Convex Optimization Part II

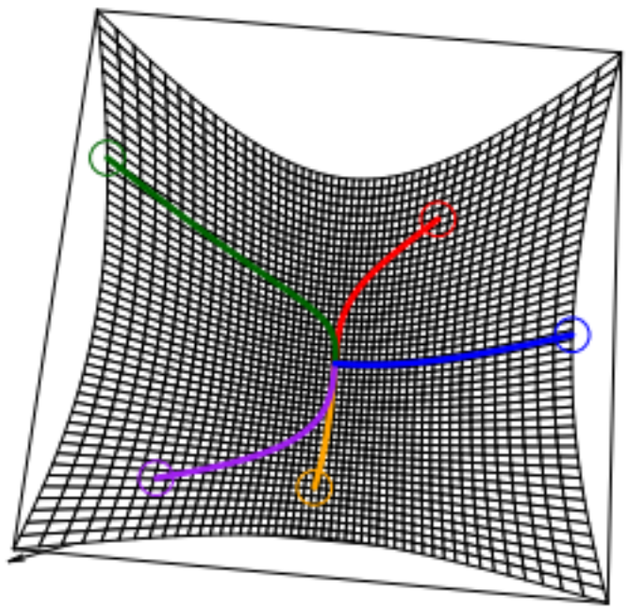
CS 584: Big Data Analytics

Material adapted from
John Duchi (https://www.cs.berkeley.edu/~jordan/courses/294-fall09/lectures/optimization/slides.pdf)
& Stephen Boyd (https://web.stanford.edu/class/ee364a)

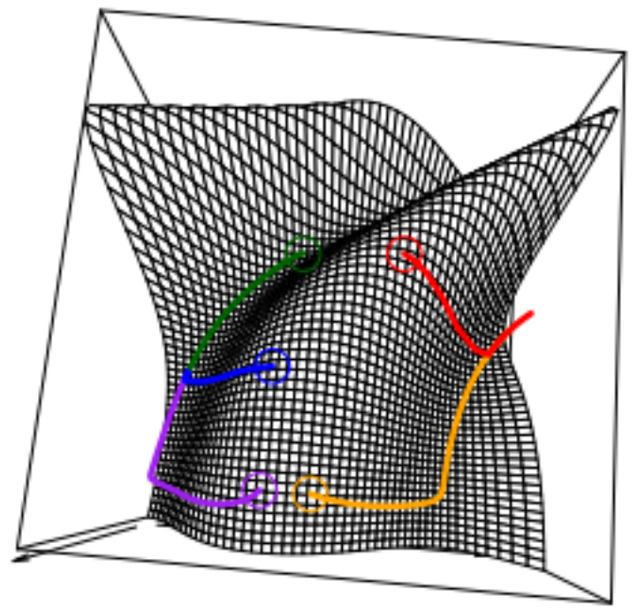
Convex Optimization Problems

$$\begin{aligned} \min_{x} \quad & f_0(x) && \text{convex function} \\ \text{s.t.} \quad & f_k(x) \leq 0, k = 1, 2, \cdots, K && \text{convex sets} \\ & h_j(x) = 0, j = 1, 2, \cdots, J && \text{affine constraints} \end{aligned}$$

Local minima are global minima



Convex



Nonconvex

Why Convex Optimization?

- Achieves global minimum, no local traps
- Highly efficient software available
- Can be solved by polynomial time complexity algorithms
- Dividing line between “easy” and “difficult” problems

Review: Regularized Regression

- Linear regression has low bias but suffers from high variance (maybe sacrifice some bias for lower variance)

- Large number of predictors makes it difficult to identify the important variables

- Regularization term imposes penalty on “less desirable solutions”

  - Ridge regression: reduces the variance by shrinking coefficients towards zero by using the squared ℓ_2 penalty

  - LASSO: feature selection by setting coefficients to zero using an ℓ_1 penalty.

Convex Optimization: LASSO

- LASSO: least absolute shrinkage and selection operator
- Coefficients are the solutions to the ℓ_1 optimization problem
- Penalize regression coefficients by shrinking many of them to 0

$$\min_{\beta \in \mathbb{R}^d} \; ||y - X\beta||_2^2$$

$$\text{subject to } ||\beta||_1 \leq s$$

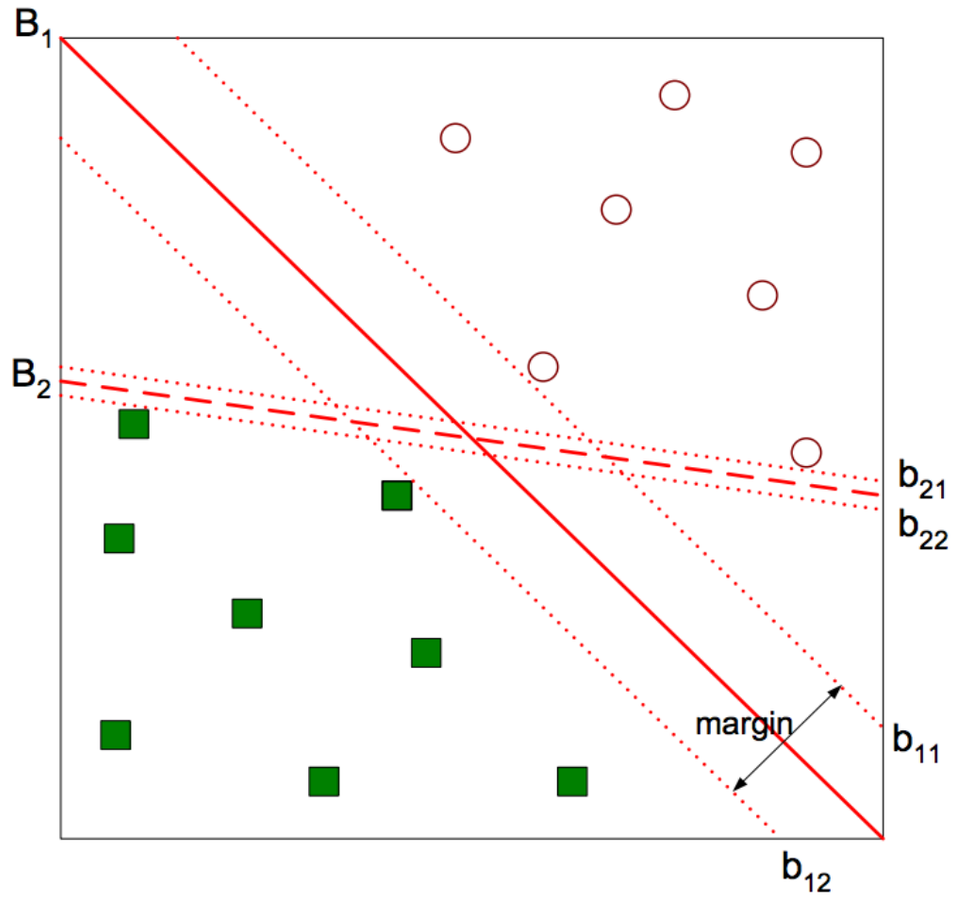
Review: Support Vector Machines (SVM)

