Optimisation et simulation numérique

Large Scale Optimization

Outline

- First-order methods: introduction
- Exploiting structure
- First order algorithms
  - Subgradient methods
  - Gradient methods
- Other algorithms
  - Coordinate descent methods
  - Franke-Wolfe
  - Dykstra, alternating projection
  - Stochastic optimization

First-order methods: introduction

- Most of these methods are very old (1950-. . .)
- Very large catalog of algorithms, no unifying theory as in IPM
- Many variations around a few key algorithmic templates
- Better scaling, worst dependence on precision target
- In practice: algorithmic choices are dictated by **problem structure**.

What subproblem (projection, etc...) can you solve efficiently?

First Order Algorithms

First-order methods: introduction

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

In theory:

- The theoretical convergence speed of gradient based methods is mostly controlled by the smoothness of the objective.
- Obviously, the geometry of the (convex) feasible set also has an impact.

Convex objective $f(x)$	**Iterations. . .**
Nondifferentiable	$O(1/\epsilon^2)$
Differentiable	$O(1/\epsilon^2)$
Smooth (Lipschitz gradient)	$O(1/\sqrt{\epsilon})$
Strongly convex	$O(\log(1/\epsilon))$

In practice:

- Compared to IPM, much larger gap between theoretical complexity guarantees and empirical performance.
- Conditioning, well-posedness, etc. also have a very strong impact.

First-order methods: introduction

Solve

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

in $x \in \mathbb{R}^n$, with $C \subset \mathbb{R}^n$ convex.

Main assumptions in the subgradient/gradient methods that follow:

- The gradient $\nabla f(x)$ or a subgradient can be computed efficiently.
- If C is not $\mathbb{R}^n$, for any $y \in \mathbb{R}^n$, the following **subproblem can be solved efficiently**

  $$\begin{array}{ll} \text{minimize} & y^T x + d(x) \\ \text{subject to} & x \in C \end{array}$$

  in the variable $x \in \mathbb{R}^n$, where $d(x)$ is a **strongly convex** function.

Typically, $d(x) = \|x\|_2$ and this is an Euclidean projection.

Subgradient Method

Subgradient

- Suppose that f is a convex function with $\mathbf{dom} f = \mathbb{R}^n$, and that there is a vector $g \in \mathbb{R}^n$ such that:

$$f(y) \geq f(x) + g^T(y - x), \quad \text{for all } y \in \mathbb{R}^n$$

- The vector g is called a **subgradient** of f at x, we write $g \in \partial f$.
- Of course, if f is differentiable, the gradient of f at x satisfies this condition
- The subgradient defines a **supporting hyperplane** for f at the point x

Subgradient Methods

Subgradient method:

- Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex
- We update the current point x_k according to:

$$x_{k+1} = x_k + \alpha_k g_k$$

  where g_k is a subgradient of f at x_k

- α_k is the step size sequence
- Similar to gradient descent but, not a descent method . . .
- Instead: use the best point and the minimum function value found so far

Subgradient Methods

Step size strategies:

- Constant step size: $\alpha_k = h$ for all $k \geq 0$
- Constant step length: $\alpha_k / \|g_k\| = h$ for all $k \geq 0$
- Square summable but not summable:

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty$$

- Nonsummable diminishing:

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \text{and} \quad \lim_{k \to \infty} \alpha_k = 0$$

Subgradient Methods

Convergence:

Assuming $\|g\|_2 \leq G$, for all $g \in \partial f$, we can show

$$f_{\text{best}} - f^\star \leq \frac{\mathbf{dist}(x_1, x^*) + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i}$$

For constant step $\alpha_i = h$, this becomes

$$f_{\text{best}} - f^\star \leq \frac{\mathbf{dist}(x_1, x^*)}{2hk} + G^2 h/2$$

to get an ϵ solution, we set $h = 2\epsilon/G^2$ and

$$\frac{\mathbf{dist}(x_1, x^*)}{2hk} \leq \epsilon$$

hence

$$k \geq \frac{\mathbf{dist}(x_1, x^*) G^2}{4\epsilon^2}.$$

Subgradient Methods

- If the problem has constraints:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

  where $C \subset \mathbb{R}^n$ is a convex set

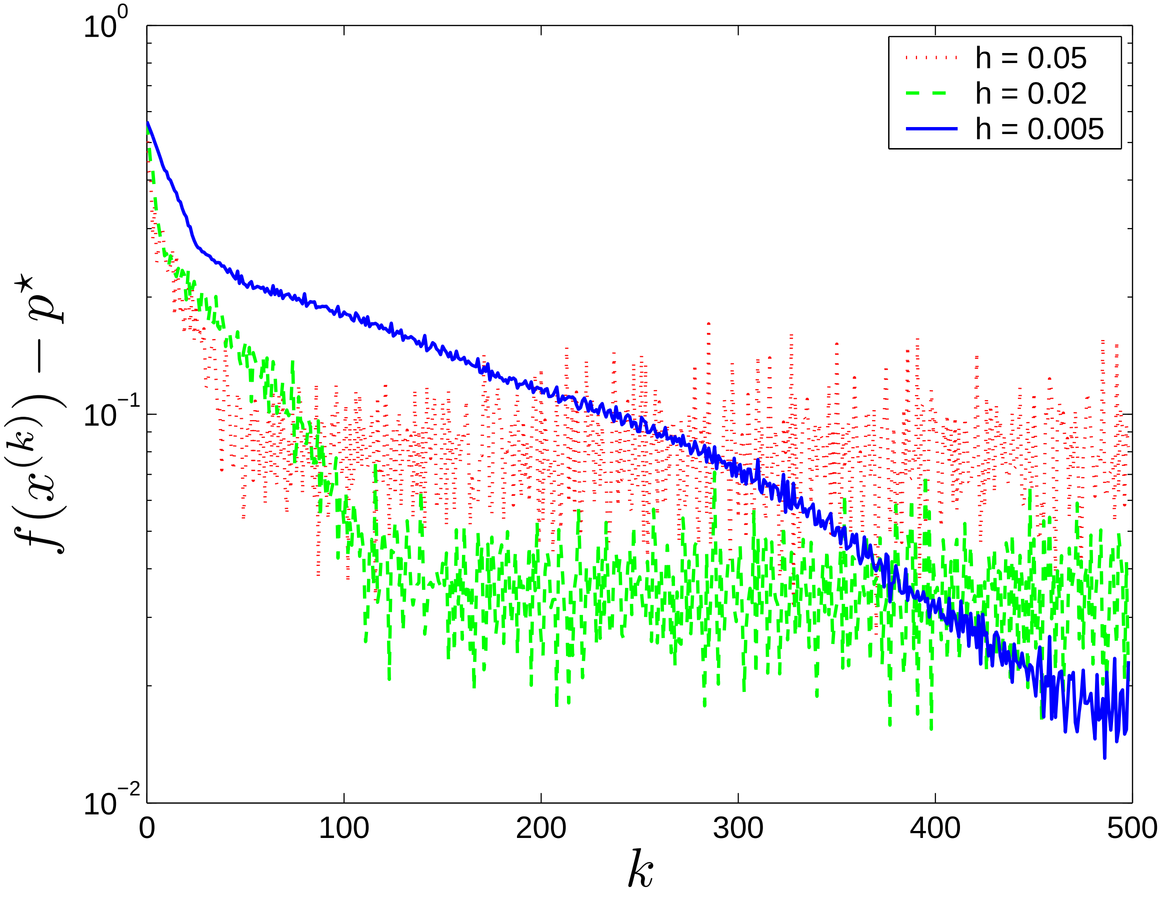
- Use the Euclidean projection $p_C(\cdot)$

$$x_{k+1} = p_C(x_k + \alpha_k g_k)$$

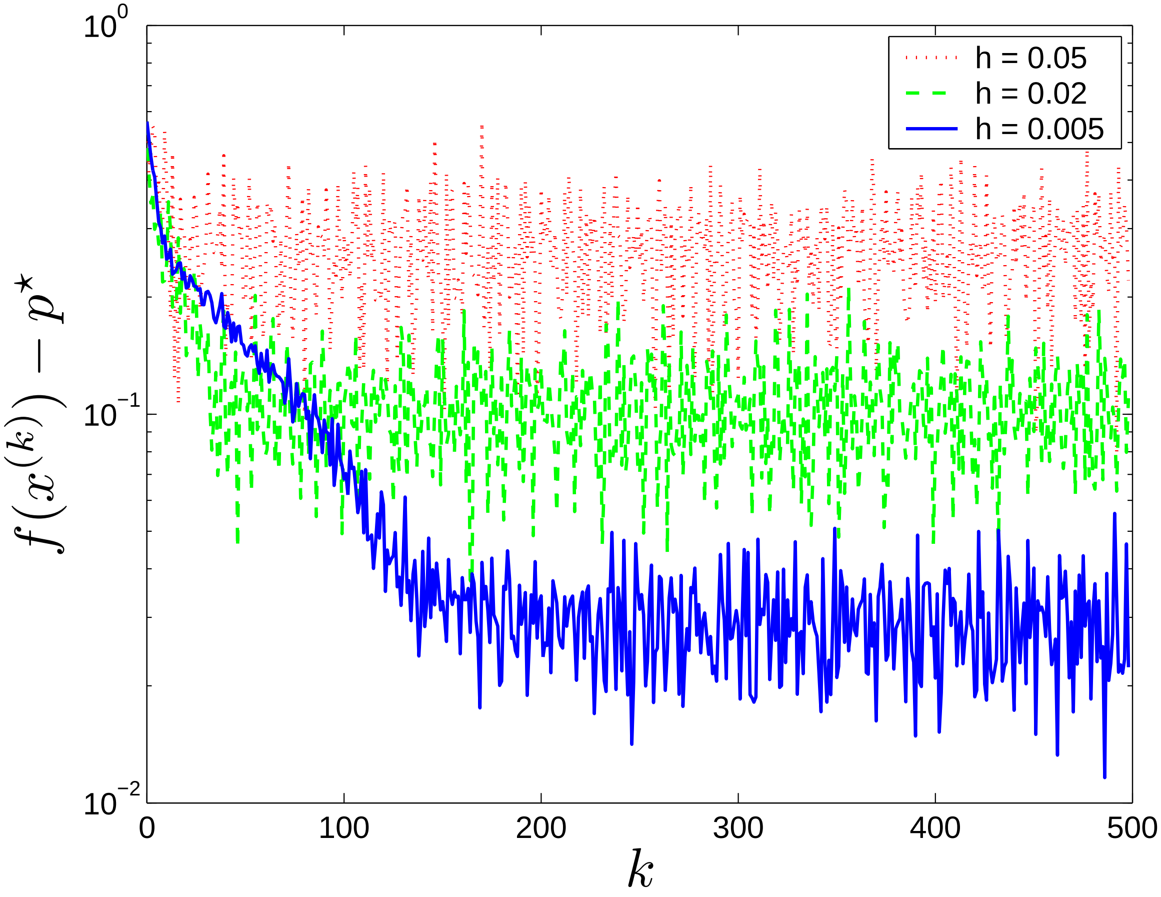
- Similar complexity analysis
- Some numerical examples on piecewise linear minimization. . . Problem instance with $n = 10$ variables, $m = 100$ terms

