Intro to Machine Learning (CS436/CS580L)

Lecture 4: Linear Algebra & Multivariate Calculus

Xi Peng, Fall 2018

Thanks to Tom Mitchell, Andrew Ng, Ben Taskar, Carlos Guestrin, Eric Xing, Hal Daume III, David Sontag, Jerry Zhu, Tina Eliassi-Rad, and Chao Chen for some slides & teaching material.

This Class

- Vectors, Matrices.
- Calculus
- Derivation with respect to a vector.
- Lagrange multiplier

Linear Algebra

- Vectors
  - A one dimensional array.
  - If not specified, assume x is a column vector.
- Matrices
  - Higher dimensional array.
  - Typically denoted with capital letters.
  - n rows by m columns

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$$

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \dots & a_{n-1,m-1} \end{pmatrix}$$

Transposition

- **Transposing** a matrix swaps columns and rows.

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{pmatrix}$$

$$\mathbf{x}^T = \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix}$$

Transposition

- **Transposing** a matrix swaps columns and rows.

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m-1} \end{pmatrix}$$

$$A^T = \begin{pmatrix} a_{0,0} & a_{1,0} & \cdots & a_{n-1,0} \\ a_{0,1} & a_{1,1} & & a_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{0,m-1} & a_{1,m-1} & \cdots & a_{n-1,m-1} \end{pmatrix}$$

Addition

- Matrices can be added to themselves iff they have the same dimensions.
  - A and B are both n-by-m matrices.

$$A + B = \begin{pmatrix} a_{0,0} + b_{0,0} & a_{0,1} + b_{0,1} & \cdots & a_{0,m-1} + b_{0,m-1} \\ a_{1,0} + b_{1,0} & a_{1,1} + b_{1,1} & & a_{1,m-1} + b_{1,m-1} \\ \vdots & & \ddots & \vdots \\ a_{n-1,0} + b_{n-1,0} & a_{n-1,1} + b_{n-1,1} & \cdots & a_{n-1,m-1} + b_{n-1,m-1} \end{pmatrix}$$

Hadamard Product

- Element-wise product (like addition)
  - A and B are both n-by-m matrices.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} a_{11}\, b_{11} & a_{12}\, b_{12} & a_{13}\, b_{13} \\ a_{21}\, b_{21} & a_{22}\, b_{22} & a_{23}\, b_{23} \\ a_{31}\, b_{31} & a_{32}\, b_{32} & a_{33}\, b_{33} \end{pmatrix}$$

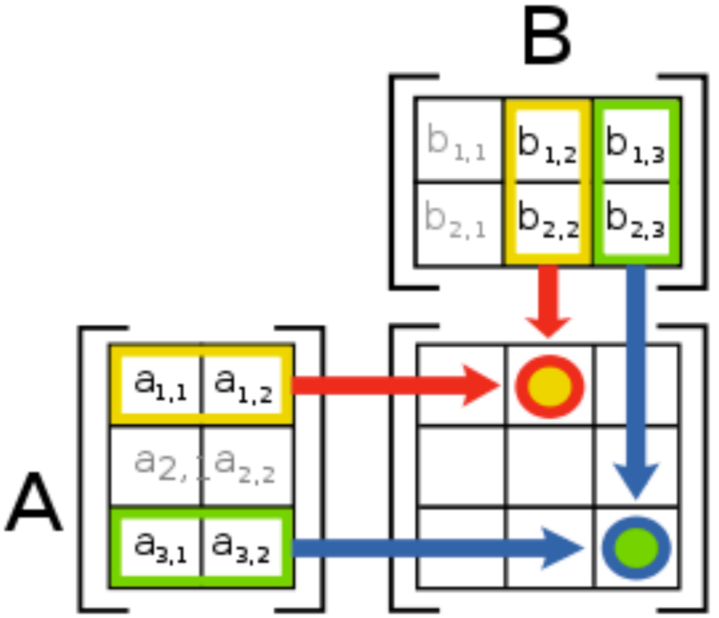
- Multiplies with a scalar

Multiplication

- To multiply two matrices, the **inner dimensions** must be the same.
  – An n-by-m matrix can be multiplied by an m-by-k matrix

