Clustering & Expectation-Maximization

STAT261: Introduction to Machine Learning

Prof. Allie Fletcher

Outline

- Clustering
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM
- Numerical Methods

K-means

- A simple iterative algorithm to determine:
  - μ_k = mean of each cluster (hence, the name K-means)
  - $C_n \in \{1, \ldots K\}$ = cluster that data point x_n belongs to
- Step 0: Start with guess at cluster centroids: μ_k
  - Random fine, but often "smart" heuristics: distance, etc
- Step 2: Update cluster membership:
$$C_n = \arg\min_k \|x_n - \mu_k\|^2$$
- Step 1: Update mean of each cluster:
$\mu_k =$ average of x_n s.t. $xn\ such\ C_n = i$
- Step 2: Update cluster membership:
$$\sigma_n = \arg\min_i \|x_n - \mu_i\|^2$$
  - Selects cluster with closest mean
- Return to step 1

Outline

- Factor Analysis
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Probabilistic Mixture Model for Clusters

- Random variable $z \in \{1, \ldots, K\}$
  - Discrete event with PMF: $P(z = i)$
  - Latent variable: often not directly observed
- Observed variable x, can be continuous
  - Probability depends on z, $p(x|z = i)$
  - One PDF, or component per state $z = i$
- Distribution of x: computed via total probability
  - PDF $p(x) = \sum p(x|z = i) P(z = i)$
  - CDF $F(x_0) = \sum P(x \leq x_0 | z = i) P(z = i)$
- Example: Mixture of two Gaussians

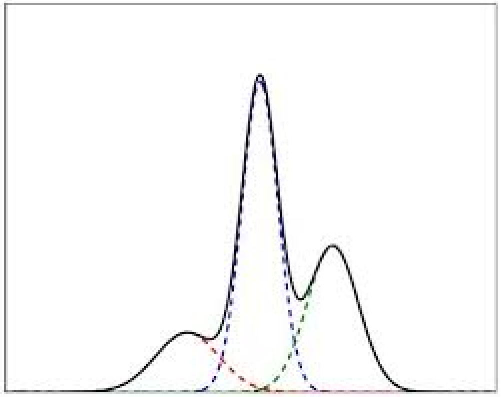
Mixture Models: Examples

- Many data occurs from underlying discrete states
- Example 1: Size of a webpage
  - $z =$ content of the webpage, e.g. number of images
- Example 2: Speech
  - $z =$ phoneme the speaker is saying
- Example 3: Image
  - $x =$ RGB values of a pixel or region of pixels
  - $z =$ one a small number of objects the pixel is part of

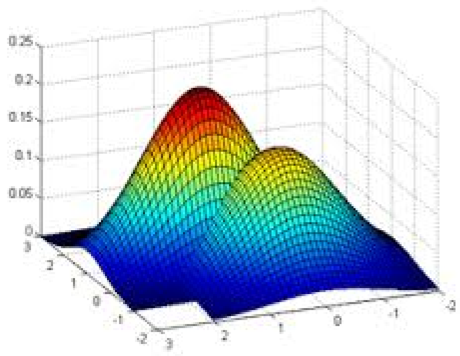
Gaussian Mixture Models

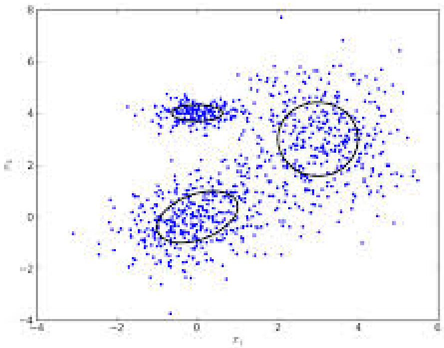
- Each $p(x|z = i)$ is a Gaussian
- Parametrized by:
  - $q_i = P(z = i)$ = Probability of each component
  - $\mu_i = E(x|z = i), P_i = var(x|z = i)$
    mean and variance in each component
- Can be vector valued

Visualizing GMMs



- 1d model with $K = 3$ components





- PDF for 2d GMM with $K = 2$ components
- Random points from a GMM with $K = 3$ components