Subgradient Methods: Numerical Examples

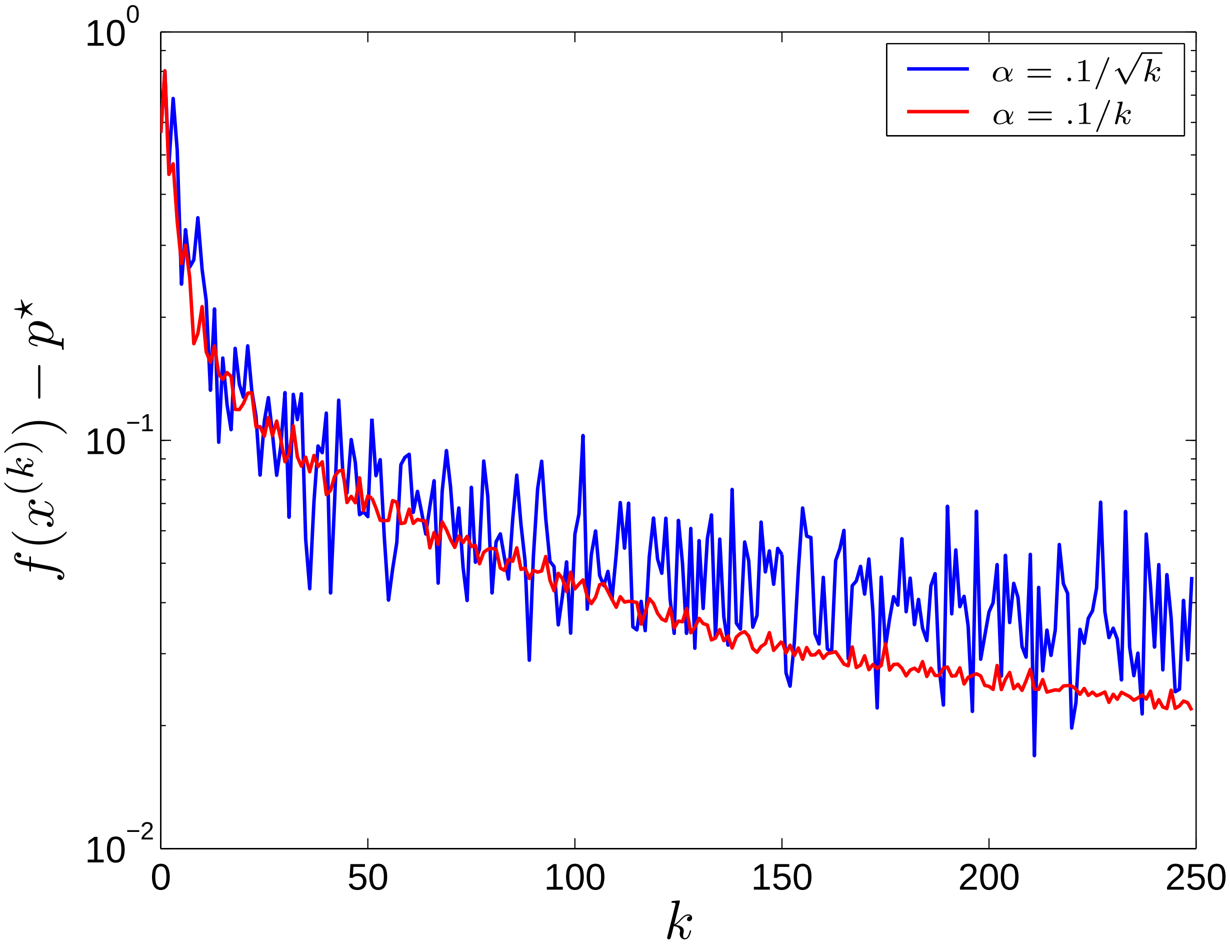
Constant step length, $h = 0.05,\ 0.02, 0.005$



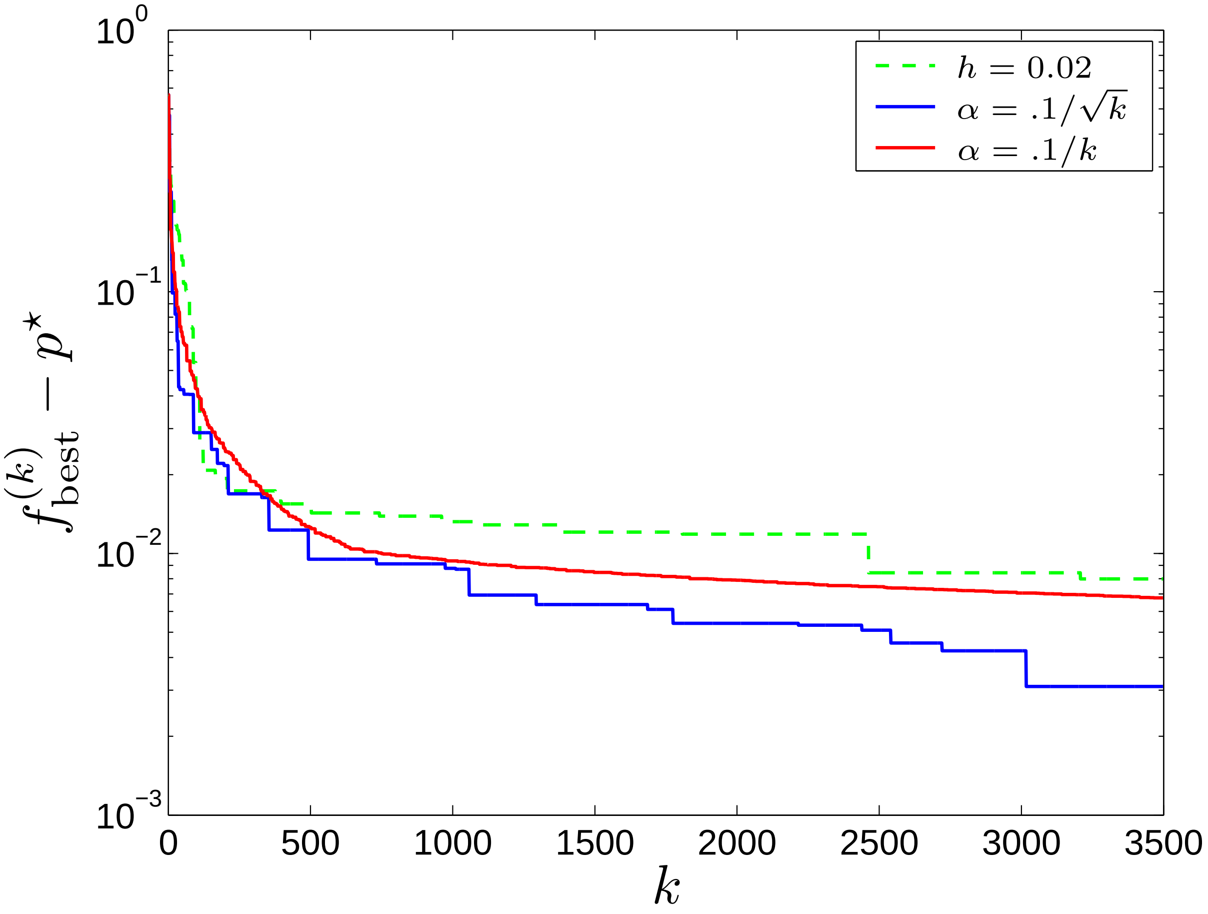
Constant step size $h = 0.05,\ 0.02,\ 0.005$



Diminishing step rule $\alpha = 0.1/\sqrt{k}$ and square summable step size rule $\alpha = 0.1/k$.



Constant step length $h = 0.02$, diminishing step size rule $\alpha = 0.1/\sqrt{k}$, and square summable step rule $\alpha = 0.1/k$



Gradient Descent

Gradient descent method

general descent method with $\Delta x = -\nabla f(x)$

```
given a starting point $x \in \mathbf{dom}\, f$.
repeat
    1. $\Delta x := -\nabla f(x)$.
    2. Line search. Choose step size $t$ via exact or backtracking line search.
    3. Update. $x := x + t\Delta x$.
until stopping criterion is satisfied.
```

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \leq \epsilon$
- convergence result: for **strongly convex** f,

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

  $c \in (0,1)$ depends on m, $x^{(0)}$, line search type.