$$AB = C$$

$$c_{ij} = \sum_{k=0}^{m} a_{ik} * b_{kj}$$

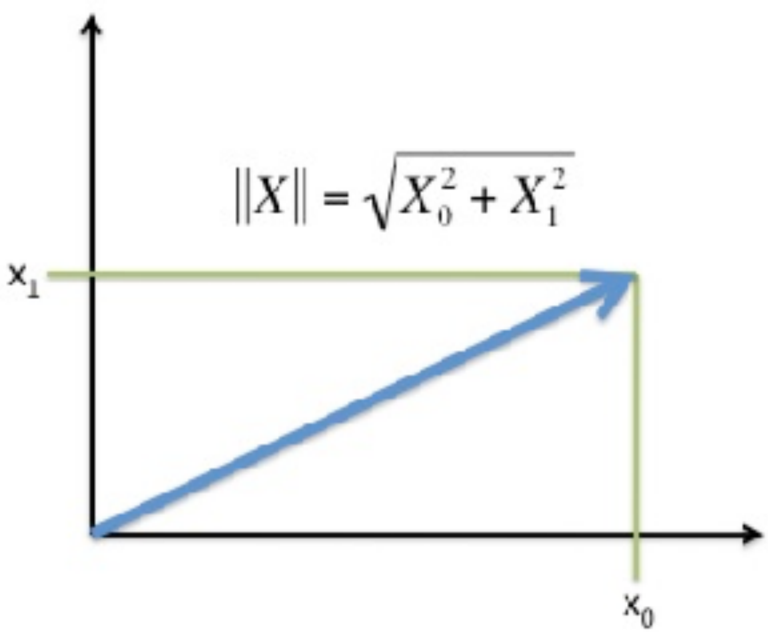


Norm

- The **norm** of a vector, x, represents the Euclidean length of a vector.

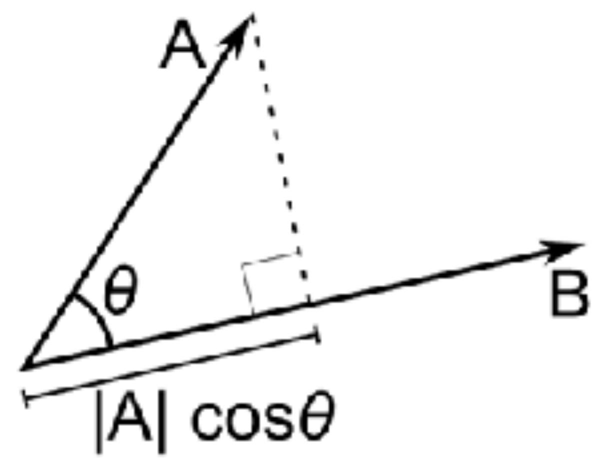
$$||\mathbf{x}|| = \sqrt{\sum_{i=0}^{n-1} x_i^2}$$

$$= \sqrt{x_0^2 + x_1^2 + \ldots + x_{n-1}^2}$$



Operations on Vectors

- $\mathbf{u} + \mathbf{v}$, $\mathbf{u} - \mathbf{v}$, $k\mathbf{u}$
- Dot product: $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = <\mathbf{u}, \mathbf{v}> = ?$
- Norm: $||\mathbf{u}||^2 = <\mathbf{u}, \mathbf{u}>$
- Geometric views
  - $<\mathbf{u}, \mathbf{v}> = ||\mathbf{u}|| \; ||\mathbf{v}|| \cos()$
  - $|<\mathbf{u}, \mathbf{v}>| \quad ||\mathbf{u}|| \; ||\mathbf{v}||$



- Q: projection of $\mathbf{u}$ on $\mathbf{v}$ direction?

View Matrices Differently

- Matrix: linear mapping
- Row, column views of matrices,
  - two views for matrix multiplication
- Transposition rules

$$
\begin{aligned}
(\mathbf{A}+\mathbf{B})^T &= \mathbf{A}^T+\mathbf{B}^T \\
(\mathbf{A}\mathbf{B})^T &= \mathbf{B}^T\mathbf{A}^T
\end{aligned}
$$

- Sanity check: dimensions (A^TB^T)

Achieve Computations without Loops

- Loop is very expensive in matlab/python
  - Avoid using it as much as possible
  - In python:
    - numpy matrix operations, list comprehension
    - http://www.jesshamrick.com/2012/04/29/the-demise-of-for-loops/
    - http://codereview.stackexchange.com/questions/38580/fastest-way-to-iterate-over-numpy-array
  - Example: entropy computation
- Exercise:
  - compute pairwise distance of given data set $X = [x_1, x_2, \ldots, x_N]^T$ without loop? (Output: NxN)

