Advanced Computational Methods in Statistics: Lecture 2 Optimisation

Axel Gandy

Department of Mathematics
Imperial College London
http://www2.imperial.ac.uk/~agandy

London Taught Course Centre
for PhD Students in the Mathematical Sciences
Autumn 2014

Lecture 2 - Optimisation

Part 1 Deterministic Optimisation

Part 2 EM algorithm

Part 3 LASSO and LARS

Part 4 NP-complete Problems

Part 5 Stochastic Approximation

Part I

Deterministic Optimisation

Introduction

Local Search Methods

Comments

Simulation study

Introduction

- $f : A \to \mathbb{R}$, $A \subset \mathbb{R}^d$.
- Goal: Find $\mathbf{x}^* \in \mathbb{R}^d$ such that

$$f(\mathbf{x}^*) = \min_{\mathbf{x} \in A} f(\mathbf{x})$$

- Example: finding the maximum likelihood estimator.
- Can have side conditions:
  $g : A \to \mathbb{R}^q$ some function. Want to

$$\text{minimise}_{\mathbf{x} \in A} f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) = 0$$

- Explicit solutions: Lagrange Multipliers.
  With inequality constraints: Kuhn-Tucker conditions.

Local Search Methods - No Side Conditions

- Main idea: create a sequence $x_0, x_1, x_2, \dots$ approximations to x^*. Hopefully $x_n \to x^*$ as $n \to \infty$.
- Choice of algorithm depends on how many derivatives of f are available. Some Examples:

  no derivatives: Nelder-Mead: works with $d+1$ points that move towards x^* and then contract around it.

  gradient ∇f: Gradient descent:

  $$x_n = x_{n-1} - \epsilon_n \nabla f(x_{n-1})$$

  other methods: conjugate gradient, ...

  gradient ∇f+Hessian H: Newton's Method:

  $$x_n = x_{n-1} - H(f, x_{n-1})^{-1} \nabla f(x_{n-1})$$

  Typically: the more derivatives are available the better the convergence rate.
- Global convergence only guaranteed if f is convex.
- If global convergence cannot be guaranteed, the very least one should do is use several starting values.

Optimisation with Side Conditions

$$\text{minimise}_{\mathbf{x}\in A} f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) = 0$$

- f linear, g linear: "linear programming", Simplex algorithm
- f quadratic, g linear: quadratic programming
- more general structure:
  sequential quadratic programming algorithms may work:
  idea: approximate the problem locally by a quadratic programming problem.
  (implemented e.g. in the NAG library)
- More heuristic approach: put side condition into objective function, i.e. minimise $f(x) + \lambda(g(x))^2$ for some *large* $\lambda > 0$.

Comments

- Optimisation (in particular with side conditions and non-convex) can be a tough problem
- Local search algorithms are not the only algorithms - many more approaches (simulated annealing, random optimisation, genetic optimisation)
- Many solutions have been developed that work well for specific problems.
- Try to use implementation of algorithms written by experts!
- Useful resource: Decision Tree for Optimisation Software `http://plato.asu.edu/guide.html`

Fortran

Simulation study of various optimization algorithms

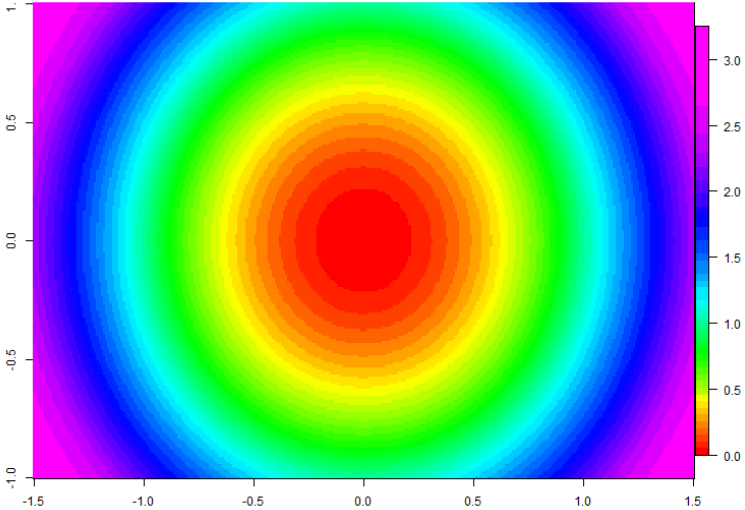
Various algorithms implemented in *optim()* in R:

- Nelder-Mead: a simplex-based method.
- BFGS: quasi-Newton method (BroydenFletcherGoldfarbShanno method)
- CG: a conjugate gradient method.
- L-BFGS-B: an algorithm that would allow bounds on the parameters.
- Simulated annealing with default settings.
- Simulated annealing with more steps and slower cooling.

Applied to 3 functions.

Example 1 - quadratic function

$$f : \mathbb{R}^2 \to \mathbb{R}, f(x,y) = x^2 + y^2$$



(global minimum at (0,0))

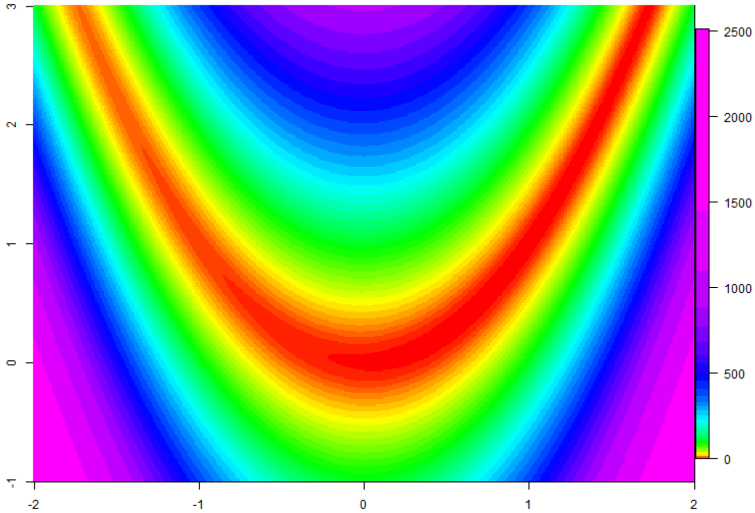
Applying standard R algorithms to the quadratic function

	N-M	BFGS	CG	L-BFGS-B	SANN1	SANN2
Conv	1	1	1	1	1	1
0%	2.73e-09	7.24e-28	7.42e-15	1.53e-41	9e-07	1.01e-07
25%	1.14e-07	1.61e-24	2.13e-14	3.96e-40	4.19e-05	2.47e-06
50%	2.46e-07	1.55e-23	3.76e-14	7.69e-40	8.21e-05	7.54e-06
75%	5.56e-07	7.11e-23	5.27e-14	1.37e-39	0.0002	1.4e-05
100%	5.04e-06	1.86e-21	8.91e-13	2.72e-39	0.000896	4.84e-05
neval	65.8	9.66	21.8	4.24	1e+04	1e+05

Table: Started from 100 different starting points in [-10,10]x[-10,10]. Conv=Proportion of successful convergence indicated; Quantiles of f(minimizer); neval=average number of function evaluations.