- A leading edge classier which uses “optimal” hyperplane in a suitable feature space for classification

- Finds the hyperplane that maximizes the margin (i.e., B1 is better than B2)

- Points closest to separating hyperplane are known as support vectors

- Kernel trick to transform space into higher-dimensional feature space where separable



Convex Optimization: SVM

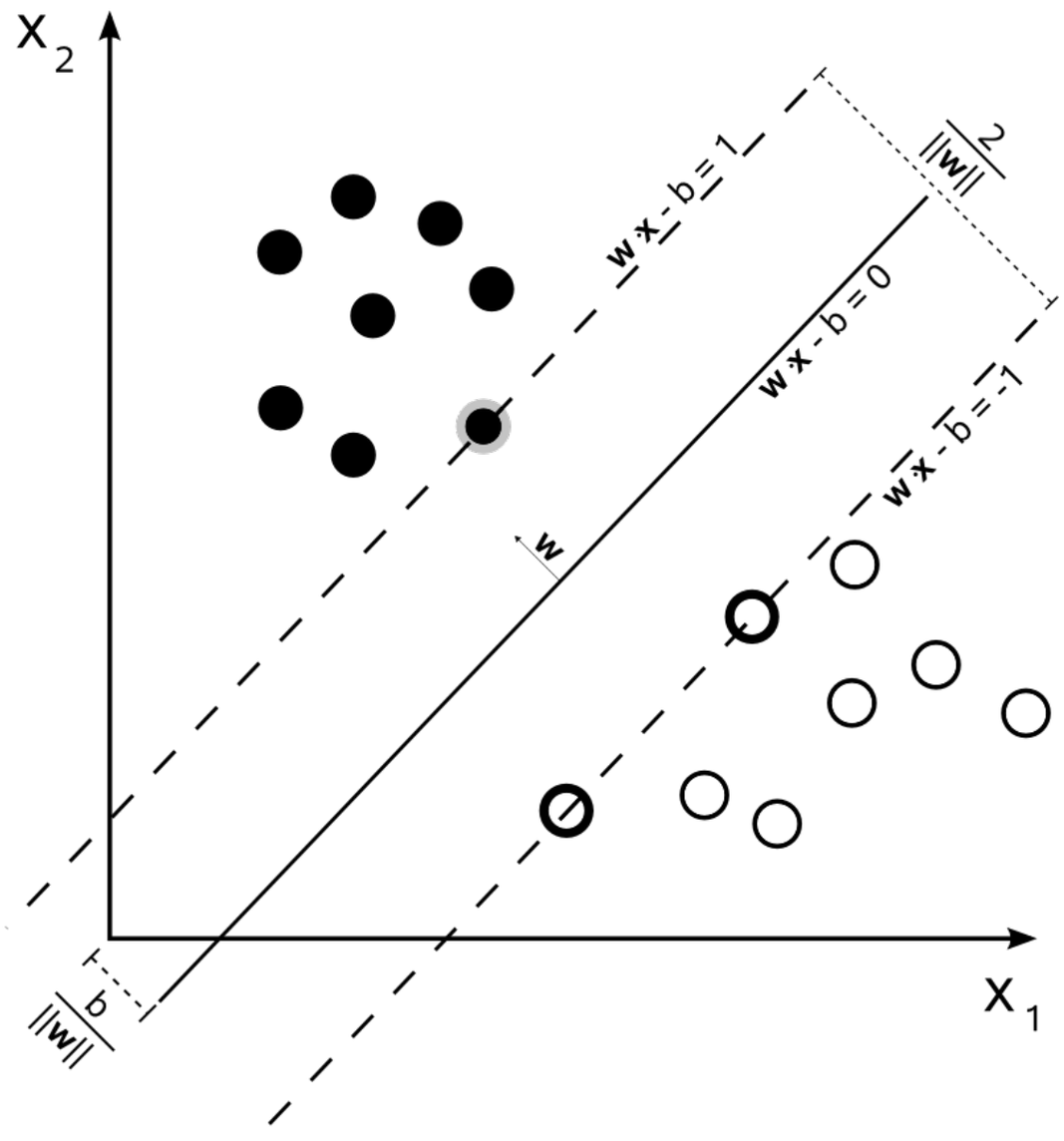
- Learning SVM is formulated as an optimization problem

- Quadratic optimization problem subject to linear constraints and there is a unique minimum

$$\min_{w} \ ||w||^2 + C \sum_i \xi_i$$

$$\text{s.t. } \xi_i \geq 1 - y_i x_i^\top w$$

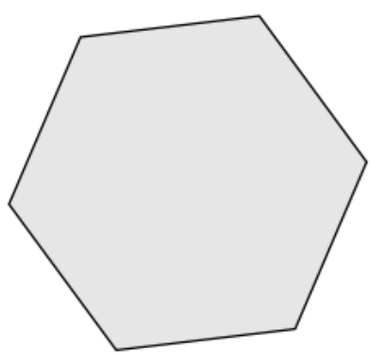
$$\xi_i \geq 0$$



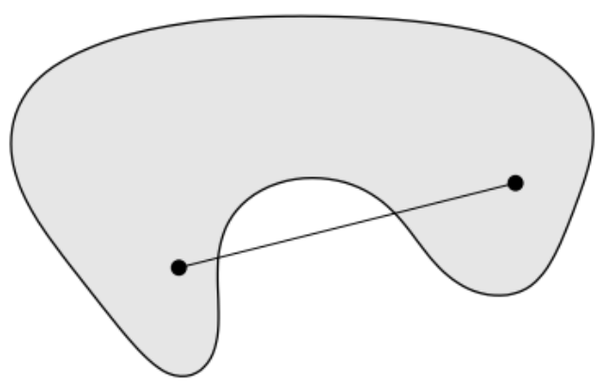
Convex Sets

$$x_1, x_2 \in C, 0 \leq \theta \leq 1 \implies \theta x_1 + (1-\theta)x_2 \in C$$