Distributive/Commutative Law,

- $A(B+C) = AB + AC$
  - Matrix/vector/scalar
- $(A+B)(C+D) = ?$
- $||u-v|| = ?$
- Commutative law:
  - Only if inner product: $\langle u,v \rangle = \langle v,u \rangle$
  - Also scalar $kA = Ak$

Inversion

- The inverse of an n-by-n or **square** matrix A is denoted A^{-1}, and has the following property.

$$AA^{-1} = I$$

- Where *I* is the **identity** matrix is an n-by-n matrix with ones along the diagonal.
  - $I_{ij} = 1$ iff $i = j$, 0 otherwise

Identity Matrix

- Matrices are invariant under multiplication by the identity matrix.

$$AI = A$$

$$IA = A$$

- What if A is m x n ?

Helpful matrix inversion properties

$$(A^{-1})^{-1} = A$$

$$(kA)^{-1} = k^{-1}A^{-1}$$

$$(A^T)^{-1} = (A^{-1})^T$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

Positive Definite-ness

- Quadratic form
  - Scalar $c_0 + c_1 x + c_2 x^2$
  - Vector $x^T A x$
- Positive Definite matrix M $x^T M x > 0$
- Positive Semi-definite $x^T M x \geq 0$

Positive Definite-ness

- Quadratic form
  - Scalar $c_0 + c_1 x + c_2 x^2$
  - Vector $x^T A x$
- Positive Definite matrix M $x^T M x > 0$
- Positive Semi-definite $x^T M x \geq 0$
- $A^T A$ is PSD, why?
- If M is symmetric and PSD, there is PSD A s.t. $M = A^T A$

Covariance Matrix

$$\sigma^2 = \mathrm{var}(X) = \mathrm{E}[(X - \mathrm{E}(X))^2] = \mathrm{E}[(X - \mathrm{E}(X)) \cdot (X - \mathrm{E}(X))].$$

$$\Sigma = \mathrm{E}\left[\left(\mathbf{X} - \mathrm{E}[\mathbf{X}] \right) \left(\mathbf{X} - \mathrm{E}[\mathbf{X}] \right)^{\mathrm{T}} \right]$$