- this means $O(\log 1/\epsilon)$ iterations to get ϵ solution.
- very simple, but often very slow; rarely used in practice

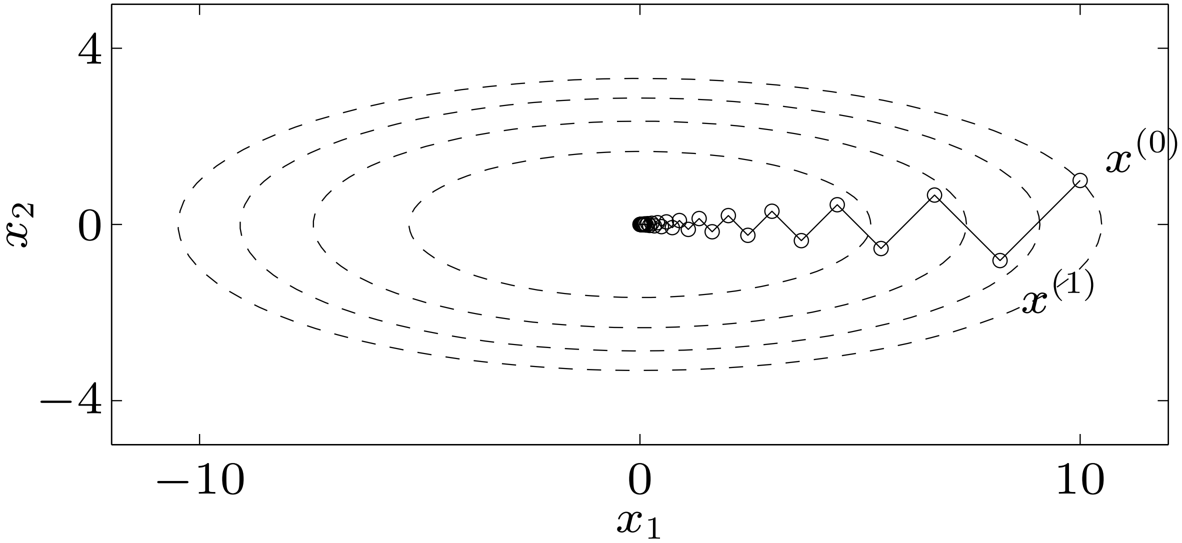
quadratic problem in $\mathbb{R}^2$

$$f(x) = (1/2)(x_1^2 + \gamma x_2^2) \qquad\qquad (\gamma > 0)$$

with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

- very slow if $\gamma \gg 1$ or $\gamma \ll 1$
- example for $\gamma = 10$:



Large Scale Optimization

Outline

- First-order methods: introduction
- Exploiting structure
- First order algorithms
  - Subgradient methods
  - Gradient methods
  - Accelerated gradient methods
- **Other algorithms**
  - Coordinate descent methods
  - Franke-Wolfe
  - Dykstra, alternating projection
  - Stochastic optimization

Coordinate Descent

Coordinate Descent

We seek to solve

$$\begin{array}{ll} \mbox{minimize} & f(x) \\ \mbox{subject to} & x \in C \end{array}$$

in the variable $x \in \mathbb{R}^n$, with $C \subset \mathbb{R}^n$ convex.

- Our main assumption here is that C **is a product of simpler sets**. We rewrite the problem

  $$\begin{array}{ll} \mbox{minimize} & f(x_1, \ldots, x_p) \\ \mbox{subject to} & x_i \in C_i, \quad i = 1, \ldots, p \end{array}$$

  where $C = C_1 \times \ldots \times C_p$.
- This helps if the minimization subproblems

  $$\min_{x_i \in C_i} f(x_1, \ldots, x_i, \ldots, x_p)$$

  can be solved very efficiently (or in closed-form).

Coordinate Descent

Algorithm. The algorithm simply computes the iterates $x^{(k+1)}$ as

$$\begin{array}{rcl} x_i^{(k+1)} & = & \underset{x_i \in C_i}{\operatorname{argmin}} f(x_1^{(k)}, \ldots, x_i^{(k)}, \ldots, x_p^{(k)}) \\ x_j^{(k+1)} & = & x_j^{(k)}, \quad j \neq i \end{array}$$

for a certain $i \in [1, p]$, cycling over all indices in $[1, p]$.

Convergence.

- Complexity analysis similar to coordinate-wise gradient descent (or steepest descent in ℓ_1 norm).
- Need $f(x)$ strongly convex to get linear complexity bound.
- Few clean results outside of this setting.

Coordinate Descent

Example.

- Consider the box constrained minimization problem

$$\begin{array}{ll} \mbox{minimize} & x^TAx+b^Tx \\ \mbox{subject to} & \|x\|_\infty \leq 1 \end{array}$$

  in the variable $x \in \mathbb{R}^n$. We assume $A \succ 0$.

- The set $\|x\|_\infty \leq 1$ is a box, i.e. a product of intervals.
- Each minimization subproblem means solving a second order equation.
- The dual is

$$\min_{y \in \mathbb{R}^n} \; (b+y)^T A^{-1}(b+y) - 4\|y\|_1$$

  which can be interpreted as a penalized regression problem in the variable $y \in \mathbb{R}^n$.

Franke-Wolfe

Franke-Wolfe

- Classical first order methods for solving

$$\begin{array}{ll}\text{minimize} & f(x)\\ \text{subject to} & x \in C,\end{array}$$

  in $x \in \mathbb{R}^n$, with $C \subset \mathbb{R}^n$ convex, relied on the assumption that the following subproblem could be solved efficiently

$$\begin{array}{ll}\text{minimize} & y^T x + d(x)\\ \text{subject to} & x \in C,\end{array}$$

  in the variable $x \in \mathbb{R}^n$, where $d(x)$ is a strongly convex function.

- The method detailed here assumes instead that the **affine minimization subproblem**

$$\begin{array}{ll}\text{minimize} & d^T x\\ \text{subject to} & x \in C\end{array}$$

  can be solved efficiently for any $y \in \mathbb{R}^n$.

Franke-Wolfe

Algorithm.

- Choose $x_0 \in C$.
- **For** $k = 1, \ldots, k^{max}$ **iterate**

1. Compute $\nabla f(x_k)$
2. Solve
$$\begin{array}{ll} \text{minimize} & x^T \nabla f(y_k) \\ \text{subject to} & x \in C \end{array}$$
in $x \in \mathbb{R}^n$, call the solution x_d.
3. Update the current point
$$x_{k+1} = x_k + \frac{2}{k+2}(x_d - x_k)$$

Note that all iterates are feasible.

Franke-Wolfe

- **Complexity.** Assume that f is differentiable. Define the curvature C_f of the function $f(x)$ as

$$C_f \triangleq \sup_{\substack{s,x\in\mathcal{M},\ \alpha\in[0,1],\\ y=x+\alpha(s-x)}} \frac{1}{\alpha^2}(f(y)-f(x)-\langle y-x,\nabla f(x)\rangle).$$

The Franke-Wolfe algorithm will then produce an ϵ solution after

$$N_{\max}=\frac{4C_f}{\epsilon}$$

iterations.

Franke-Wolfe

- **Stopping criterion.** At each iteration, we get a lower bound on the optimum as a byproduct of the affine minimization step. By convexity

$$f(x_k) + \nabla f(x_k)^T (x_d - x_k) \leq f(x), \quad \text{for all } x \in C$$

and finally, calling f^* the optimal value of problem, we obtain

$$f(x_k) - f^* \leq \nabla f(x_k)^T (x_k - x_d).$$

This allows us to bound the suboptimality of iterate at no additional cost.

Dykstra, alternating projection

Dykstra, alternating projection

We focus on a simple **feasibility problem**

$$\text{find } x \in C_1 \cap C_2$$

in the variable $x \in \mathbb{R}^n$ with $C_1, C_2 \subset \mathbb{R}^n$ two convex sets.

We assume now that the projection problems on C_i are easier to solve

$$\begin{array}{ll} \text{minimize} & \|x-y\|_2 \\ \text{subject to} & x \in C_i \end{array}$$

in $x \in \mathbb{R}^n$.

Dykstra, alternating projection

Algorithm (alternating projection)

- Choose $x_0 \in \mathbb{R}^n$.
- **For** $k = 1, \ldots, k^{max}$ **iterate**

1. Project on C_1
$$x_{k+1/2} = \operatorname*{argmin}_{x \in C_1} \|x - x_k\|_2$$
2. Project on C_2
$$x_{k+1} = \operatorname*{argmin}_{x \in C_2} \|x - x_{k+1/2}\|_2$$

Convergence. We can show $\mathbf{dist}(x_k, C_1 \cap C_2) \to 0$. Linear convergence provided some additional regularity assumptions.

Dykstra, alternating projection

Algorithm (Dykstra)

- Choose $x_0, z_0 \in \mathbb{R}^n$.
- **For** $k = 1, \ldots, k^{max}$ **iterate**

1. Project on C_1

$$x_{k+1/2} = \underset{x \in C_1}{\operatorname{argmin}} \|x - z_k\|_2$$

2. Update

$$z_{k+1/2} = 2x_{k+1/2} - z_k$$

3. Project on C_2

$$x_{k+1} = \underset{x \in C_2}{\operatorname{argmin}} \|x - z_{k+1/2}\|_2$$

4. Update

$$z_{k+1} = z_k + x_{k+1} - x_{k+1/2}$$

Convergence. Usually faster than simple alternating projection.

Stochastic Optimization

Stochastic Optimization

Solve

$$\begin{array}{ll} \text{minimize} & \mathbf{E}[f(x,\xi)] \\ \text{subject to} & x \in C, \end{array}$$

in $x \in \mathbb{R}^n$, where C is a simple convex set. The key difference here is that the function we are minimizing is **stochastic.**

Batch method. A simple option is to approximate the problem by

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} f(x,\xi_m) \\ \text{subject to} & x \in C, \end{array}$$

where ξ_i are sampled from the distribution of ξ.

Sampling is costly, we can do better. . .

Stochastic Optimization

Let $p_C(\cdot)$ be the Euclidean projection operator on C.

Algorithm (Robust stochastic averaging)

- Choose $x_0 \in C$ and a step sequence $\gamma_j > 0$.
- **For** $k = 1, \ldots, k^{max}$ **iterate**

  1. Compute a subgradient

  $$g \in \partial f(x_k, \xi_k)$$

  2. Update the current point

  $$x_{k+1} = p_C(x_k - \gamma_k g)$$

Stochastic Optimization

Complexity.