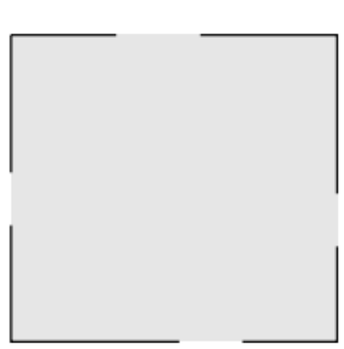
Any line segment joining any two elements lies entirely in set



convex



non-convex



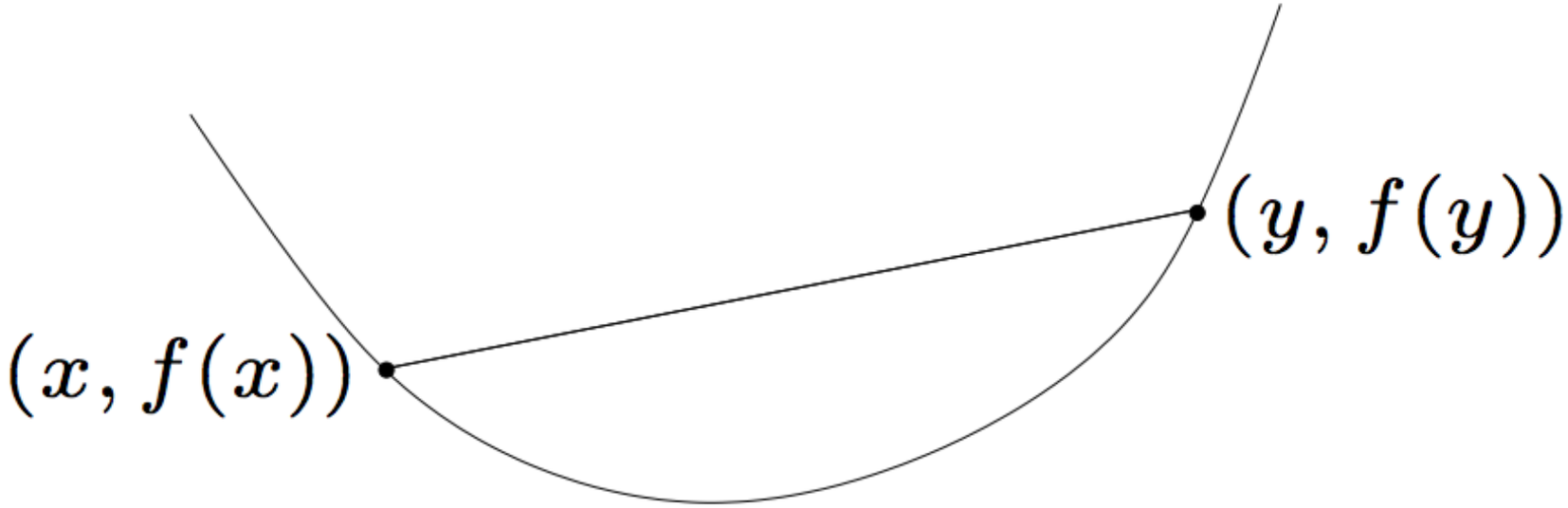
non-convex

Convex Function

$f : \mathbb{R}^n \to \mathbb{R}$ is convex if **dom** f is a convex set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

for all $x, y \in$ **dom** $f, 0 \leq \theta \leq 1$



f lies below the line segment joining f(x), f(y)

Properties of Convex Functions

- Convexity over all lines

$$f(x) \text{ is convex} \implies f(x_0 + th) \text{ is convex in } t \text{ for all } x_0, h$$

- Positive multiple

$$f(x) \text{ is convex} \implies \alpha f(x) \text{ is convex for all } \alpha \geq 0$$

- Sum of convex functions

$$f_1(x), f_2(x) \text{ convex} \implies f_1(x) + f_2(x) \text{ is convex}$$

- Pointwise maximum

$$f_1(x), f_2(x) \text{ convex} \implies \max\{f_1(x), f_2(x)\} \text{ is convex}$$