UCLA

Expectation and Variance

- Can compute expectation and variance by total probability
  - Expectation: $\mu = E(x) = \sum q_i \mu_i$
  - Variance:

$$var(x) = \sum_i q_i P_i \quad + q_i(\mu_i - \mu)(\mu_i - \mu)^T$$

Variance within component

Variance between components

- Proof on board

Estimating the Latent Variable

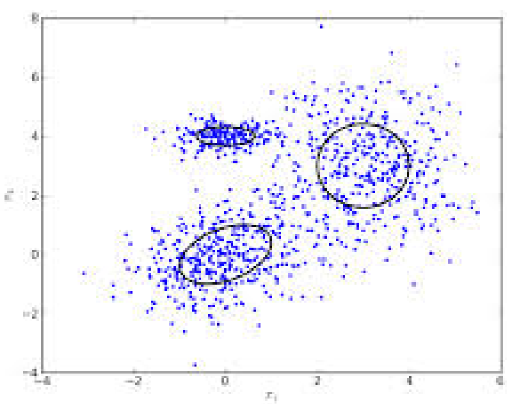
- Given x, can we estimate z if we knew parameters:
- Use Bayes' rule:

$$P(z = i|x) = \frac{P(x|z = i)q_i}{\sum_k P(x|z = k)q_k}$$

- Example: Scalar Gaussian
  - Illustration on board

Fitting a Mixture Model

- Given data $x = (x_1, \ldots, x_N)$
- Find GMM parameters
  - Mean and variance in each component
  - Probability of each component
- Can be interpreted as "clustering"
- Parametric probabilistic model versus K-means



Maximum Likelihood Estimation

- Unknown parameters in GMM:

$$\theta = (q_1, \dots, q_K, \mu_1, \dots, \mu_K, P_1, \dots, P_K)$$

- Data $x = (x_1, \dots, x_N)$
- Likelihood of x_n:

$$p(x_n|\theta) = \sum_{k=1}^{K} p(x_n|z_n = k, \theta) P(z_n = k|\theta) = \sum_{k=1}^{K} q_k N(x_n|\mu_k, P_k)$$

- Negative log likelihood of all data

$$L(\theta) = -\ln p(x|\theta) = -\sum_{n=1}^{N} \ln \left[\sum_{i=1}^{K} q_i N(x_n|\mu_i, P_i) \right]$$

- ML estimation:

$$\hat{\theta} = \arg\min_{\theta} L(\theta)$$

- Type equation here.

Outline

- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Expectation Maximization Algorithm

- Optimization of $L(\theta)$ is hard
  - No simple way to directly optimize
  - Likelihood is non-convex

- Expectation maximization:
  - Simple iterative procedure:
  - Generates a sequence of estimates $\hat{\theta}^0, \hat{\theta}^1, \ldots$
  - Attempts to approach MLE

$$\hat{\theta}^k \to \arg\min_{\theta} L(\theta)$$

f

EM Steps

- E-step: Estimate the latent variables
  - Find the posterior of the latent variables given $\hat{\theta}^k$

$$P(z|x, \theta = \hat{\theta}^k)$$

  - Compute function, Q, auxiliary function

$$Q(\theta, \hat{\theta}^k) := E\left[\ln p(x, z|\theta) | \hat{\theta}^k\right]$$
$$= \sum_z \ln p(x, z|\theta) \, P(z|x, \theta = \hat{\theta}^k)$$

- M-step: Update parameters

$$\hat{\theta}^{k+1} = \arg\max_\theta Q(\theta, \hat{\theta}^k)$$

E-Step for a GMM: Finding the posterior

- Given parameters q_i, μ_i, P_i
- Find posterior by Bayes rule

$$\gamma_{ni} = P(z_n = i|x) = \frac{P(x_n|z_n = i)q_i}{\sum_k P(x_n|z_n = k)q_k}$$
$$= \frac{N(x_n|\mu_i, P_i)q_i}{\sum_k P(x_n|\mu_k, P_k)q_k}$$