- Call $\tilde{x}_k = \sum_{i=1}^k \gamma_i x_i$ and assume

$$\max_{x \in C} \mathbf{E}[\|g\|_2^2] \leq M^2, \quad \text{and} \quad D_C = \max_{x,y \in C} \|x-y\|_2$$

- If we set $\gamma_i = D_C/(M\sqrt{k})$, we have

$$\mathbf{E}[f(\tilde{x}_k) - f^*] \leq \frac{D_C M}{\sqrt{k}}$$

- Furthermore, if we assume

$$\mathbf{E}\left[\exp\left(\frac{\|g\|_2^2}{M^2}\right)\right] \leq e, \quad \text{for all } g \in \partial f(x_k, \xi) \text{ and } x \in C$$

we get

$$\mathbf{Prob}\left[f(\tilde{x}_k) - f^* \geq \frac{D_C M}{\sqrt{k}}(12 + 2t)\right] \leq 2\exp(-t).$$

Applications

Outline

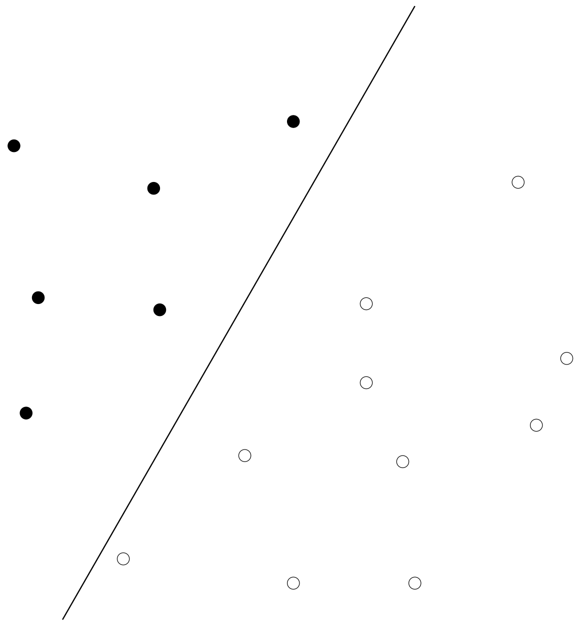
- Geometrical problems
- Approximation problems
- Distance reconstruction
- Mixing and unfolding
- Collaborative prediciton

Geometrical problems

Linear discrimination

separate two sets of points $\{x_1, \ldots, x_N\}$, $\{y_1, \ldots, y_M\}$ by a hyperplane:

$$a^T x_i + b_i > 0, \quad i = 1, \ldots, N, \qquad a^T y_i + b_i < 0, \quad i = 1, \ldots, M$$



homogeneous in a, b, hence equivalent to

$$a^T x_i + b_i \geq 1, \quad i = 1, \ldots, N, \qquad a^T y_i + b_i \leq -1, \quad i = 1, \ldots, M$$

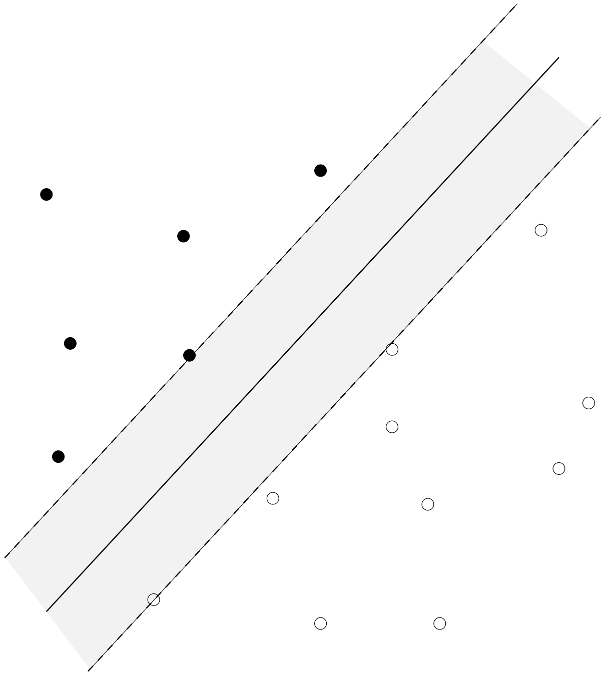
a set of linear inequalities in a, b

Robust linear discrimination

(Euclidean) distance between hyperplanes

$$\begin{aligned} \mathcal{H}_1 &= \{z \mid a^T z + b = 1\} \\ \mathcal{H}_2 &= \{z \mid a^T z + b = -1\} \end{aligned}$$

is $\mathbf{dist}(\mathcal{H}_1, \mathcal{H}_2) = 2/\|a\|_2$



to separate two sets of points by maximum margin,

$$\begin{array}{ll} \text{minimize} & (1/2)\|a\|_2 \\ \text{subject to} & a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1, \quad i = 1, \ldots, M \end{array} \qquad (1)$$

(after squaring objective) a QP in a, b

Lagrange dual of maximum margin separation problem

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T\lambda + \mathbf{1}^T\mu \\ \text{subject to} & 2\left\|\sum_{i=1}^N \lambda_i x_i - \sum_{i=1}^M \mu_i y_i\right\|_2 \leq 1 \\ & \mathbf{1}^T\lambda = \mathbf{1}^T\mu, \quad \lambda \succeq 0, \quad \mu \succeq 0 \end{array} \qquad (2)$$

from duality, optimal value is inverse of maximum margin of separation

interpretation

- change variables to $\theta_i = \lambda_i/\mathbf{1}^T\lambda$, $\gamma_i = \mu_i/\mathbf{1}^T\mu$, $t = 1/(\mathbf{1}^T\lambda + \mathbf{1}^T\mu)$
- invert objective to minimize $1/(\mathbf{1}^T\lambda + \mathbf{1}^T\mu) = t$

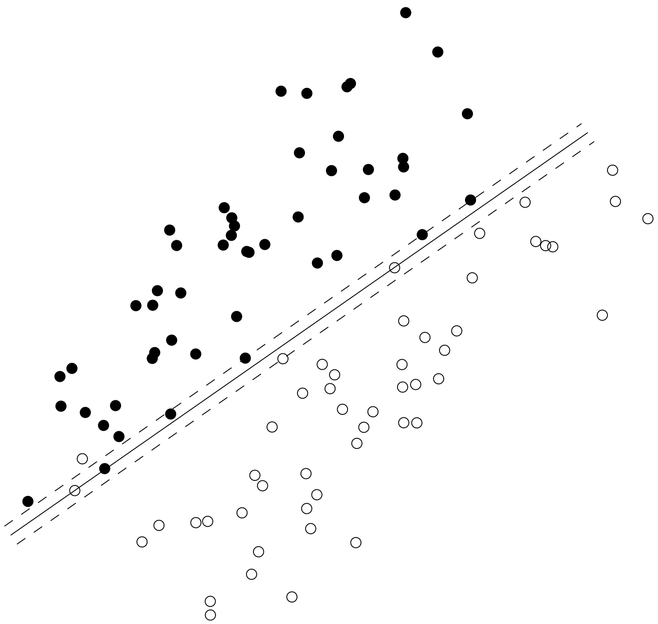
$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left\|\sum_{i=1}^N \theta_i x_i - \sum_{i=1}^M \gamma_i y_i\right\|_2 \leq t \\ & \theta \succeq 0, \quad \mathbf{1}^T\theta = 1, \quad \gamma \succeq 0, \quad \mathbf{1}^T\gamma = 1 \end{array}$$

optimal value is distance between convex hulls

Approximate linear separation of non-separable sets

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u + \mathbf{1}^T v \\ \text{subject to} & a^T x_i + b \geq 1 - u_i, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1 + v_i, \quad i = 1, \ldots, M \\ & u \succeq 0, \quad v \succeq 0 \end{array}$$

- an LP in a, b, u, v
- at optimum, $u_i = \max\{0, 1 - a^T x_i - b\}$, $v_i = \max\{0, 1 + a^T y_i + b\}$
- can be interpreted as a heuristic for minimizing #misclassified points

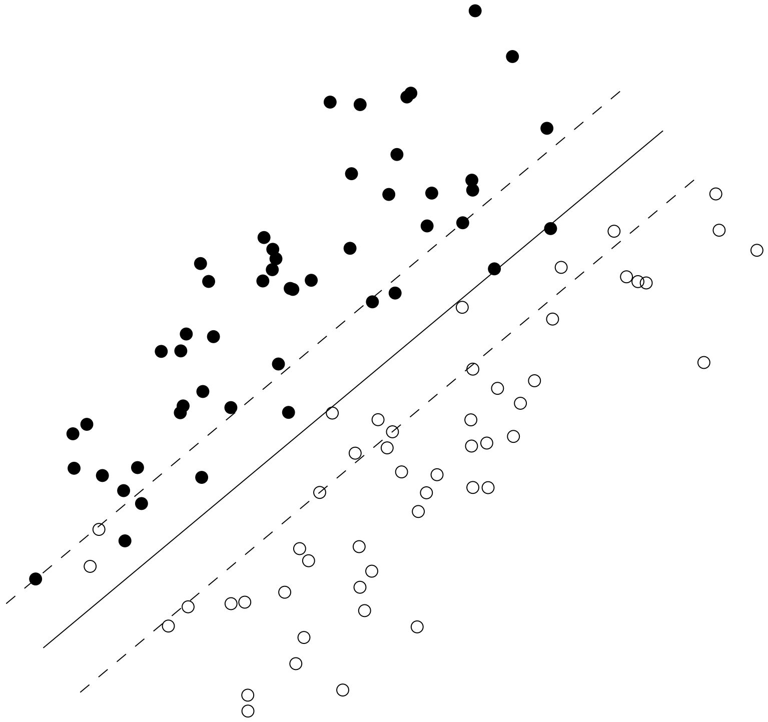


Support vector classifier

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \gamma(\mathbf{1}^T u + \mathbf{1}^T v) \\ \text{subject to} & a^T x_i + b \geq 1 - u_i, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1 + v_i, \quad i = 1, \ldots, M \\ & u \succeq 0, \quad v \succeq 0 \end{array}$$

produces point on trade-off curve between inverse of margin $2/\|a\|_2$ and classification error, measured by total slack $\mathbf{1}^T u + \mathbf{1}^T v$

same example as previous page, with $\gamma = 0.1$:



Support Vector Machines: Duality

Given m data points $x_i \in \mathbb{R}^n$ with labels $y_i \in \{-1, 1\}$.