Example 2 - Rosenbrock Banana function

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$



(global minimum at (1,1))

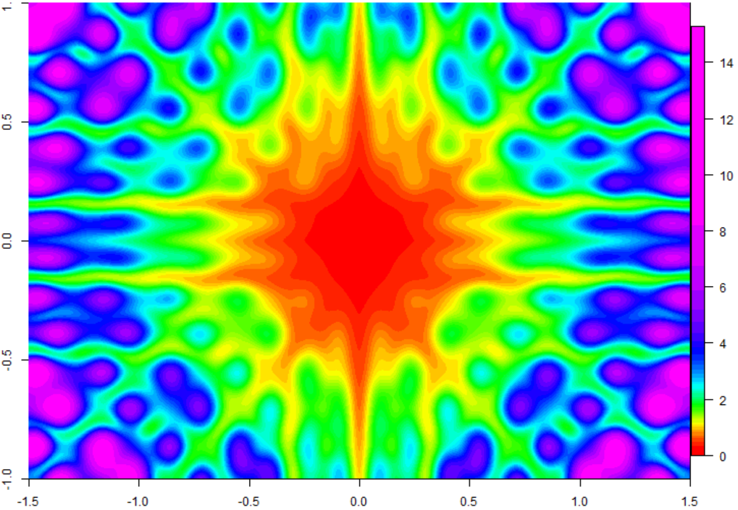
Applying standard R algorithms to the Banana function

	N-M	BFGS	CG	L-BFGS-B	SANN1	SANN2
Conv	1	0.85	0.01	0.99	1	1
0%	4.32e-08	9.93e-11	0.000187	1.98e-10	1.76e-06	8.25e-07
25%	3.8e-05	2.93e-08	0.0765	3.03e-08	0.000194	9.81e-06
50%	0.000747	3.95e-08	0.203	3.99e-08	0.000444	2.18e-05
75%	0.0489	4e-08	3.66	4e-08	0.00105	4.58e-05
100%	1e+06	1e+06	1e+06	1e+06	2.36	0.000199
neval	129	111	253	54	1e+04	1e+05

Table: Started from 100 different starting points in [-10,10]x[-10,10]. Conv=Proportion of successful convergence indicated; Quantiles of f(minimizer); neval=average number of function evaluations.

Example 3

$f : \mathbb{R}^2 \to \mathbb{R}, f(x,y) = (x\sin(20y) + y\sin(20x))^2 \cosh(\sin(10x)x) + (x\cos(10y) - y\sin(10x))^2 \cosh(\cos(20y)y) + |x| + |y|$



(global minimum at (0,0))

Applying standard R algorithms to Example 3

	N-M	BFGS	CG	L-BFGS-B	SANN1	SANN2
Conv	0.99	1	0.21	0.78	1	1
0%	2.4e-08	3.59e-20	4.57e-10	1.35e-14	0.00125	0.000241
25%	6.69	3.06e-18	2.79	8.25	0.0123	0.00148
50%	9.74	6.75	7.99	11.5	6.01	0.003
75%	13.4	11.9	11.7	29.4	10.9	0.00427
100%	181	263	269	178	22.7	18.4
neval	103	65.1	413	41.2	1e+04	1e+05

Table: Started from 100 different starting points in [-10,10]x[-10,10]. Conv=Proportion of successful convergence indicated; Quantiles of f(minimizer); neval=average number of function evaluations.

Comments

- Functions that are “nice” (no local minima, maybe even convex): standard numerical algorithms work best, the more derivatives are used the better.
- Functions with local minima: Need to add noise to avoid getting trapped (needs tuning)
- General advice:
  - Use several starting values
  - Plot function (if possible)

Part II

The EM Algorithm

Introduction

Example - Mixtures

Theoretical Properties

EM Algorithm - Introduction

- Expectation-Maximisation algorithm; two steps:
    - E-step
    - M-Step
- General-purpose algorithm for maximum likelihood estimation in incomplete data problems.
- Main reference: Dempster et al. (1977)
- According to `scholar.google.com`: cited $>$ 14000 times! (narrowly beating e.g. Cox "Regression Models and Life Tables" with roughly 13500 citations)
  [citation count on 19/1/2009]
- Most of the material in this chapter is based on McLachlan & Krishnan (2008). An overview article is Ng et al. (2004).