- A “soft” selection

E-Step for a GMM

- Auxilliary function separates

$$Q\left(\theta, \hat{\theta}^k\right) = E\left[\ln p(x, z) | \hat{\theta}^k\right]$$

$$= \sum_{i=1}^{K} \sum_{n=1}^{N} \gamma_{ni} \ln P(x_n, z_n = i)$$

$$= \sum_{i=1}^{K} \sum_{n=1}^{N} \gamma_{ni} [\ln q_i + \ln N(x_n | \mu_i, P_i)]$$

M-Step for the GMM

- Maximize $Q\left(\theta, \hat{\theta}^k\right)$
- Update for q_i (proof on board)

$$q_i = \frac{N_i}{\sum_j N_j}\,, \qquad N_i = \sum_n \gamma_{ni}$$

- Update for μ_i

$$\mu_i = \frac{1}{N_i} \sum_n \gamma_{ni}\, x_n$$

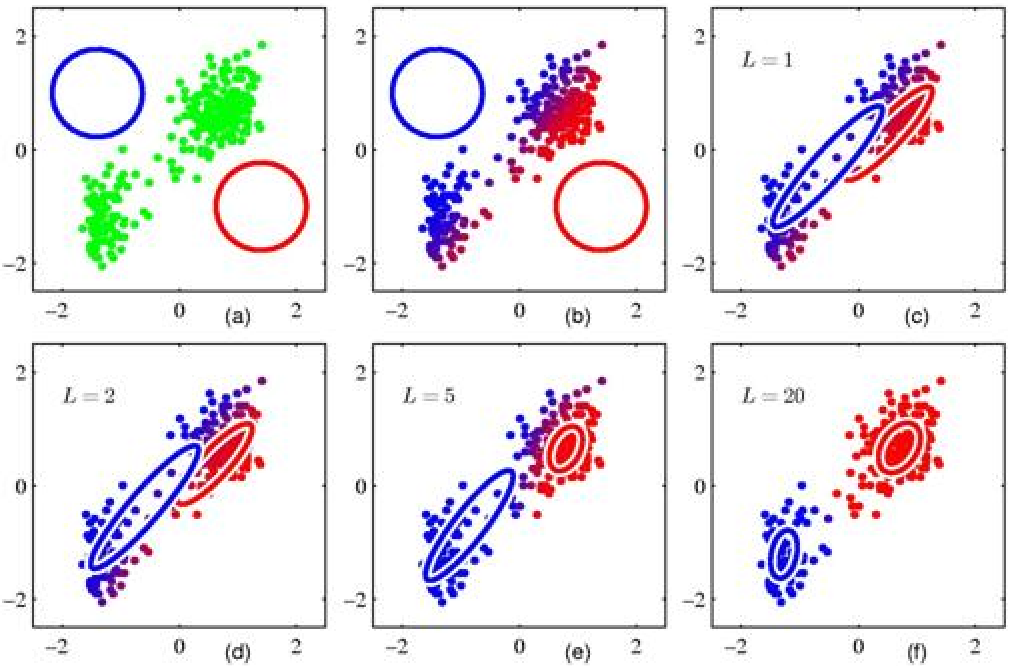
- Update for P_i

$$P_i = \frac{1}{N_i} \sum_n \gamma_{ni}\, (x_n - \mu_i)(x_n - \mu_i) \hat{}\, T$$

Relation to K means

- EM can be seen as a "soft" version
  - In K-Means: $\gamma_{ni} = 1$ or 0
- Variance
  - In K-means: $P_i = I$
  - In EM, this is estimated
- EM provides "scaling" of various dimensions

EM Illustrated



- Simple example with K=2 clusters
- Dimension = 2
- Convergence from a bad initial condition

Outline

- LDA - multiple
- Clustering
  - K-means
  - Hierarchical Clustering
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Majorization Minimization

- Suppose we wish to minimize $f(\theta)$
- MM algorithm: find a majorizing function $F(\theta, \theta^k)$:
  - $f(\theta^k) = F(\theta^k, \theta^k)$
  - $f(\theta) \leq F(\theta, \theta^k)$ for all θ
- Take $\theta^{k+1} = \arg\min_{\theta} F(\theta, \theta^k)$ (minimize majorization)
- Theorem: $f(\theta^{k+1}) \leq f(\theta^k)$
- Proof:

$$f(\theta^{k+1}) \leq F(\theta^{k+1}, \theta^k) \leq F(\theta^{k+1}, \theta^k) \leq f(\theta^k)$$