- Affine transformation of domain

$$f(x) \text{ is convex} \implies f(Ax + b) \text{ is convex}$$

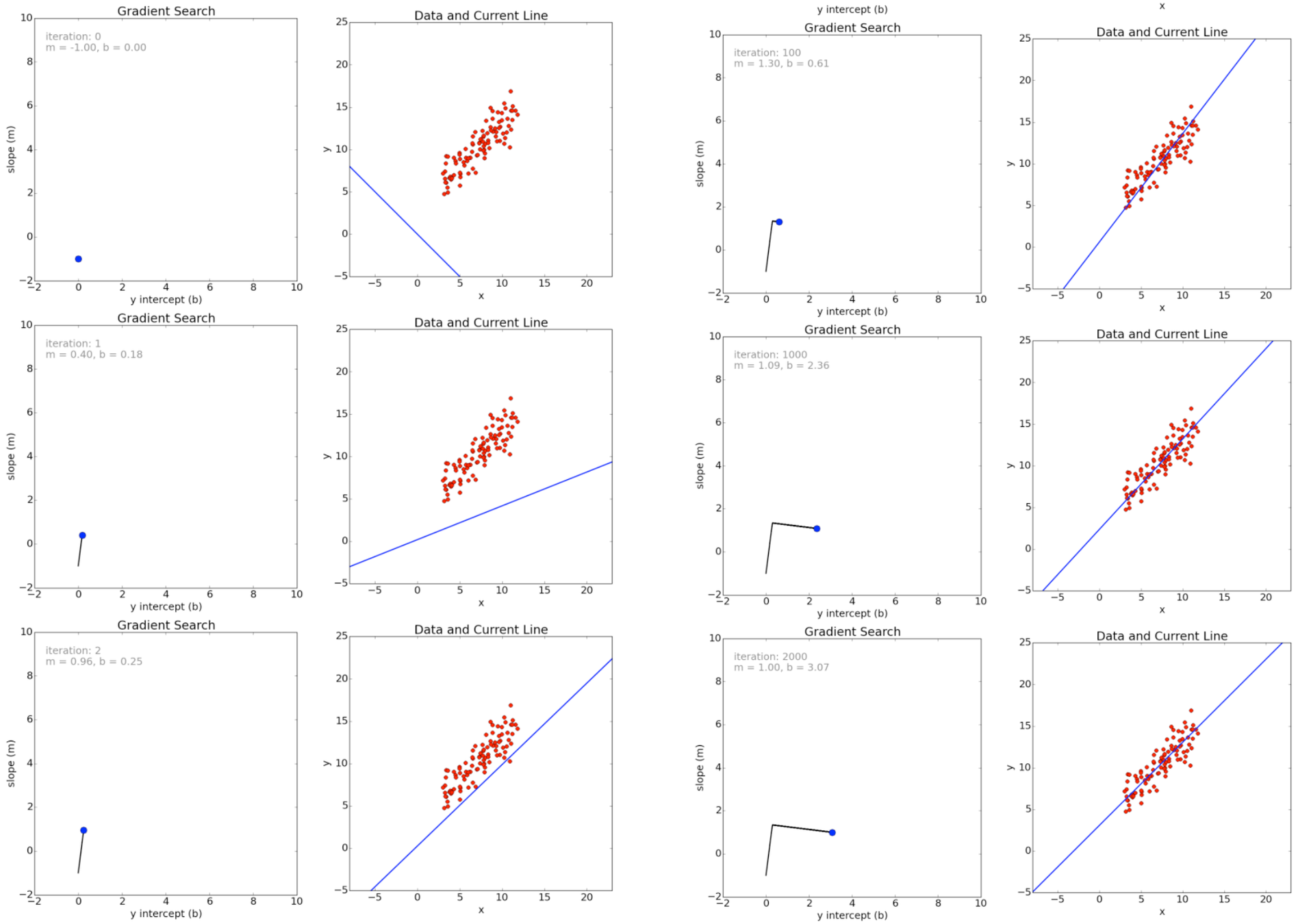
Gradient Descent (Steepest Descent)

- Consider unconstrained, smooth convex optimization problem (i.e., f is convex and differentiable)

- At each iteration, take a small step in the steepest descent direction

- Very simple to use and implement and is a numerical solution to the problem

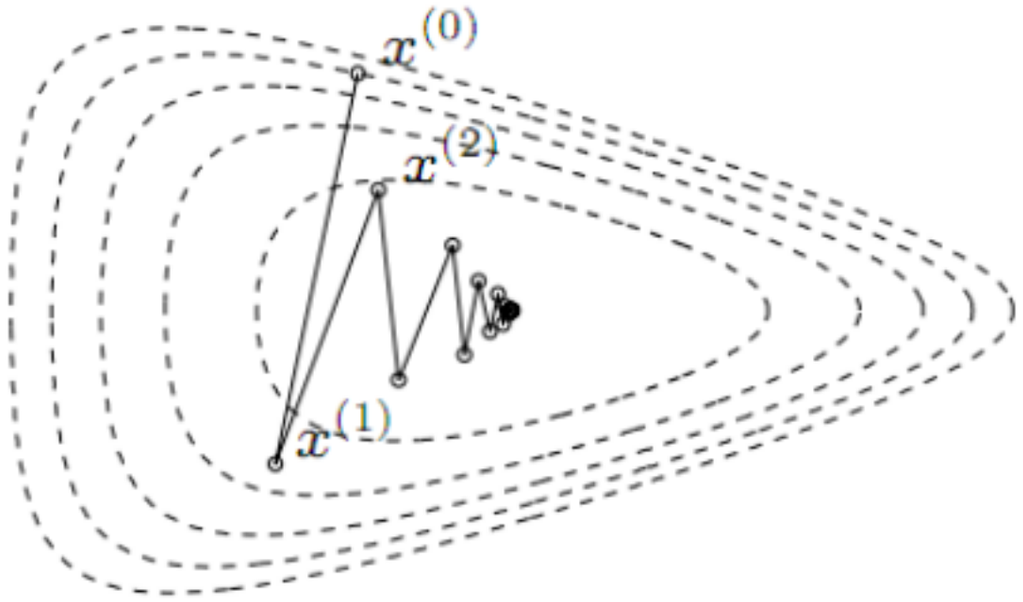
Gradient Descent: Linear Regression



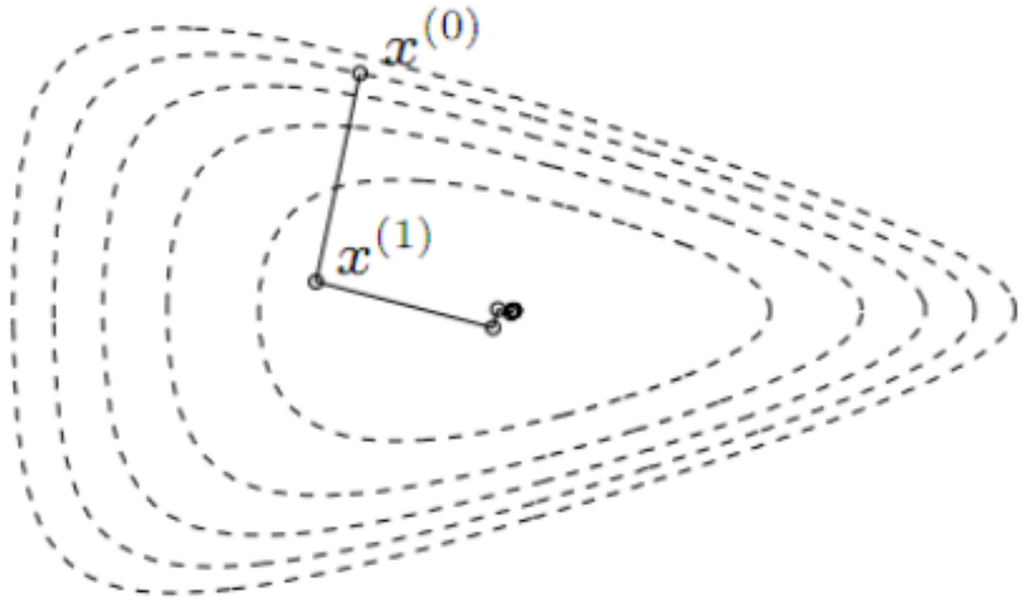
http://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/