- The maximum margin classification problem can be written

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|_2^2 + C\mathbf{1}^T z \\ \text{subject to} & y_i(w^T x_i) \geq 1 - z_i, \quad i = 1, \ldots, m \\ & z \geq 0 \end{array}$$

  in the variables w, $z \in \mathbb{R}^n$, with parameter $C > 0$.

- We can set $w = (w, \mathbf{1})$ and increase the problem dimension by 1. So we can assume w.l.o.g. $b = 0$ in the classifier $w^T x_i + b$.
- The Lagrangian is written

$$L(w, z, \alpha) = \frac{1}{2}\|w\|_2^2 + C\mathbf{1}^T z + \sum_{i=1}^m \alpha_i (1 - z_i - y_i w^T x_i)$$

  with dual variable $\alpha \in \mathbb{R}_+^m$.

Support Vector Machines: Duality

- The Lagrangian can be rewritten

$$L(w,z,\alpha)=\frac{1}{2}\left(\left\|w-\sum_{i=1}^{m}\alpha_i y_i x_i\right\|_2^2-\left\|\sum_{i=1}^{m}\alpha_i y_i x_i\right\|_2^2\right)+(C\mathbf{1}-\alpha)^T z+\mathbf{1}^T\alpha$$

  with dual variable $\alpha \in \mathbb{R}^n_+$.

- Minimizing in (w,z) we form the dual problem

$$\begin{array}{ll}\text{maximize} & -\frac{1}{2}\left\|\sum_{i=1}^{m}\alpha_i y_i x_i\right\|_2^2+\mathbf{1}^T\alpha\\ \text{subject to} & 0\leq\alpha\leq C\end{array}$$

- At the optimum, we must have

$$w=\sum_{i=1}^{m}\alpha_i y_i x_i \quad \text{and} \quad \alpha_i = C \text{ if } z_i>0$$

  (this is the representer theorem).

Support Vector Machines: the kernel trick

- If we write X the data matrix with columns x_i, the dual can be rewritten

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\alpha^T \,\mathbf{diag}(y) X^T X \,\mathbf{diag}(y)\alpha + \mathbf{1}^T\alpha \\ \text{subject to} & 0 \leq \alpha \leq C \end{array}$$

- This means that the data only appears in the dual through the gram matrix

$$K = X^T X$$

  which is called the **kernel** matrix.

- In particular, the original dimension n **does not appear in the dual.** SVM complexity only grows with the number of samples.
- In particular, the x_i are allowed to be infinite dimensional.
- The only requirement on K is that $K \succeq 0$.

Approximation problems

Norm approximation

$$\text{minimize} \quad \|Ax - b\|$$

($A \in \mathbb{R}^{m \times n}$ with $m \geq n$, $\| \cdot \|$ is a norm on $\mathbb{R}^m$)

interpretations of solution $x^\star = \mathrm{argmin}_x \|Ax - b\|$:

- **geometric**: $Ax^\star$ is point in $\mathcal{R}(A)$ closest to b
- **estimation**: linear measurement model

  $$y = Ax + v$$

  y are measurements, x is unknown, v is measurement error

  given $y = b$, best guess of x is $x^\star$
- **optimal design**: x are design variables (input), Ax is result (output)

  $x^\star$ is design that best approximates desired result b

examples

- least-squares approximation ($\|\cdot\|_2$): solution satisfies normal equations

$$A^T A x = A^T b$$

  ($x^\star = (A^T A)^{-1} A^T b$ if $\mathbf{Rank}\, A = n$)

- Chebyshev approximation ($\|\cdot\|_\infty$): can be solved as an LP

$$\begin{array}{ll}\text{minimize} & t\\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1}\end{array}$$

- sum of absolute residuals approximation ($\|\cdot\|_1$): can be solved as an LP

$$\begin{array}{ll}\text{minimize} & \mathbf{1}^T y\\ \text{subject to} & -y \preceq Ax - b \preceq y\end{array}$$

Penalty function approximation

$$\begin{array}{ll} \text{minimize} & \phi(r_1) + \cdots + \phi(r_m) \\ \text{subject to} & r = Ax - b \end{array}$$

($A \in \mathbb{R}^{m\times n}$, $\phi : \mathbb{R} \to \mathbb{R}$ is a convex penalty function)

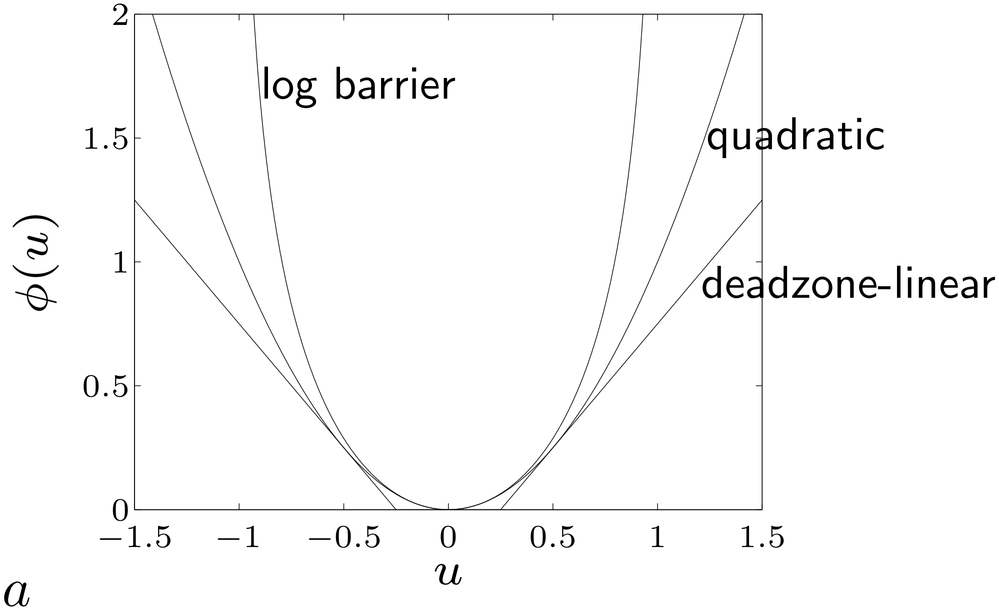
examples

- quadratic: $\phi(u) = u^2$
- deadzone-linear with width a:

$$\phi(u) = \max\{0, |u| - a\}$$

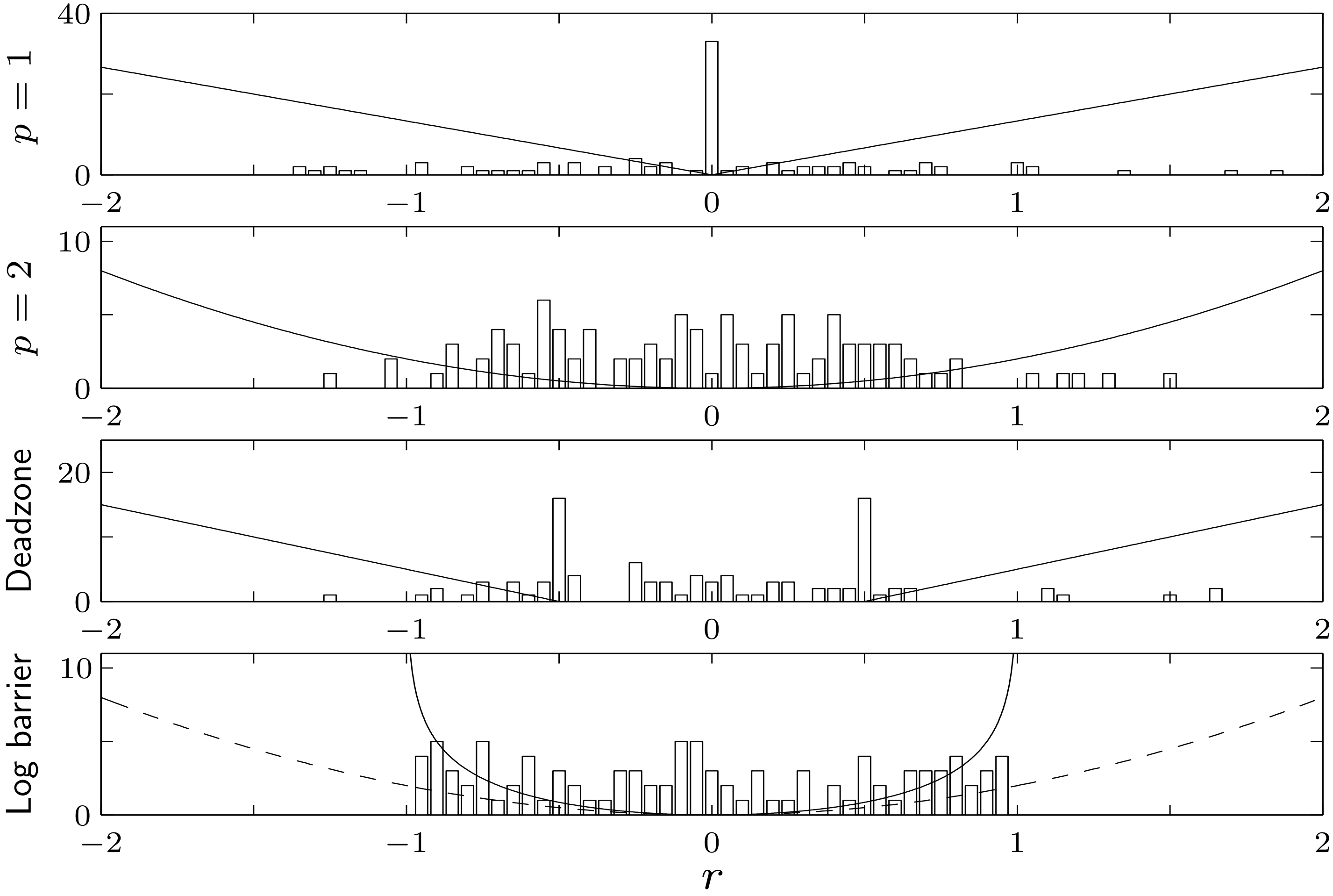
- log-barrier with limit a:

$$\phi(u) = \begin{cases} -a^2 \log(1 - (u/a)^2) & |u| < a \\ \infty & \text{otherwise} \end{cases}$$



example ($m = 100$, $n = 30$): histogram of residuals for penalties

$$\phi(u) = |u|, \quad \phi(u) = u^2, \quad \phi(u) = \max\{0, |u| - a\}, \quad \phi(u) = -\log(1 - u^2)$$