Gradient Descent as a MM

- Find $\alpha \geq f''(\theta)$
- Define

$$F(\theta, \theta^k) = f(\theta^k) + \nabla f(\theta^k)(\theta - \theta^k) + \frac{\alpha}{2}\|\theta - \theta^k\|^2$$

- By Taylor's theorem, this is a majorizing function
- Gradient descent:

$$\theta^{k+1} = \arg\min_{\theta} F(\theta, \theta^k) = \theta^k - \frac{1}{\alpha}\nabla f(\theta^k)$$

Convergence

- $p(z|x,\theta) = p(x,z|\theta)/p(x|\theta)$
- $J(\theta) = \ln p(x|\theta) = \ln p(x,z|\theta) - \ln p(z|x,\theta)$
- $J(\theta) = E[\ln p(x,z|\theta)\,|\theta^k] - E[\ln p(z|x,\theta)\,|\theta^k]$
$$= Q(\theta,\theta^k) + H(\theta,\theta^k)$$
- EM algorithm: $J(\theta^{k+1}) \geq J(\theta^k)$
  - Proof on board
- Algorithm may get stuck in local maxima

Outline

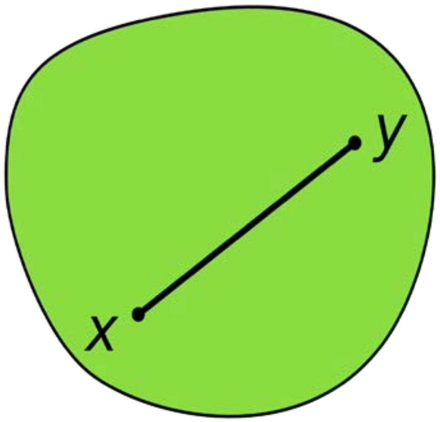
- Convex sets and functions
- Gradient descent
- Newton's method
- Using MATLAB for optimization
- Other topics

Convex Sets

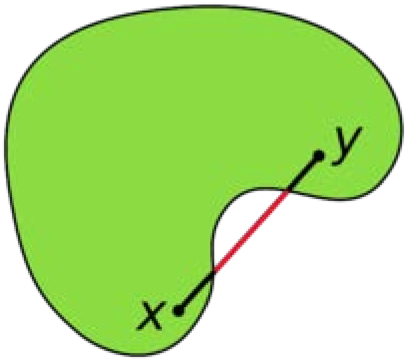
- Definition: A set X is convex if for any $x, y \in X$,
$$tx + (1-t)y \in X \text{ for all } t \in [0,1]$$
- Any line between two points remains in the set.
- Examples:
  - Square, circle, ellipse
  - $\{x \mid Ax \leq b\}$ for any matrix A and vector b
  - Not a start
- Will draw pictures on board