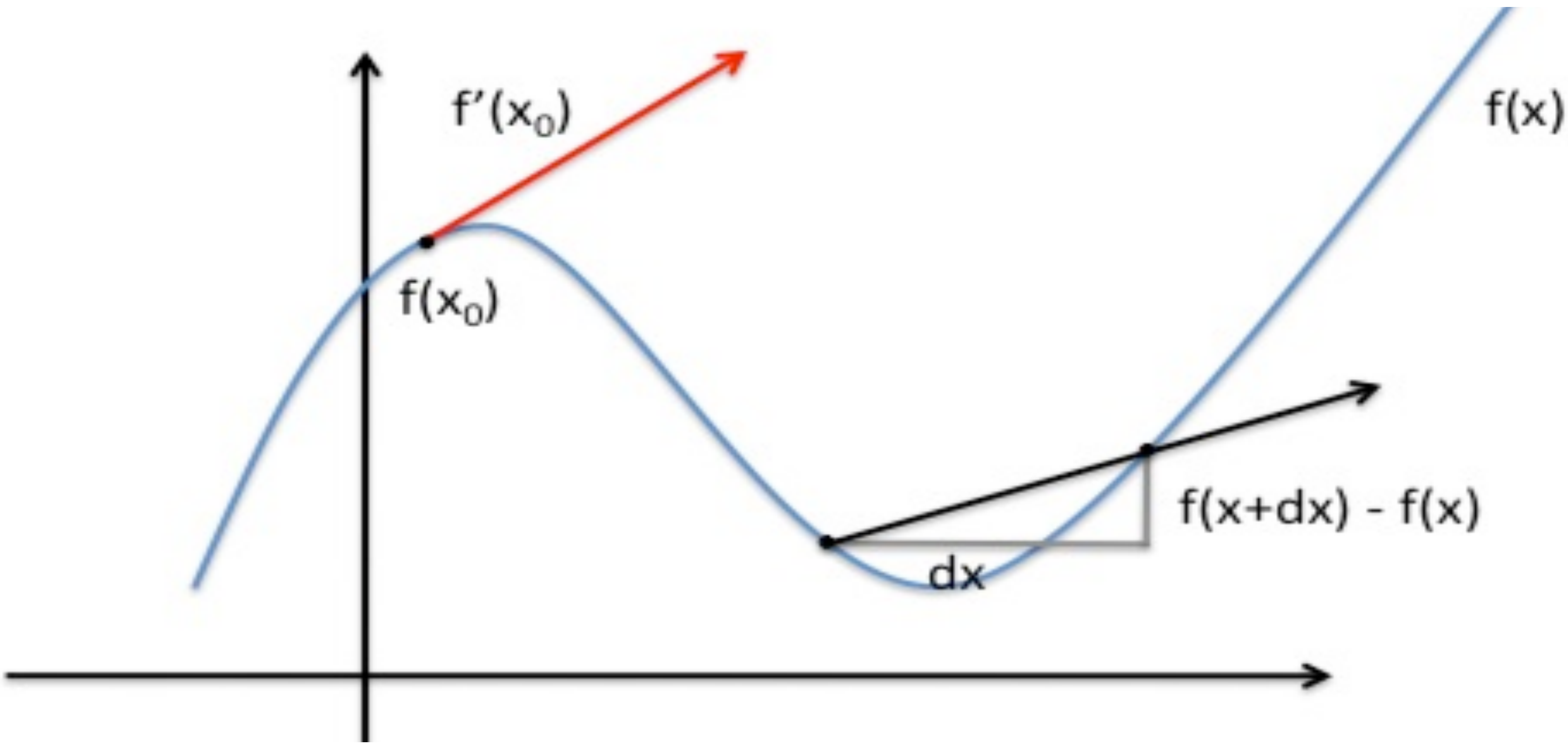
$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \mathrm{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_1 - \mu_1)(X_n - \mu_n)] \\ \\ \mathrm{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \mathrm{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_2 - \mu_2)(X_n - \mu_n)] \\ \\ \vdots & \vdots & \ddots & \vdots \\ \\ \mathrm{E}[(X_n - \mu_n)(X_1 - \mu_1)] & \mathrm{E}[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}$$

Calculus

- Derivatives and Integrals
- Matrix calculus
- Optimization

Derivatives

- A **derivative** of a function defines the slope at a point x.



Integrals

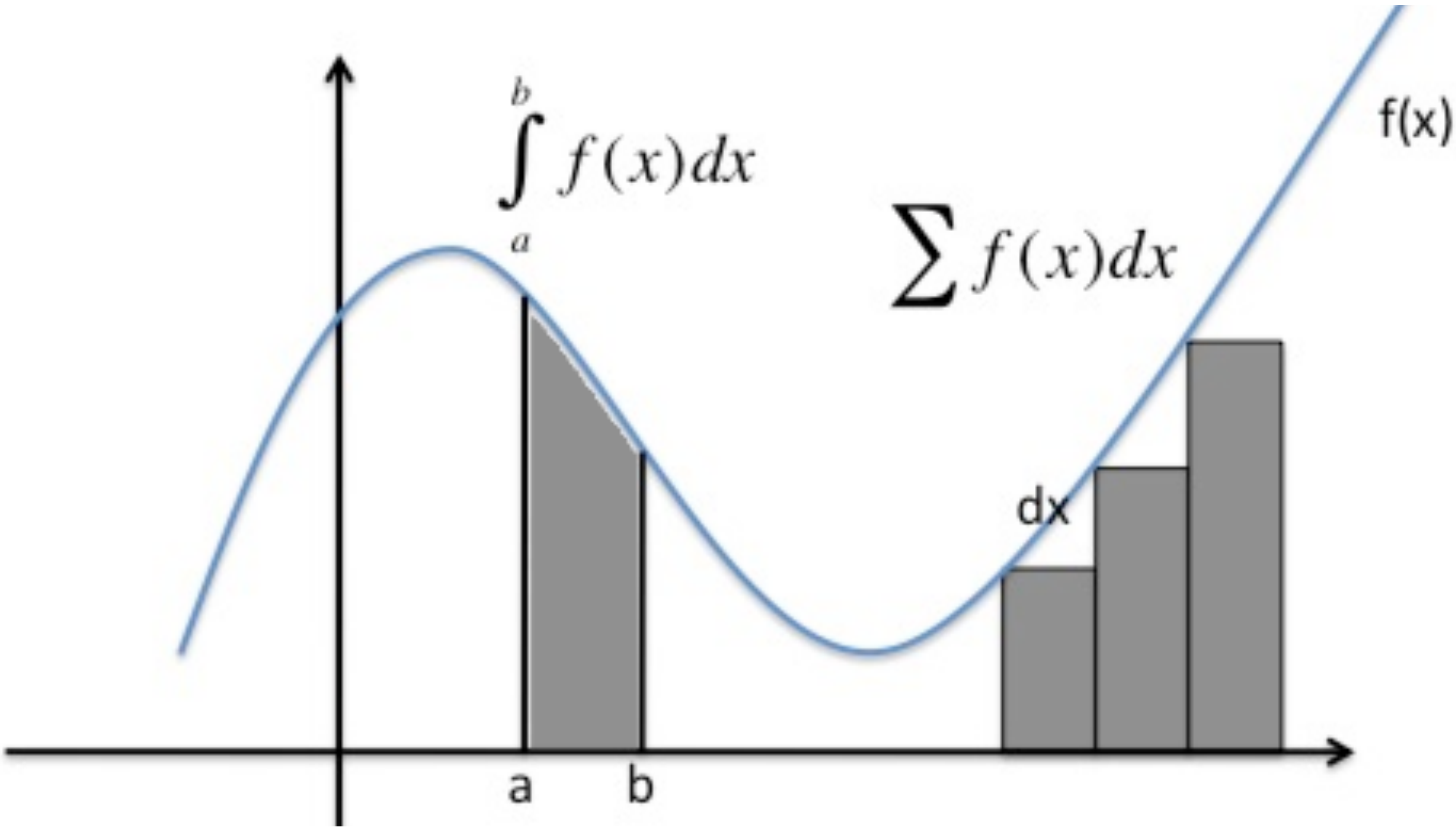
- **Integration** is the inverse operation of derivation (plus a constant)