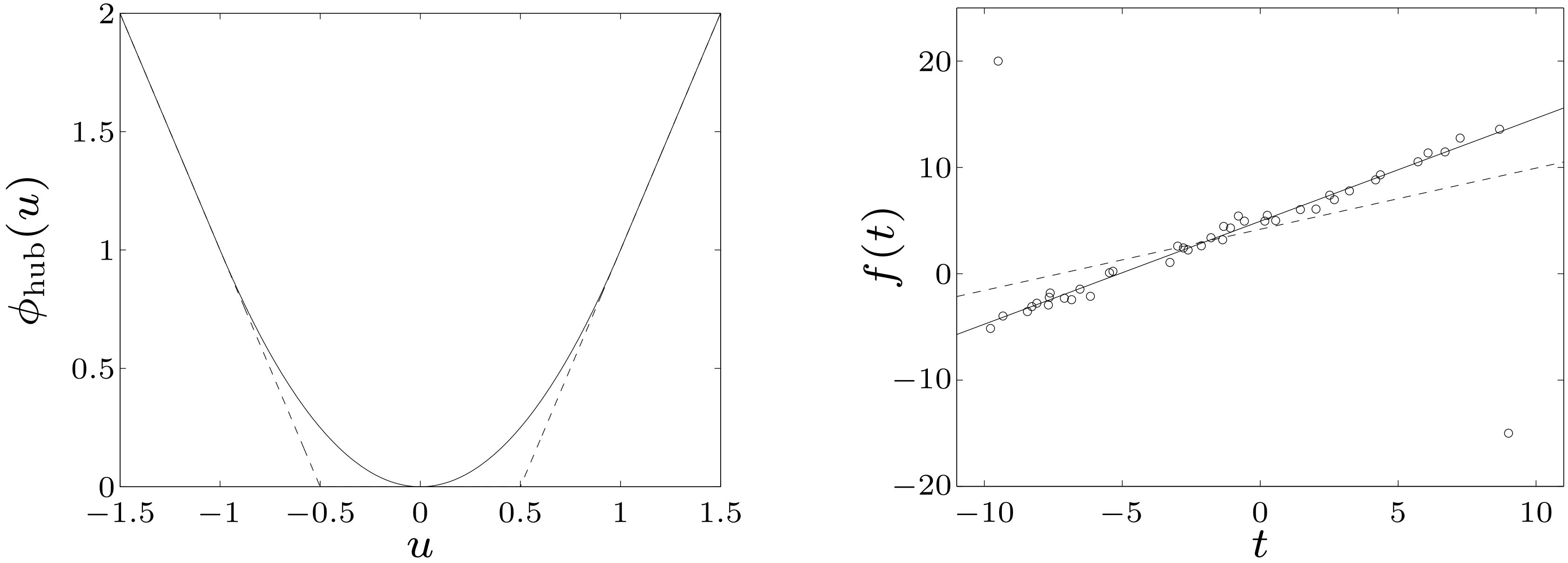


shape of penalty function has large effect on distribution of residuals

Huber penalty function (with parameter M)

$$\phi_{\text{hub}}(u) = \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M \end{cases}$$

linear growth for large u makes approximation less sensitive to outliers



- left: Huber penalty for $M = 1$
- right: affine function $f(t) = \alpha + \beta t$ fitted to 42 points t_i, y_i (circles) using quadratic (dashed) and Huber (solid) penalty

Distance matrices

Distance matrices . . .

- The problem of reconstructing an N-point Euclidean metric, given **partial** information on pairwise distances between points $v_i,\ i = 1, \ldots, N$ can also be cast as an SDP, known as and **Euclidean Distance Matrix Completion** problem.

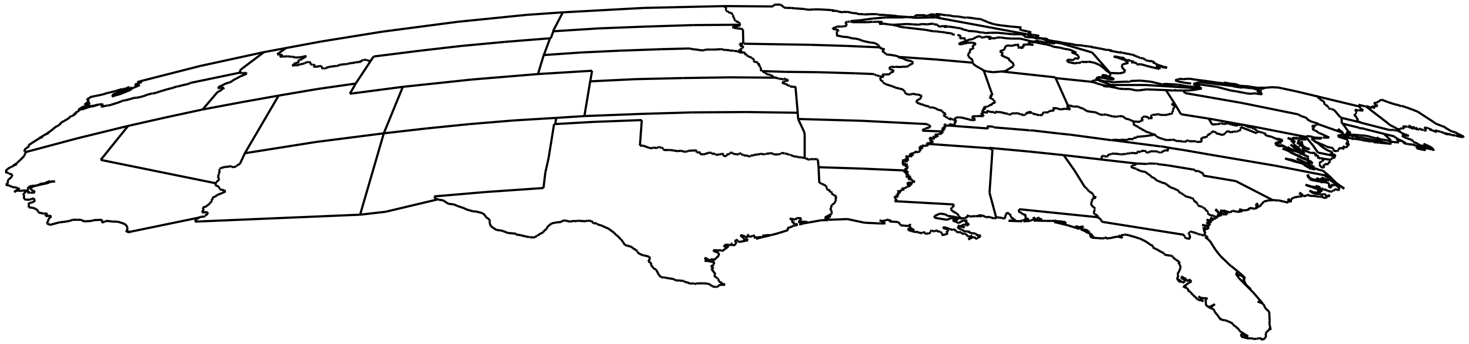
$$\begin{array}{ll} \text{find} & D \\ \text{subject to} & \mathbf{1}v^T + v\mathbf{1}^T - D \succeq 0 \\ & D_{ij} = \|v_i - v_j\|_2^2, \quad (i,j) \in S \\ & v \geq 0 \end{array}$$

  in the variables $D \in \mathbf{S}_n$ and $v \in \mathbb{R}^n$, on a subset $S \subset [1, N]^2$.

- We can add further constraints to this problem given additional structural info on the configuration.

- Applications in sensor networks, molecular conformation reconstruction etc. . .

Distance matrices . . .

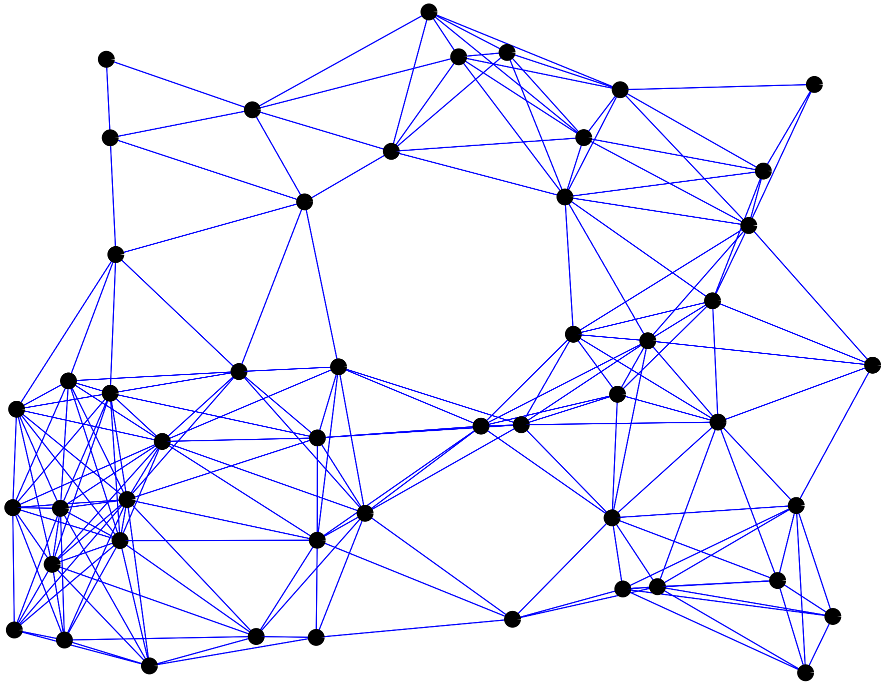


[Dattorro, 2005] 3D map of the USA reconstructed from pairwise distances on 5000 points. Distances reconstructed from Latitude/Longitude data.

Mixing rates for Markov chains & maximum variance unfolding

Mixing rates for Markov chains & unfolding

- Let $G = (V, E)$ be an **undirected graph** with n vertices and m edges.
- We define a **Markov chain** on this graph, and let $w_{ij} \geq 0$ be the transition rate for edge $(i, j) \in V$.



Mixing rates for Markov chains & unfolding

- Let $\pi(t)$ be the state distribution at time t, its evolution is governed by the heat equation

  $$d\pi(t) = -L\pi(t)dt$$

  with

  $$L_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j,\ (i,j) \in V \\ 0 & \text{if } (i,j) \notin V \\ \sum_{(i,k)\in V} w_{ik} & \text{if } i = j \end{cases}$$

  the **graph Laplacian** matrix, which means

  $$\pi(t) = e^{-Lt}\pi(0).$$

Mixing rates for Markov chains & unfolding

[Sun, Boyd, Xiao, and Diaconis, 2006]

- Maximizing the mixing rate of the Markov chain means solving

$$\begin{array}{ll}\text{maximize} & t\\ \text{subject to} & L(w) \succeq t(\mathbf{I}-(1/n)\mathbf{1}\mathbf{1}^T)\\ & \sum_{(i,j)\in V} d_{ij}^2 w_{ij} \leq 1\\ & w \geq 0\end{array}$$

  in the variable $w \in \mathbb{R}^m$, with (normalization) parameters $d_{ij}^2 \geq 0$.

- Since $L(w)$ is an affine function of the variable $w \in \mathbb{R}^m$, this is a **semidefinite program** in $w \in \mathbb{R}^m$.