Convex Set Visualized

- Convex

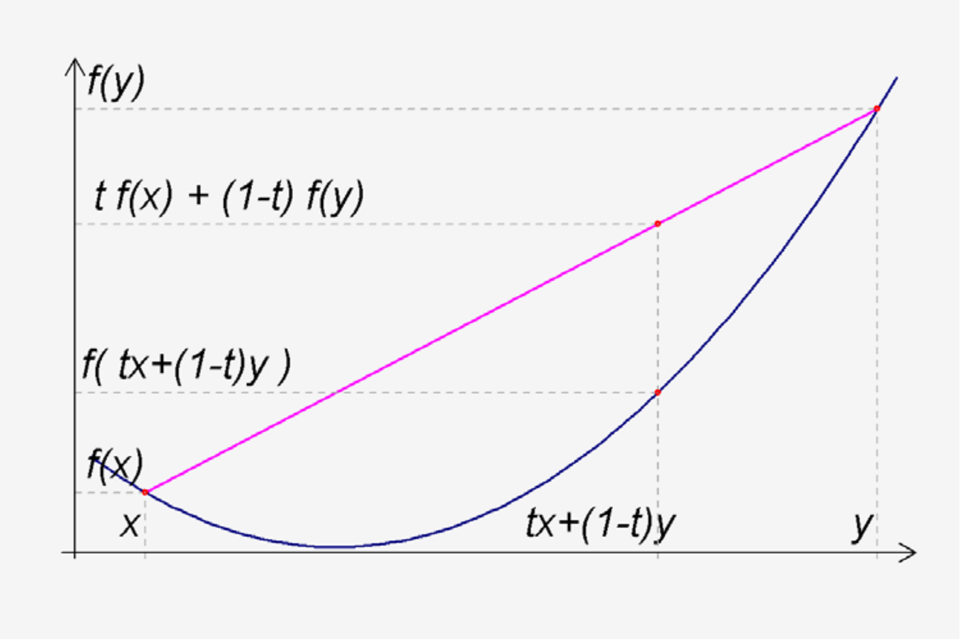


- Not convex



Convex Functions

- A real-valued function $f(x)$ is convex if:
  - Its domain is a convex set, and
  - For all x, y and $t \in [0,1]$:
    $$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$



Convex Function Examples

- Linear function of a scalar $f(x) = ax + b$
- Linear function of a vector $f(x) = a^T x + b$
- Quadratic $f(x) = \frac{1}{2}ax^2 + bx + c$ is convex iff $a \geq 0$
- If $f''(x)$ exists everywhere, $f(x)$ is convex iff $f''(x) \geq 0$.
  - When x is a vector $f''(x) \geq 0$ means the Hessian must be positive semidefinite
- $f(x) = e^x$
- If $f(x)$ is convex, so is $f(Ax + b)$

Properties

- If $f(x)$ is convex, it is continuous
- If $f(x)$ has a derivative, then

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x)$$

Outline

- Convex sets and functions
- Gradient descent
- Newton's method
- Using MATLAB for optimization
- Other topics

Unconstrained optimization

- Problem: Given $f(x)$ find the minimum:

$$x^* = \arg\min_x f(x)$$

  - $f(x)$ is called the objective function
  - $x = (x_1, \cdots, x_p)$ is a vector of decision variables or parameters

- Called unconstrained since there are no constraints on x
- Will discuss constrained optimization briefly later

Numerical Optimization

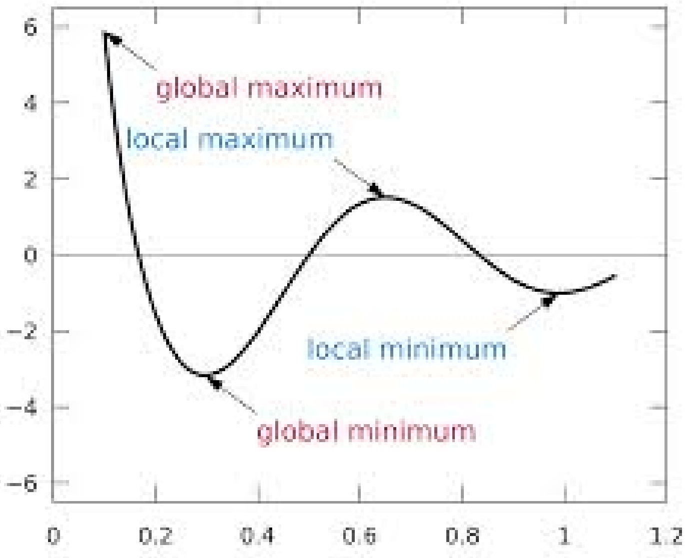
- We saw that we can find minima by setting $\nabla f(x) = 0$
  - p equations and p unknowns.
  - May not have closed-form solution

- Numerical methods: Finds a sequence of estimates x^k

$$x^k \to x^*$$

  - Or converges to some other "good" minima
  - Run on a computer program, like MATLAB

Local vs. Global Minima



- Definitions:
  - x^* is a global minima if $f(x) \geq f(x^*)$ for all x
  - x^* is a local minima if $f(x) \geq f(x^*)$ for all x in some open neighborhood of x^*
- Most numerical methods only guarantee convergence to local minima