$$\int f(x)dx = F(x) + c$$

$$F'(x) = f(x)$$

- Graphically, an integral can be considered the area under the curve defined by f(x)

Integration Example



Matrix Calculus

- Derivation with respect to a matrix or vector
- Gradient

Derivative w.r.t. a vector

- Given a vector **x**, and a function f(**x**), how can we find f'(**x**)? Extend to a matrix?

$$f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$$

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_0} \\ \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_{n-1}} \end{pmatrix}$$

Example Derivation

$$f(\vec{x}) = x_0 + 4x_1x_2$$

$$\frac{\partial f(\vec{x})}{\partial x_0} = 1$$

$$\frac{\partial f(\vec{x})}{\partial x_1} = 4x_2$$

$$\frac{\partial f(\vec{x})}{\partial x_2} = 4x_1$$

Example Derivation

$$f(\vec{x}) = x_0 + 4x_1x_2$$

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_0} \\ \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 \\ 4x_2 \\ 4x_1 \end{pmatrix}$$

Also referred to as the **gradient** of a function.

$$\nabla f(x) \text{ or } \nabla f$$

Matrix derivative

- Given two vectors y and x, how can we find y'? Jacobian matrix

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}.$$

Matrix derivative

- Second order derivative (Hessian)

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Useful Vector Calculus identities

- Scalar Multiplication

$$\frac{\partial}{\partial \vec{x}}(\vec{x}^T \vec{a}) = \frac{\partial}{\partial \vec{x}}(\vec{a}^T \vec{x}) = \vec{a}$$

$$\frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B} + \mathbf{B}^T)\mathbf{x}$$

- Product Rule

$$\frac{\partial}{\partial x}(AB) = \frac{\partial A}{\partial x} B + A \frac{\partial B}{\partial x}$$

$$\frac{\partial}{\partial x}(x^T A) = A$$

$$\frac{\partial}{\partial x}(Ax) = A^T$$

Best References

- Comprehensive reference
  - https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf
- G. Strang: linear algebra
  - textbook, video lectures