Mixing rates for Markov chains & unfolding

[Weinberger and Saul, 2006, Sun et al., 2006]

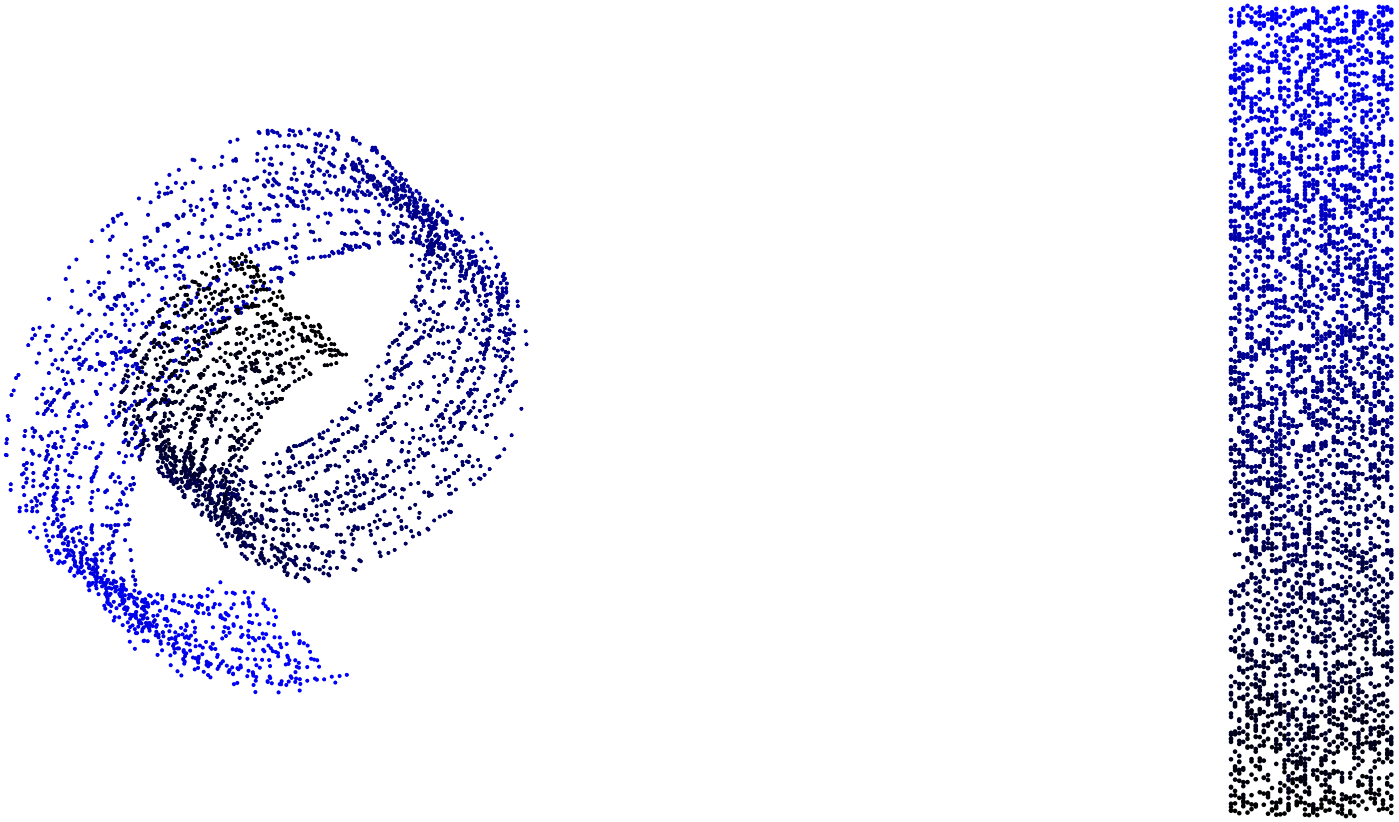
- The **dual** means solving

$$\begin{array}{ll}\text{maximize} & \mathbf{Tr}(X(\mathbf{I}-(1/n)\mathbf{1}\mathbf{1}^T))\\ \text{subject to} & X_{ii}-2X_{ij}+X_{jj}\leq d_{ij}^2, \quad (i,j)\in V\\ & X\succeq 0,\end{array}$$

  in the variable $X \in \mathbf{S}_n$.

- This is a **maximum variance unfolding problem.**

Mixing rates for Markov chains & unfolding



From [Sun et al., 2006]: we are given pairwise 3D distances for k-nearest neighbors in the point set on the right. We plot the maximum variance point set satisfying these pairwise distance bounds on the right.

Collaborative prediction

Collaborative prediction

- Users assign **ratings** to a certain number of movies:

Users

	2		1			4				5	
	5		4				?		1		3
		3		5			2				
4			?			5		3		?	
		4		1	3				5		
			2				1	?			4
	1					5		5		4	
		2		?	5		?		4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3	?			5	
2	?		1		1						
		5			2	?		4		4	
	1		3		1	5		4		5	
1		2			4				5	?	

Movies

- Objective: make recommendations for other movies. . .

Collaborative prediction

- Infer **user preferences** and **movie features** from user ratings.
- We use a linear prediction model:

$$rating_{ij} = u_i^T v_j$$

  where u_i represents user characteristics and v_j movie features.

- This makes collaborative prediction a **matrix factorization** problem
- Overcomplete representation. . .

Collaborative prediction

- **Inputs**: a matrix of ratings $M_{ij} = \{-1, +1\}$ for $(i,j) \in S$, where S is a subset of all possible user/movies combinations.
- We look for a linear model by factorizing $M \in \mathbb{R}^{n \times m}$ as:

$$M = U^T V$$

  where $U \in \mathbb{R}^{n \times k}$ represents user characteristics and $V \in \mathbb{R}^{k \times m}$ movie features.
- **Parsimony**. . . We want k to be as small as possible.
- **Output**: a matrix $X \in \mathbb{R}^{n \times m}$ which is a low-rank approximation of the ratings matrix M.

Least-Squares

- Choose Means Squared Error as measure of discrepancy.
- Suppose S is the full set, our problem becomes:

$$\min_{\{X:\,\mathbf{Rank}(X)=k\}} \|X-M\|^2$$

- This is just a **singular value decomposition** (SVD). . .

Problem: Not true when S is not the full set (partial observations). Also, MSE not a good measure of prediction performance. . .

Soft Margin

$$\text{minimize} \quad \mathbf{Rank}(X) + c \sum_{(i,j)\in S} \max(0, 1 - X_{ij}M_{ij})$$

non-convex and numerically hard. . .

- Relaxation result in Fazel et al. [2001]: replace $\mathbf{Rank}(X)$ by its convex envelope on the spectahedron to solve:

  $$\text{minimize} \quad \|X\|_* + c \sum_{(i,j)\in S} \max(0, 1 - X_{ij}M_{ij})$$

  where $\|X\|_*$ is the **nuclear norm**, *i.e.* sum of the singular values of X.
- Srebro [2004]: This relaxation also corresponds to multiple large margin SVM classifications.

Soft Margin

- The dual of this program:

$$\begin{array}{ll}\mbox{maximize} & \sum_{ij} Y_{ij}\\ \mbox{subject to} & \|Y \odot M\|_2 \leq 1\\ & 0 \leq Y_{ij} \leq c\end{array}$$

  in the variable $Y \in \mathbb{R}^{n \times m}$, where $Y \odot M$ is the Schur (componentwise) product of Y and M and $\|Y\|_2$ the largest singular value of Y.

- This problem is **sparse**: $Y_{ij}^* = c$ for $(i,j) \in S^c$

Semidefinite Program

- How do we solve it?
- Rewrite the dual

$$\begin{array}{ll}\text{maximize} & \sum_{ij} Y_{ij}\\ \text{subject to} & \|Y \odot M\|_2 \leq 1\\ & 0 \leq Y_{ij} \leq c\end{array}$$

as:

$$\begin{array}{ll}\text{maximize} & \sum_{ij} Y_{ij}\\ \text{subject to} & \left[\begin{array}{cc} I & -(Y \odot M)\\ -(Y \odot M)^T & I \end{array}\right] \succeq 0\\ & 0 \leq Y_{ij} \leq c\end{array}$$

which is a sparse **semidefinite program** in $Y \in \mathbb{R}^{n \times m}$.

Complexity

Complexity?

- Small subset S: the dual in Y is sparse, primal (in ratings X) is **dense**.
- Interior point solvers work fine for problem sizes up to 400...
- We need to solve much larger instances.
- High precision is not necessary. . .

Applications in Statistics

Outline

- MLE problems
- Experiment Design

Parametric distribution estimation

- distribution estimation problem: estimate probability density $p(y)$ of a random variable from observed values
- parametric distribution estimation: choose from a family of densities $p_x(y)$, indexed by a parameter x

maximum likelihood estimation

$$\text{maximize (over } x\text{)} \quad \log p_x(y)$$

- y is observed value
- $l(x) = \log p_x(y)$ is called log-likelihood function
- can add constraints $x \in C$ explicitly, or define $p_x(y) = 0$ for $x \notin C$
- a convex optimization problem if $\log p_x(y)$ is concave in x for fixed y

Linear measurements with IID noise

linear measurement model

$$y_i = a_i^T x + v_i, \quad i = 1, \ldots, m$$

- $x \in \mathbb{R}^n$ is vector of unknown parameters
- v_i is IID measurement noise, with density $p(z)$
- y_i is measurement: $y \in \mathbb{R}^m$ has density $p_x(y) = \prod_{i=1}^m p(y_i - a_i^T x)$

maximum likelihood estimate: any solution x of

$$\text{maximize} \quad l(x) = \sum_{i=1}^m \log p(y_i - a_i^T x)$$

(y is observed value)

examples

- Gaussian noise $\mathcal{N}(0,\sigma^2)$: $p(z) = (2\pi\sigma^2)^{-1/2}e^{-z^2/(2\sigma^2)}$,

$$l(x) = -\frac{m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(a_i^Tx - y_i)^2$$

  ML estimate is LS solution

- Laplacian noise: $p(z) = (1/(2a))e^{-|z|/a}$,

$$l(x) = -m\log(2a) - \frac{1}{a}\sum_{i=1}^{m}|a_i^Tx - y_i|$$

  ML estimate is ℓ_1-norm solution

- uniform noise on $[-a,a]$:

$$l(x) = \begin{cases} -m\log(2a) & |a_i^Tx - y_i| \leq a, \quad i = 1,\ldots,m \\ -\infty & \text{otherwise} \end{cases}$$