Gradient Descent: Example 2

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



backtracking line search



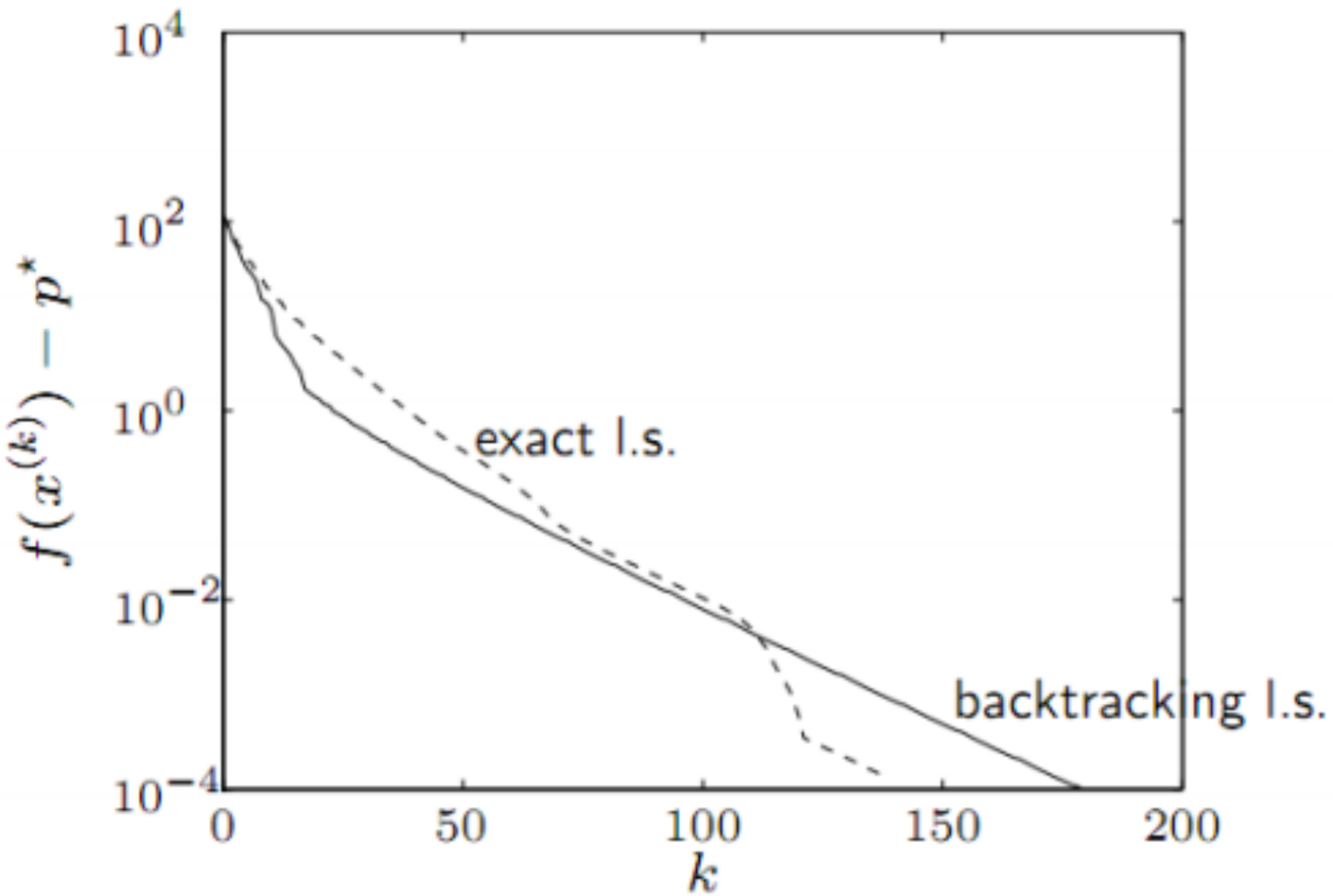
exact line search

Boyd & Landenberghe's Book on Convex Optimization

Gradient Descent: Example 3

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$

A problem in R^{100}



Boyd & Landenberghe's Book on Convex Optimization

Limitations of Gradient Descent

- Step size search may be expensive
- Convergence is slow for ill-conditioned problems
- Convergence speed depends on initial starting position
- Does not work for non differentiable or constrained problems

Newton's Method

- Assumes function is locally quadratic:

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^\top \Delta x + \frac{1}{2} \Delta x^\top \nabla^2 f(x) \Delta x$$

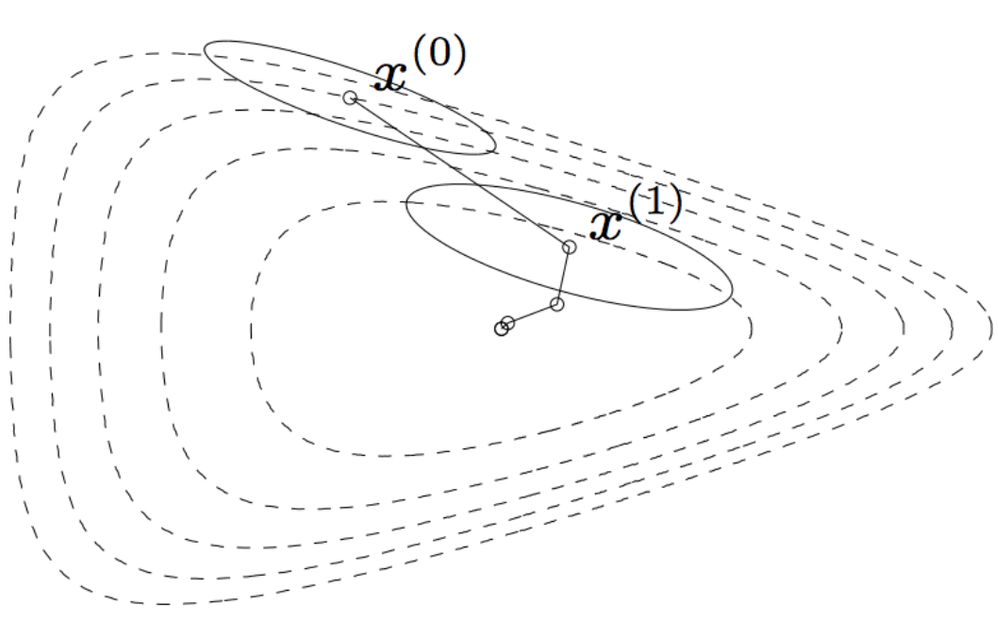
- Choose step direction:

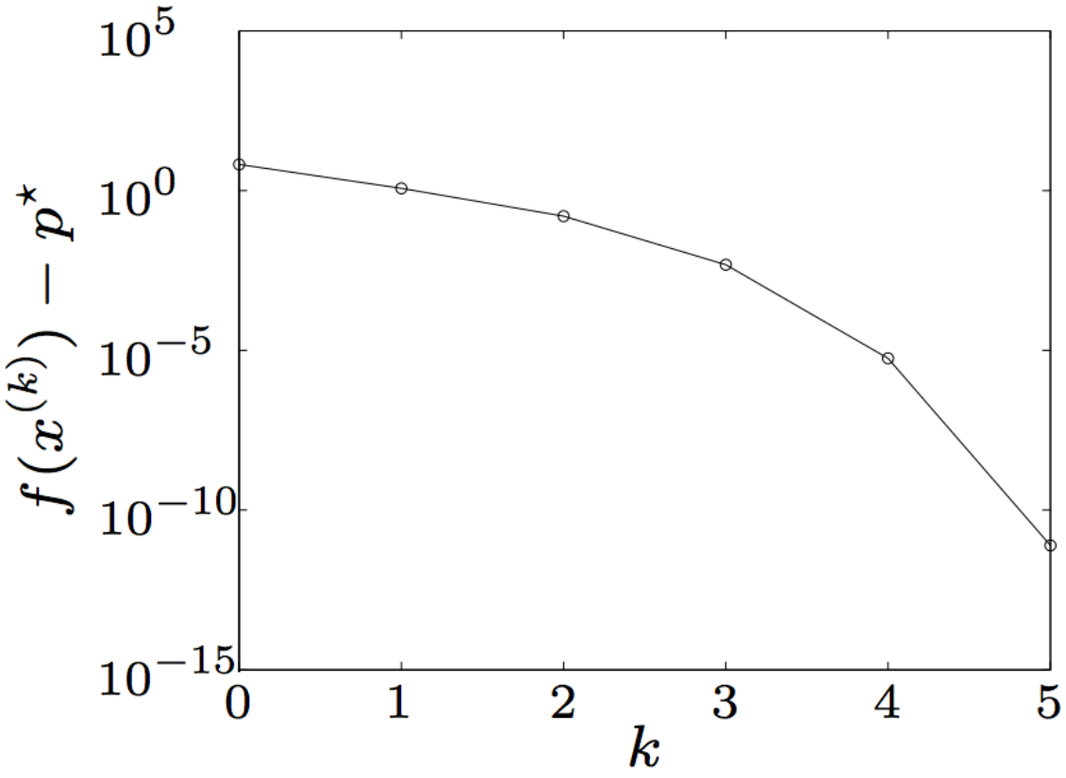
$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- Method is often faster than gradient descent

Newton's Method: Example 2

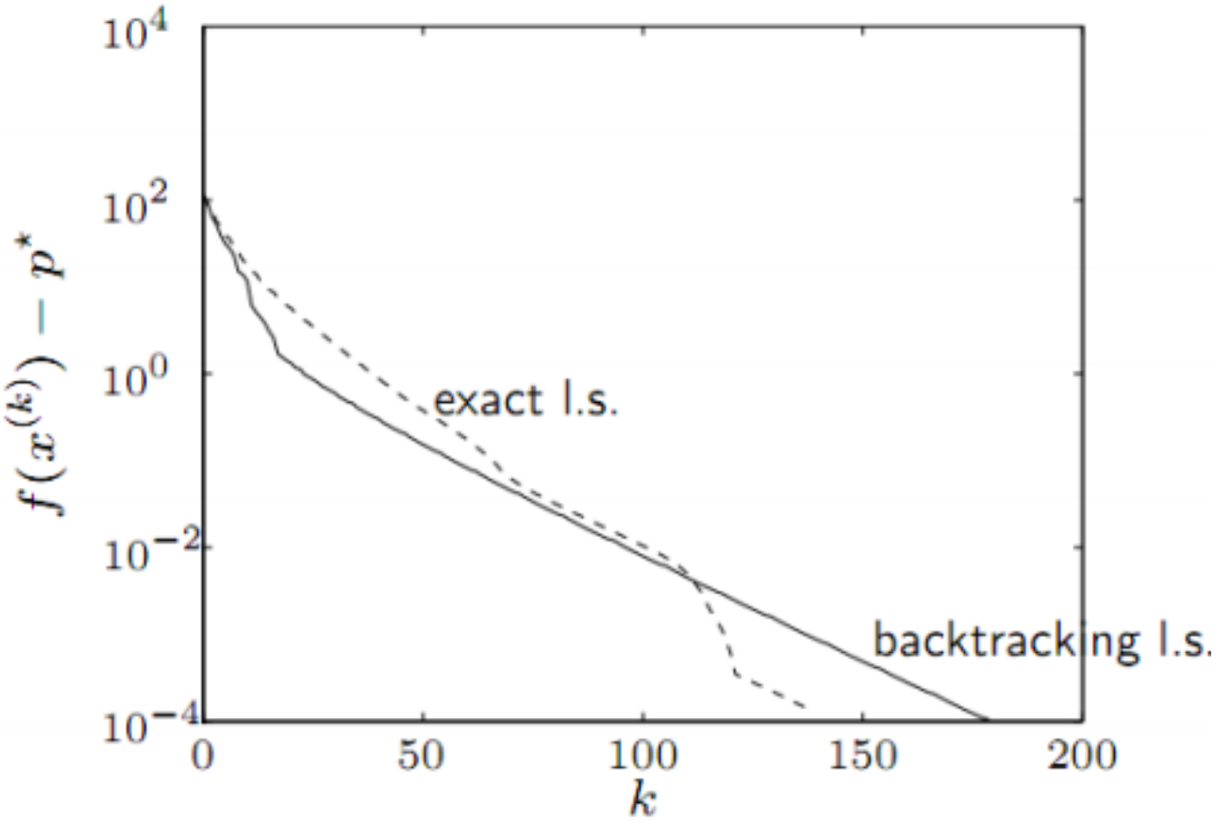
$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