Optimization

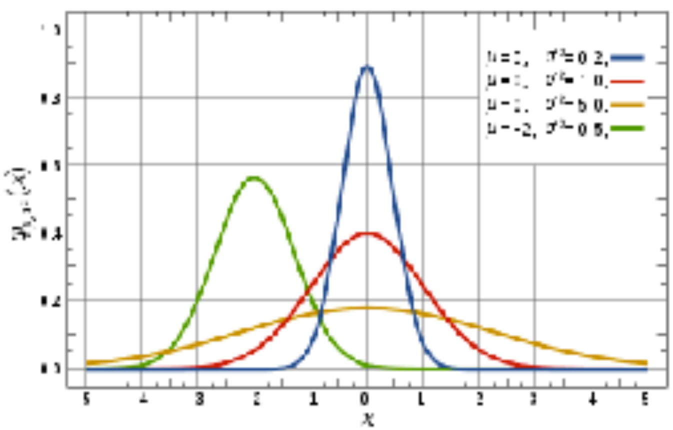
- Have an objective function that we’d like to maximize or minimize, f(x)
- Set the first derivative to zero.

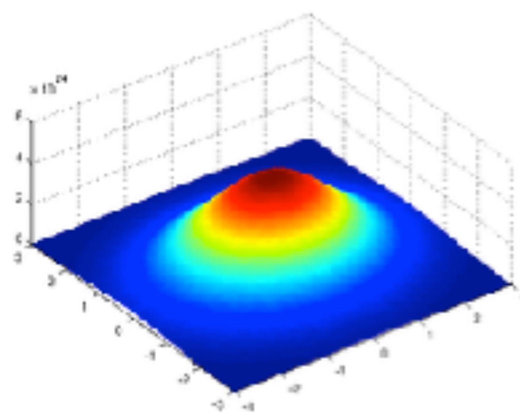
Exercise

- Finding the mode: highest probability point

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

$$N(x|\mu,\Sigma)=\frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$





Exercise

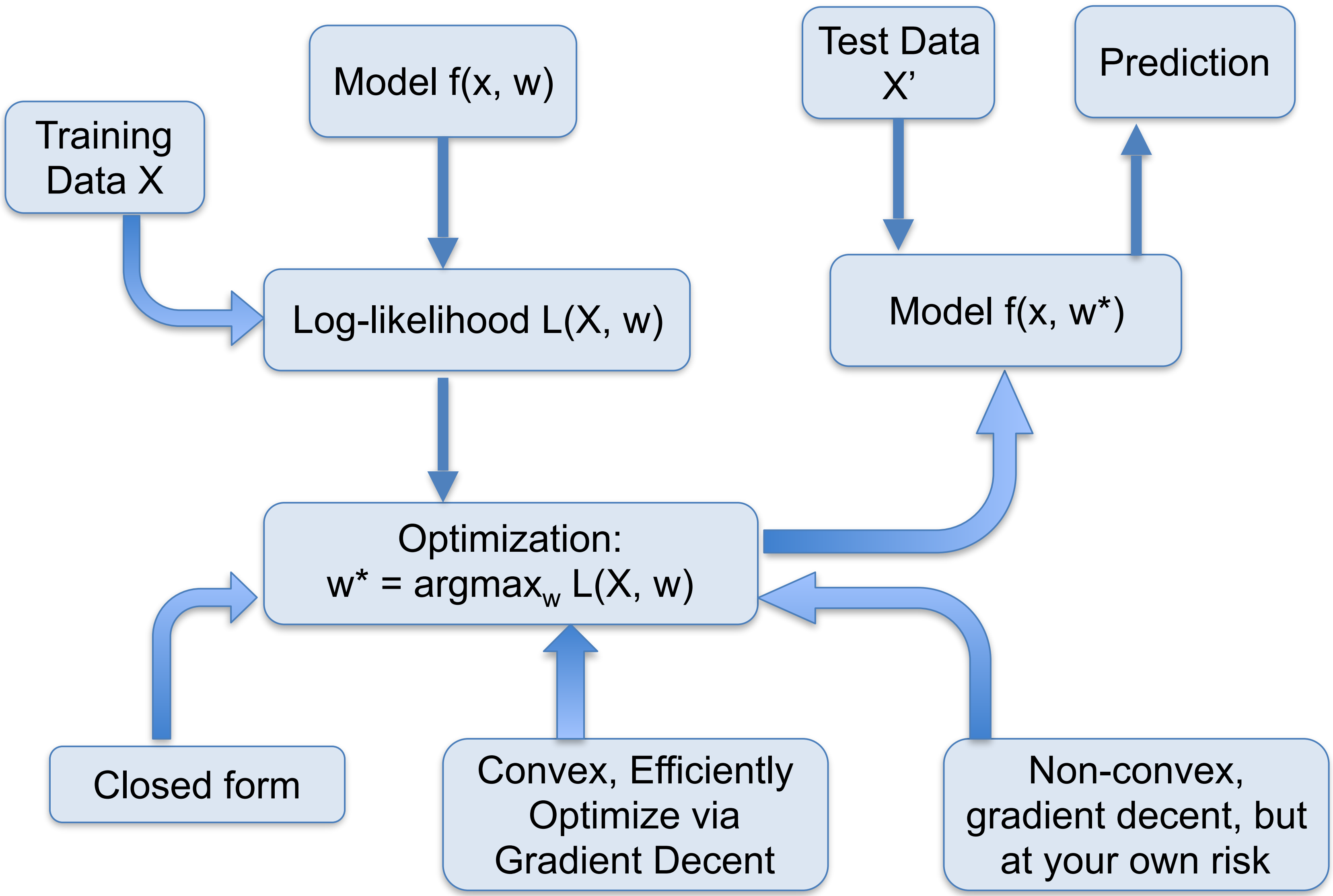
- Given observations, estimate the model (parameters)
- Maximum Likelihood Estimation (MLE)
  - Likelihood: $p\left(\mathbf{x}|\mu, \sigma^2\right)$
  - Find parameters maximizing the (log) likelihood
  - Partial derivative = 0