Local Minima and Convex Function

- Theorem: If $f(x)$ is convex and x^* is a local minima, then it is a global minima
- Also, if $f(x)$ is strictly convex, then the global minima is unique

- Implication: If $f(x)$ is convex, a numerical method that converges to a local minima, will converge to a global minima.
- Many methods can find local minima
- For convex objectives, these methods will find global minima

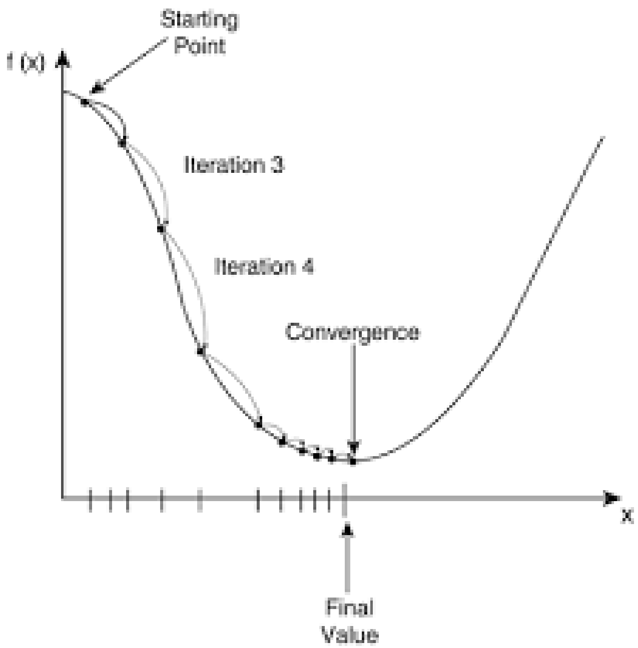
Gradient Descent

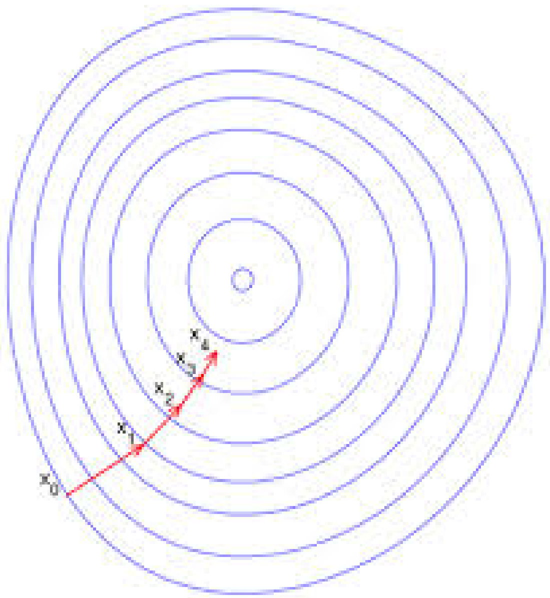
- Most simple method for unconstrained optimization
- Recall gradient:

$$\nabla f(x) = \left(\partial f(x)/\partial x_1, \cdots, \partial f(x)/\partial x_p\right)^T$$

  - Column vector
- Gradient descent algorithm:
  - Start with initial x^0
  - $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$
  - Repeat until some stopping criteria
- α_k is called the step size

Gradient Descent Illustrated





- $p = 1$

- $p = 2$

Gradient Descent Analysis

- Using gradient update rule

$$f(x^{k+1})$$
$$= f(x^k) + \nabla f(x^k) \cdot (x^{k+1} - x^k) + O\|x^{k+1} - x^k\|^2$$
$$= f(x^k) \; - \alpha \, \nabla f(x^k) \cdot (x^{k+1} - x^k) + O(\alpha^2)$$

- Consequence: If step size α is small, then $f(x^k)$ decreases

- Theorem: If $f''(x)$ is bounded above, $f(x)$ is bounded below, and α is chosen sufficiently small, then gradient descent converges to local minima

Step Size Selection

- Theorem shows we can always converge to a local minima
  - Global minima if $f(x)$ is convex
- But, step size selection is problematic
  - Need to know $f''(x)$ to find maximum step size
  - Practical choice tends to be conservative
- Very slow step size, many steps to convergence