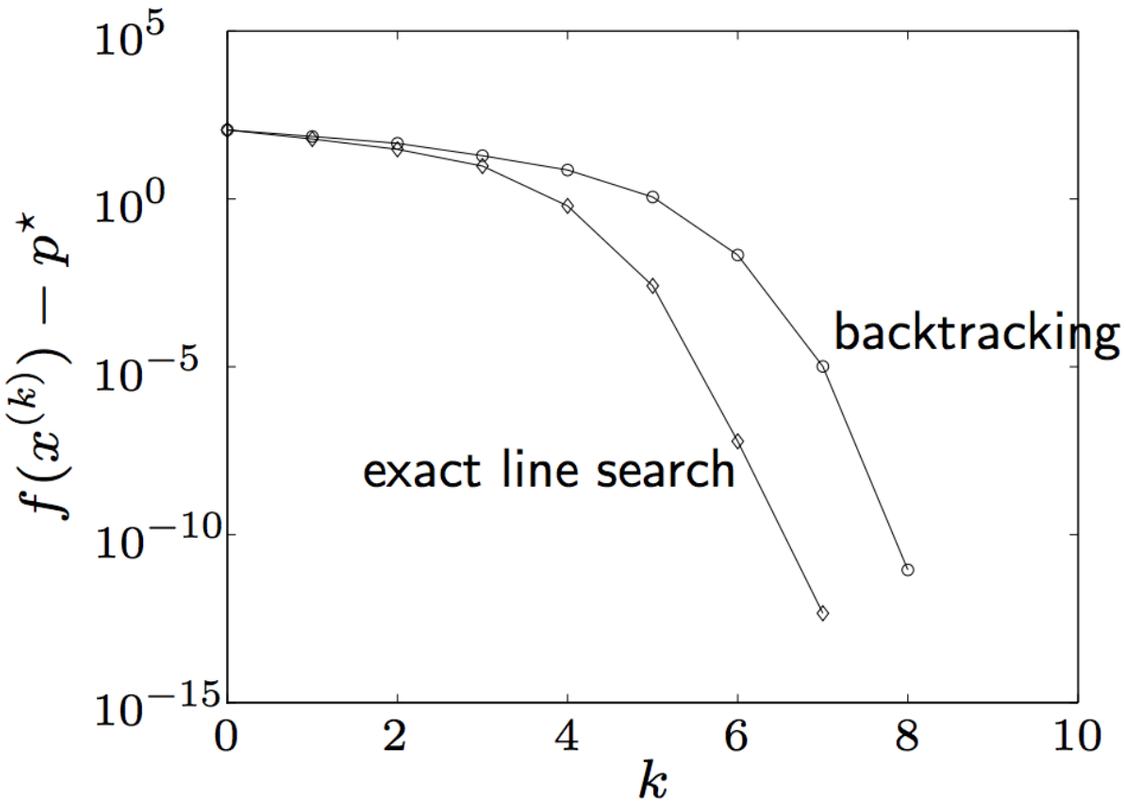


converges in only 5 steps

Newton's Method: Example 3



Gradient Descent



Newton's Method

Disadvantages of Newton's Method

- Hessian is expensive to invert
- Hessian must be positive definite
- May make huge, uncontrolled steps which can cause instability

There are several other methods which build on gradient descent and Newton's method such as conjugate gradient descent and Quasi-Newton methods

Constrained Optimization Algorithms

$$\min_x \ f_0(x)$$

$$\text{s.t } f_k(x) \leq 0, \ k = 1, \cdots, K$$

Lagrange Duality

- Bound or solve an optimization problem via a different optimization problem

- Optimization problems (even non-convex) can be transformed to their dual problems

- Purpose of the dual problem is to determine the lower bounds for the optimal value of the original problem

- Under certain conditions, solutions of both problems are equal and the dual problem often offers easier and analytical way to the solution

Reasons Why Dual is Easier

- Dual problem is unconstrained or has simple constraints
- Dual objective is differentiable or has a simple non differentiable term
- Exploit separable structure in the decomposition for easier algorithm

Construct the Dual

Original optimization problem or primal problem

$$\min_x \ f_0(x)$$

$$\text{s.t. } f_k(x) \leq 0, k = 1, 2, \cdots, K$$

$$h_j(x) = 0, j = 1, 2, \cdots, J$$

Lagrangian

$$L(x, \lambda, v) = f_0(x) + \sum_k \lambda_k f_k(x) + \sum_j v_j h_j(x)$$

Lagrange multipliers or dual variables

Construct the Dual

Original optimization problem or primal problem

$$\min_x \ f_0(x)$$

$$\text{s.t. } f_k(x) \leq 0, k = 1, 2, \cdots, K$$

$$h_j(x) = 0, j = 1, 2, \cdots, J$$

infimum is the element that is smallest or equal to all elements in the set

Dual problem

$$\max g(\lambda, v) = \inf_x L(x, \lambda, v)$$

$$\text{subject to } \ \lambda \geq 0$$

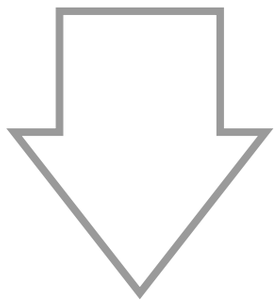
dual function is always lower bound for optimal value of original function

$$g(\lambda, v) \leq L(\tilde{x}, \lambda, v) \leq f_0(\tilde{x})$$

Lagrange Dual: Separable Example

$$\begin{aligned} \min \;\; & f_1(x_1) + f_2(x_2) \\ \text{subject to } & A_1 x_1 + A_2 x_2 \leq b \end{aligned}$$

coupling constraint in primal problem



$$\begin{aligned} \max \;\; & -f_1^*(-A_1^\top z) - f_2^*(-A_2^\top z) - b^\top z \\ \text{subject to } & z \geq 0 \end{aligned}$$

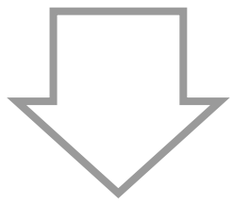
dual problem can be easily solved by gradient projection