$$\ln p\left(\mathbf{x}|\mu, \sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2$$

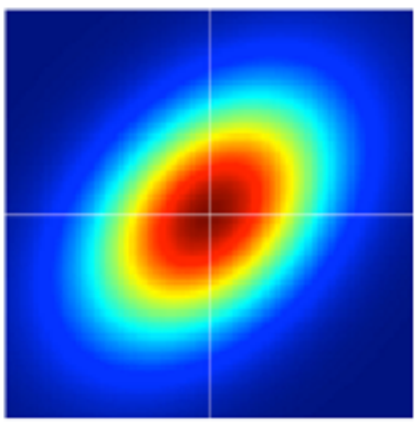
- High dim?

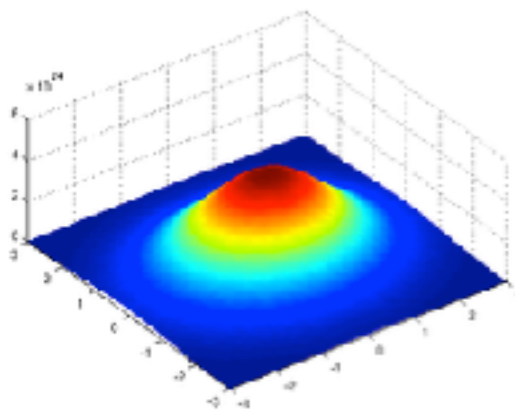
Machine Learning Pipeline

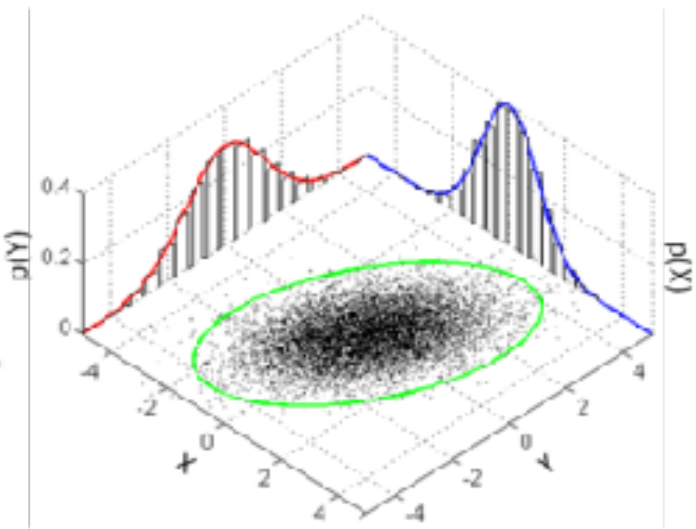


High Dimension

- Solve p(x) = sigma or any constant
  - Ellipse: principle axes = eigenvectors of
  - Marginal $p(x_i)$ – Gaussian
  - Conditional $p(x_1|x_2)$ – also a Gaussian







$$N(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

Optimization with constraints

- What if I want to constrain the parameters of the model.
  - The mean is less than 10