  ML estimate is any x with $|a_i^Tx - y_i| \leq a$

Logistic regression

random variable $y \in \{0,1\}$ with distribution

$$p = \mathbf{Prob}(y=1) = \frac{\exp(a^T u + b)}{1 + \exp(a^T u + b)}$$

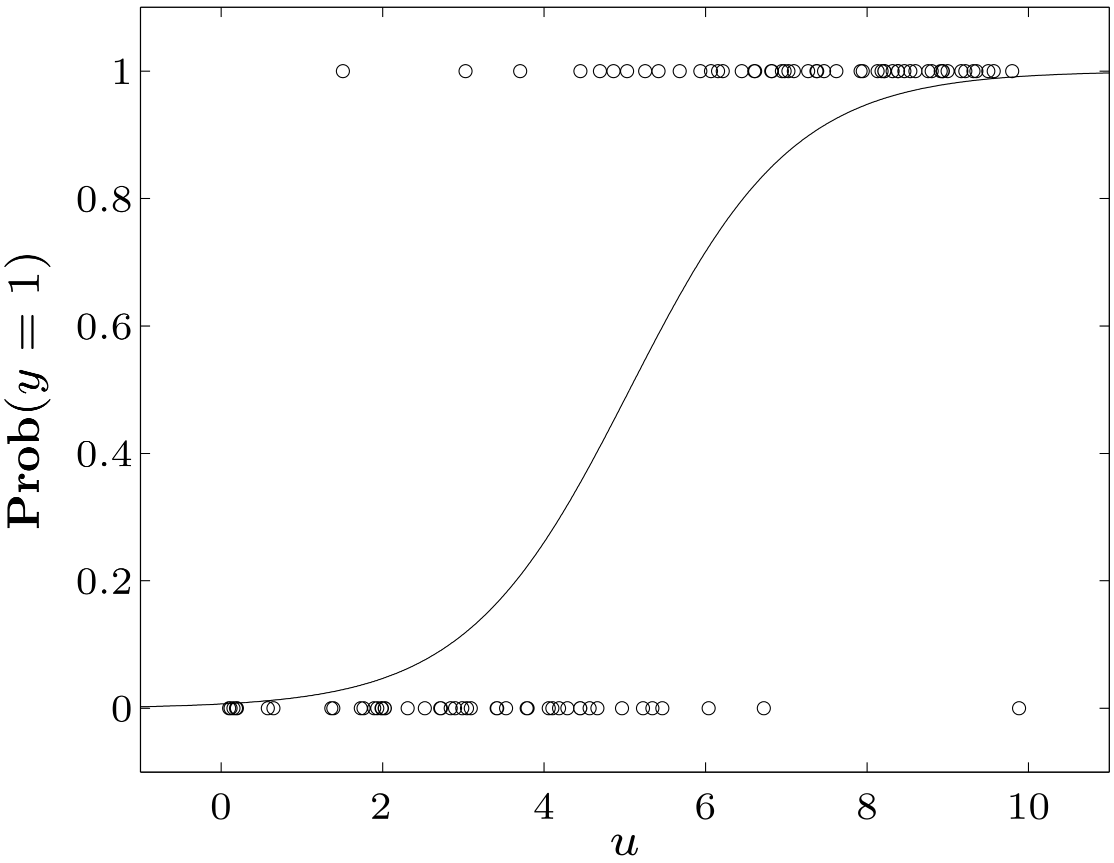
- a, b are parameters; $u \in \mathbb{R}^n$ are (observable) explanatory variables
- estimation problem: estimate a, b from m observations (u_i, y_i)

log-likelihood function (for $y_1 = \cdots = y_k = 1$, $y_{k+1} = \cdots = y_m = 0$):

$$\begin{aligned} l(a,b) &= \log\left(\prod_{i=1}^{k} \frac{\exp(a^T u_i + b)}{1+\exp(a^T u_i + b)} \prod_{i=k+1}^{m} \frac{1}{1+\exp(a^T u_i + b)}\right) \\ &= \sum_{i=1}^{k}(a^T u_i + b) - \sum_{i=1}^{m} \log(1 + \exp(a^T u_i + b)) \end{aligned}$$

concave in a, b

example ($n = 1$, $m = 50$ measurements)



- circles show 50 points (u_i, y_i)
- solid curve is ML estimate of $p = \exp(au + b)/(1 + \exp(au + b))$

Experiment design

m linear measurements $y_i = a_i^T x + w_i$, $i = 1, \ldots, m$ of unknown $x \in \mathbb{R}^n$

- measurement errors w_i are IID $\mathcal{N}(0, 1)$
- ML (least-squares) estimate is

$$\hat{x} = \left(\sum_{i=1}^m a_i a_i^T \right)^{-1} \sum_{i=1}^m y_i a_i$$

- error $e = \hat{x} - x$ has zero mean and covariance

$$E = \mathbf{E}\, ee^T = \left(\sum_{i=1}^m a_i a_i^T \right)^{-1}$$

confidence ellipsoids are given by $\{x \mid (x - \hat{x})^T E^{-1} (x - \hat{x}) \leq \beta\}$

experiment design: choose $a_i \in \{v_1, \ldots, v_p\}$ (a set of possible test vectors) to make E 'small'

vector optimization formulation

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{S}_+^n) & E = \left(\sum_{k=1}^p m_k v_k v_k^T\right)^{-1} \\ \text{subject to} & m_k \geq 0, \quad m_1 + \cdots + m_p = m \\ & m_k \in \mathbf{Z} \end{array}$$

- variables are m_k (# vectors a_i equal to v_k)
- difficult in general, due to integer constraint

relaxed experiment design

assume $m \gg p$, use $\lambda_k = m_k/m$ as (continuous) real variable

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{S}_+^n) & E = (1/m)\left(\sum_{k=1}^p \lambda_k v_k v_k^T\right)^{-1} \\ \text{subject to} & \lambda \succeq 0, \quad \mathbf{1}^T\lambda = 1 \end{array}$$

- common scalarizations: minimize $\log\det E$, $\mathbf{Tr}\, E$, $\lambda_{\max}(E)$, . . .
- can add other convex constraints, *e.g.*, bound experiment cost $c^T\lambda \leq B$

Experiment design

D-optimal design

$$\begin{array}{ll} \text{minimize} & \log\det\left(\sum_{k=1}^{p}\lambda_k v_k v_k^T\right)^{-1} \\ \text{subject to} & \lambda \succeq 0, \quad \mathbf{1}^T\lambda = 1 \end{array}$$

interpretation: minimizes volume of confidence ellipsoids

dual problem

$$\begin{array}{ll} \text{maximize} & \log\det W + n\log n \\ \text{subject to} & v_k^T W v_k \leq 1, \quad k = 1, \ldots, p \end{array}$$

interpretation: $\{x \mid x^T W x \leq 1\}$ is minimum volume ellipsoid centered at origin, that includes all test vectors v_k

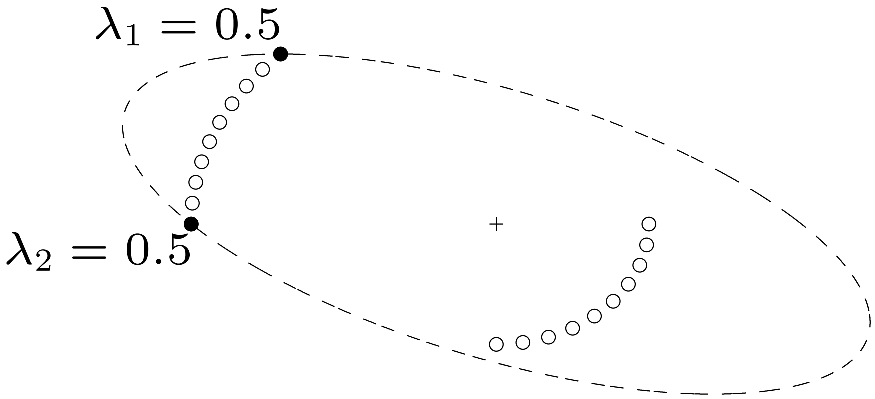
complementary slackness: for λ, W primal and dual optimal

$$\lambda_k(1 - v_k^T W v_k) = 0, \quad k = 1, \ldots, p$$

optimal experiment uses vectors v_k on boundary of ellipsoid defined by W

Experiment design

example $(p = 20)$



design uses two vectors, on boundary of ellipse defined by optimal W

Experiment design

Derivation of dual.

first reformulate primal problem with new variable X

$$\begin{array}{ll} \text{minimize} & \log\det X^{-1} \\ \text{subject to} & X = \sum_{k=1}^{p} \lambda_k v_k v_k^T, \quad \lambda \succeq 0, \quad \mathbf{1}^T \lambda = 1 \end{array}$$

$$L(X, \lambda, Z, z, \nu) = \log\det X^{-1} + \mathbf{Tr}\left(Z \left(X - \sum_{k=1}^{p} \lambda_k v_k v_k^T \right) \right) - z^T \lambda + \nu(\mathbf{1}^T \lambda - 1)$$

- minimize over X by setting gradient to zero: $-X^{-1} + Z = 0$
- minimum over λ_k is $-\infty$ unless $-v_k^T Z v_k - z_k + \nu = 0$

Dual problem

$$\begin{array}{ll} \text{maximize} & n + \log\det Z - \nu \\ \text{subject to} & v_k^T Z v_k \leq \nu, \quad k = 1, \ldots, p \end{array}$$

change variable $W = Z/\nu$, and optimize over ν to get dual of page 84.

References

J. Dattorro. *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA, 2005.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. *Proceedings American Control Conference*, 6:4734–4739, 2001.

N. Srebro. *Learning with Matrix Factorization*. PhD thesis, Massachusetts Institute of Technology, 2004.

J. Sun, S. Boyd, L. Xiao, and P. Diaconis. The fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Review*, 48(4):681–699, 2006.

K.Q. Weinberger and L.K. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.