Lagrange Dual: SVM Example

$$\min \ \frac{1}{2}||w||^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ i = 1, \cdots, m$$

classical SVM problem assuming linearly separable can be solved using commercial quadratic programming



$$L(w, b, \lambda) = \frac{1}{2}||w||^2 - \sum_i \lambda_i [y_i(w^\top x_i + b) - 1]$$

Lagrange Dual: SVM Example II

- Take derivative of L with respect to w and b and set them to zero

- Replace the definitions in the Lagrangian for the dual formulation and simplifying

$$\max_{\lambda \geq 0, \sum_i \lambda_i y_i = 0} \sum_i \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j < x_i, x_j >$$

- Note that the formulation only uses inner products between x and the support vectors which allows the kernel trick for SVMs

Projected Gradient Descent

- Constrained optimization subject to convex set

$$\min \ f(x)$$
$$\text{s.t. } x \in C$$

- Projected gradient descent step:

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

- Projection onto a set c is:

$$P_C(x) = \arg\min_{v \in C} ||x - v||$$

Projected Gradient Descent: Example 1

- Linear regression with non-negative weights

$$\min \ ||y - X\beta||_2^2$$

$$\text{s.t. } \beta \in \mathbb{R}_+^d$$

- Projection is of the form:

$$P_{\mathbb{R}_+}(x)_i = \max(x_i, 0)$$

- Algorithm: same as gradient descent for linear regression except that any values less than 0 are set to 0

Projected Gradient Descent: Ridge Regression

- Ridge Regression (one form):

$$\begin{aligned} \min \ & ||y - X\beta||_2^2 \\ \text{s.t.} \ & ||\beta||_2 \leq s \end{aligned}$$

- Projection is of the form:

$$P_X(z) = \frac{z}{\max(s, ||z||_2)}$$

- Algorithm: same as gradient descent for linear regression except that all values are scaled by the norm or s

Proximal Gradient Descent

- Can be referred to as composite gradient descent or generalized gradient descent

- Formulation for decomposable functions where one of the functions may not necessarily be differentiable

$$f(x) = \underbrace{g(x)}_{\text{convex, differentiable}} + \underbrace{h(x)}_{\text{convex}}$$

- If both are differentiable or h(x) = 0, then it's standard gradient descent

- If h(x) is the indicator function, then its projected gradient

Proximal Gradient Descent Step

- Define proximal mapping:

$$\text{prox}_t(x) = \text{argmin} \frac{1}{2t}||x - z||_2^2 + h(z)$$

- Proximal gradient step has the form:

$$x^+ = \text{prox}_t(x - t\nabla g(x))$$

- But… we just swapped one minimization problem for another

Proximal Gradient Descent Advantages

- The proximal mapping can be computed analytically for many important functions h

- Mapping does not depend on g at all, just on h

- Smooth part g can be complicated, but we only need to compute its gradients

- Simple to implement and is a fast first-order method assuming the proximal map is well-known and inexpensive to compute

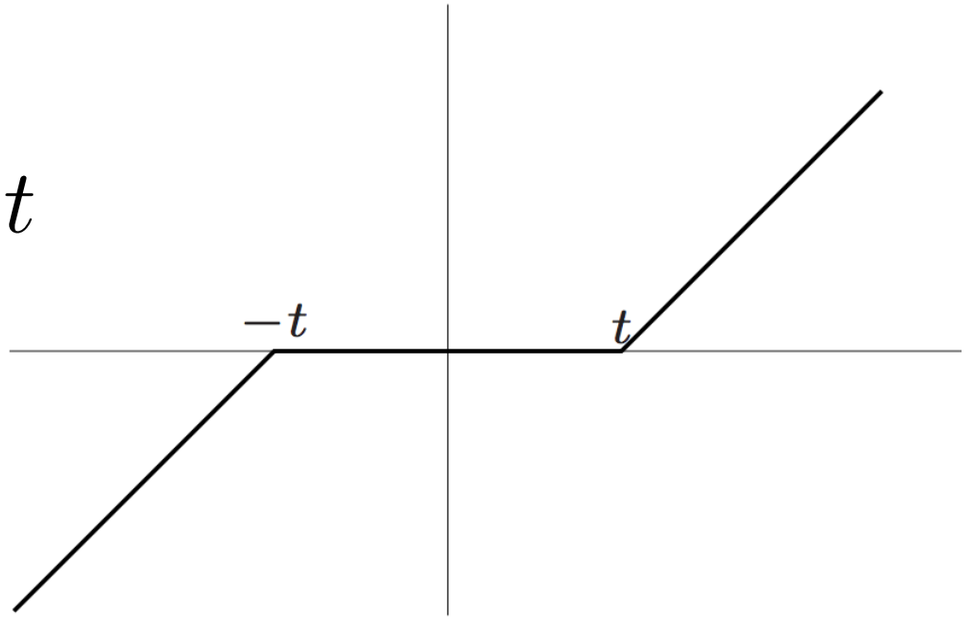
Proximal Gradient Descent: LASSO

- Optimization problem:

$$\min g(x) + ||x||_1$$

- Proximal mapping:

$$\text{prox}_t(z)_i = \begin{cases} z_i - t, & z_i \geq t \\ 0, & -t \leq z_i \leq t \\ z_i + t, & z_i \geq t \end{cases}$$



- Known as iterative soft-thresholding

Proximal Gradient Descent: Matrix Completion

- Given a matrix Y and only some observed entries, we want to fill in the remaining entries (e.g., recommendation system)

$$\min \frac{1}{2} \sum_{(i,j)\in\Omega} (Y_{ij} - B_{ij})^2 + \lambda ||B||_{\text{tr}}$$

- Proximal gradient update step:

$$B^+ = S_{\lambda t}\left(B + t(P_\Omega(Y) - P_\Omega(B))\right)$$

- Soft-impute algorithm, which is simple and effective for matrix completion

http://stat.cmu.edu/~ryantibs/convexopt/lectures/08-prox-grad.pdf

Some Resources for Today's Lecture

- Boyd & Landenberghe's book on Convex Optimization
  https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

- Nocedal & Wright's book on Numerical Optimization
  http://home.agh.edu.pl/~pba/pdfdoc/Numerical_Optimization.pdf

- Parish & Boyd on Proximal Algorithms
  https://web.stanford.edu/~boyd/papers/pdf/prox_algs.pdf

- Ryan Tibshirani's course on Convex Optimization
  http://stat.cmu.edu/~ryantibs/convexopt/