Adaptive Step Size Selection

- Practical algorithms change step size adaptively

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k)$$

- Tradeoff: Selecting large α_k:
  - Larger steps, faster convergence
  - But, may overshoot

Armijo Rule

- Recall that we know if $x^{k+1} = x^k - \alpha \nabla f(x^k)$

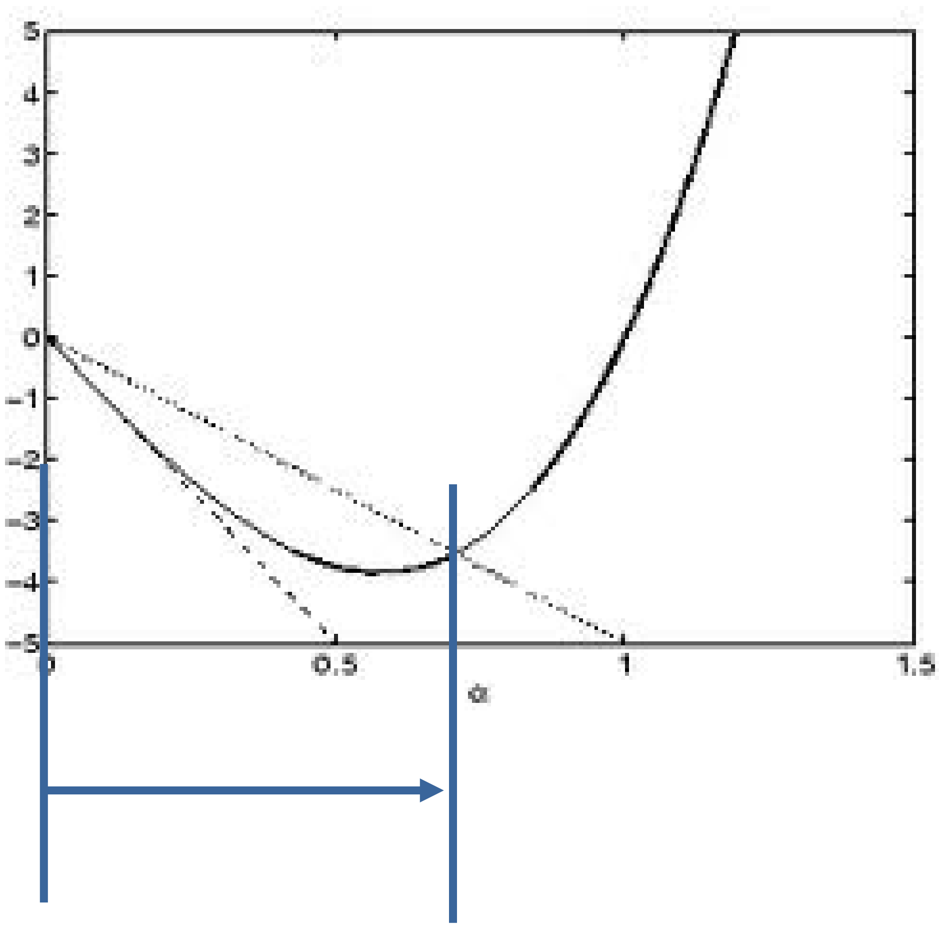
$$f(x^{k+1}) = f(x^k) - \alpha \left\|\nabla f(x^k)\right\|^2 + O(\alpha^2)$$

- Armijo Rule:
  - Select some $c \in (0,1)$. Usually $c = 1/2$
  - Select α such that

$$f(x^{k+1}) \le f(x^k) - c\alpha \left\|\nabla f(x^k)\right\|^2$$

  - Decreases by at least at fraction c predicted by linear approx.
- Step size α selected by a line search to find largest α satisfying above conditions

Armijo Rule Illustrated



Feasible region for x^{k+1}

- Armijo rule: $f\left(x^{k+1}\right) \leq f\left(x^k\right) - c\alpha \left\|\nabla f\left(x^k\right)\right\|^2$
- Guarantees function decrements in each iteration
- No overshoot