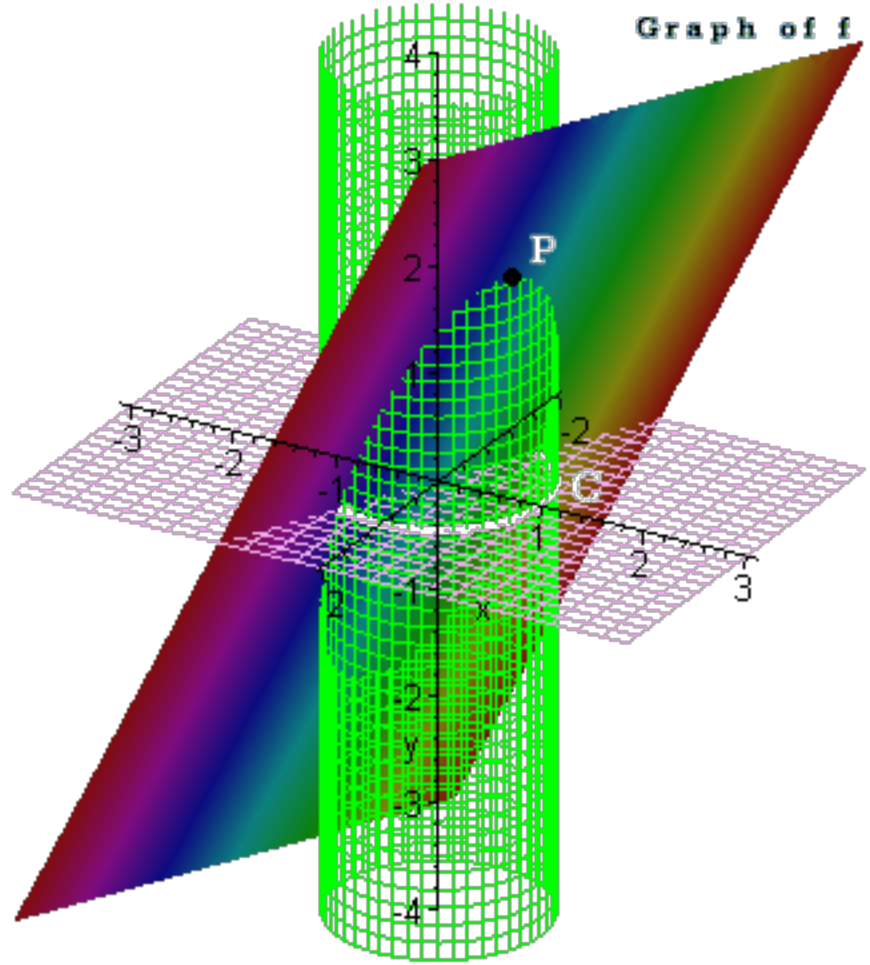
- Find the best likelihood, subject to a constraint.
- Two functions:
  - An objective function to maximize
  - An inequality that must be satisfied

Lagrange Multipliers

- Find maxima of f(x,y) subject to a constraint.

$$f(x, y) = x + 2y$$

$$x^2 + y^2 = 1$$



General form

- Maximizing: $f(x, y)$
- Subject to: $g(x, y) = c$

- Introduce a new variable, and find a maxima.

$$\Lambda(x, y, \lambda) = f(x, y) + \lambda(g(x, y) - c)$$

Example

- Maximizing: $f(x, y) = x + 2y$
- Subject to: $x^2 + y^2 = 1$

Why does Machine Learning need these tools?

- Calculus
  - We need to identify the maximum likelihood, or minimum risk. Optimization
  - Integration allows the marginalization of continuous probability density functions
- Linear Algebra
  - Many features leads to high dimensional spaces
  - Vectors and matrices allow us to compactly describe and manipulate high dimension al feature spaces.

Why does Machine Learning need these tools?

- Vector Calculus
  - All of the optimization needs to be performed in high dimensional spaces
  - Optimization of multiple variables simultaneously – Gradient Descent
  - Want to take a marginal over high dimensional distributions like Gaussians.

Next Class

- Linear Regression

To Do

- Review “Bishop”: Ch 1 & 2, Slides L3, L4
- Read “Bishop”: Ch 3.
- Homework 1.