - This gives us (x_p,y_p,z_p) where f is maximum. Find f_{max}
  - Rejoice !

Examples are for wimps !

What is the largest square that can be inscribed in the ellipse $x^2+2y^2=1$?

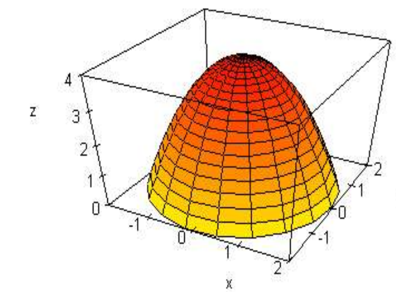


Area of Square = 4xy

And all that math ...

- Maximize f = 4xy subject to $x^2+2y^2=1$
- grad(f) = [4y, 4x], grad(g) = [2x, 4y]
- Solve:
  - 2y – Lx = 0
  - x – Ly = 0
  - $x^2+2y^2-1=0$
- Solution : $(x_p, y_p) = \left(\frac{\sqrt{2}}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$ & $\left(-\frac{\sqrt{2}}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right)$
- $f_{max} = 4\sqrt{2}/3$

Why does it work?



- Think of an f, say, a paraboloid
- Its “level curves” will be enclosing circles
- Optima points lie along g and on one of these circles
- 'f' and 'g' MUST be tangent at these points:
  - If not, then they cross at some point where we can move along g and have a lower or higher value of f
  - So this cannot be an point of optima, but it is!
  - Therefore, the 2 curves are tangent.
- Therefore, their gradients(normals) are parallel
- Therefore, grad(f) = L * grad(g)

Expectation Maximization

- We are given data that we assume to be generated by a stochastic process
- We would like to fit a model to this process, i.e., get estimates of model parameters
- These estimates should be such that they maximize the likelihood of the observed data – MLE estimates
- EM does precisely that – and quite efficiently

Obligatory Contrived Example

- Let observed events be grades given out in a class
- Assume that there is a stochastic process generating these grades (yeah ... right !)
- $P(A) = 1/2$, $P(B) = \mu$, $P(C) = 2\mu$, $P(D) = \frac{1}{2} - 3\mu$
- Observations:
  - Number of A's = 'a'
  - Number of B's = 'b'
  - Number of C's = 'c'
  - Number of D's = 'd'
- What is the ML estimate of 'μ' given a,b,c,d ?

Obligatory Contrived Example

- $P(A) = ½, P(B) = μ, P(C) = 2μ, P(D) = ½ – 3μ$
- $P(Data | Model) = P(a,b,c,d | μ) = K (½)^a(μ)^b(2μ)^c(½-3μ)^d$ = Likelihood
- $\log P(a,b,c,d | μ) = \log K + a \log ½ + b \log μ + c \log 2μ + d \log (½-3μ)$
  = Log Likelihood [easier to work with this, since we have sums instead of products]
- To maximize this, set $\partial \mathrm{LogP}/\partial μ = 0$
- $\frac{b}{\mu}+\frac{2c}{2\mu}-\frac{3d}{1/2-3\mu}=0$ => $μ = \frac{b+c}{6(b+c+d)}$
- So, if the class got 10 A's, 6 B's, 9 C's and 10 D's, then $μ = 1/10$
- This is the regular and boring way to do it
- Let's make things more interesting ...

Obligatory Contrived Example

- $P(A) = ½$, $P(B) = \mu$, $P(C) = 2\mu$, $P(D) = ½ – 3\mu$
- A part of the information is now hidden:
  - Number of high grades (**A's +B's**) = h
- What is an ML estimate of μ now?
- Here is some delicious circular reasoning:
  - If we knew the value of μ, we could compute the expected values of 'a' and 'b' EXPECTATION
  - If we knew the values of 'a' and 'b', we could compute the ML estimate for μ MAXIMIZATION
- Voila ... EM !!

Obligatory Contrived Example

Dance the EM dance

- Start with a guess for μ
- Iterate between Expectation and Maximization to improve our estimates of μ and b:
  - $\mu(t)$, $b(t)$ = estimates of μ & b on the t'th iteration
  - $\mu(0)$ = initial guess
  - $b(t) = \mu(t) / (\frac{1}{2} + \mu(t)) = E[b \mid \mu]$: **E-Step**
  - $\mu(t) = (b(t) + c) / 6(b(t) + c + d)$: **M-step** [**Maximum** LE of μ given $b(t)$]
  - Continue iterating until convergence
- Good news : It **will** converge to a maximum.
- Bad news : It will converge to **a** maximum

Where's the intuition?

- Problem: Given some measurement data X, estimate the parameters Ω of the model to be fit to the problem
- Except there are some nuisance “hidden” variables Y which are not observed and which we want to integrate out
- In particular we want to maximize the posterior probability of Ω given data X, marginalizing over Y:

$$\Omega' = \underset{\Omega}{\operatorname{argmax}} \sum_{Y} P(\Omega, Y \mid X)$$

- The E-step can be interpreted as trying to construct a lower bound for this posterior distribution
- The M-step optimizes this bound, thereby improving the estimates for the unknowns

So people actually use it?

- Umm … yeah !
- Some fields where EM is prevalent:
  - Medical Imaging
  - Speech Recognition
  - Statistical Modelling
  - NLP
  - Astrophysics
- Basically anywhere you want to do parameter estimation

... and in NLP ?

- You bet.

- Almost everywhere you use an HMM, you need EM:
  - Machine Translation
  - Part-of-speech tagging
  - Speech Recognition
  - Smoothing

Where did the math go?

We have to do SOMETHING in the next class !!!