Situations in which the EM algorithm is applicable

- Incomplete data situations such as
  - missing data
  - truncated distributions
  - censored or grouped observations
- Statistical models such as
  - random effects
  - mixtures
  - convolutions
  - latent class/variable structures
  - ...
- Even if a problem appears not to be an incomplete data problem - writing it as such a problem can sometimes simplify its analysis (by simplifying the likelihood).

The EM algorithm - Notation

$\mathbf{y}$	observed data, incomplete data (corresponding r.v.: $\mathbf{Y}$)
$g(\cdot, \psi)$	density of $\mathbf{Y}$
ψ	unknown parameter vector

Likelihood $L(\psi) := g(\mathbf{y}, \psi)$.
Want to find the MLE, i.e. maximise L.

$\mathbf{z}$	missing data (corresponding r.v.: $\mathbf{Z}$)
$\mathbf{x} = (\mathbf{y}, \mathbf{z})$	complete data (corresponding r.v.: $\mathbf{X}$)
$g_c(\cdot; \psi)$	density of $\mathbf{X}$

Note: $\boxed{g(\mathbf{y}, \psi) = \mathrm{E}[g_c(\mathbf{Y}, \mathbf{Z}; \psi) | \mathbf{Y} = \mathbf{y}]}$

The EM-algorithm

- Let ψ^0 be some initial value for ψ.
- For $k = 0, 1, \dots$

  E-step Calculate $Q(\psi, \psi^k)$, where

  $$Q(\psi, \psi^k) = \mathrm{E}[\log g_c(\mathbf{X}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]$$

  M-step

  $$\psi^{k+1} = \mathrm{argmax}_{\psi}\, Q(\psi, \psi^k)$$

- Employ some convergence criterion (e.g. based on $\log g_c(\mathbf{x}; \psi^k)$)

Note:

$$Q(\psi, \psi^k) = \int \log g_c(\mathbf{y}, \mathbf{z}; \psi) k(\mathbf{z}|\mathbf{y}; \psi) d\mathbf{z},$$

where $k(\mathbf{z}|\mathbf{y}; \psi) = g_c(\mathbf{y}, \mathbf{z}; \psi)/g(\mathbf{y}; \psi)$ is the conditional density of $\mathbf{z}$ given $\mathbf{Y} = \mathbf{y}$.

The EM-algorithm

- Let ψ^0 be some initial value for ψ.
- For $k = 0, 1, \dots$

  E-step Calculate $Q(\psi, \psi^k)$, where

  $$Q(\psi, \psi^k) = \mathrm{E}[\log g_c(\mathbf{X}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]$$

  M-step

  $$\psi^{k+1} = \operatorname{argmax}_{\psi} Q(\psi, \psi^k)$$

- Employ some convergence criterion (e.g. based on $\log g_c(\mathbf{x}; \psi^k)$)

Note:

$$Q(\psi, \psi^k) = \int \log g_c(\mathbf{y}, \mathbf{z}; \psi) k(\mathbf{z}|\mathbf{y}; \psi) d\mathbf{z},$$

where $k(\mathbf{z}|\mathbf{y}; \psi) = g_c(\mathbf{y}, \mathbf{z}; \psi)/g(\mathbf{y}; \psi)$ is the conditional density of $\mathbf{z}$ given $\mathbf{Y} = \mathbf{y}$.

Monotonicity of the EM Algorithm

- Then $\log g(\mathbf{y}; \psi) = \log(g_c(\mathbf{x}; \psi)) - \log k(\mathbf{x}|\mathbf{y}; \psi)$.
- Take expectations with density $k(\mathbf{x}|\mathbf{y}; \psi)$

$$\log g(\mathbf{y}; \psi) = Q(\psi, \psi^k) - \underbrace{\mathrm{E}[\log k(\mathbf{X}|\mathbf{y}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]}_{=: H(\psi, \psi^k)}$$

- Thus

$$\log g(\mathbf{y}; \psi^{k+1}) - \log g(\mathbf{y}; \psi^k) =$$
$$= \underbrace{(Q(\psi^{k+1}, \psi^k) - Q(\psi^k, \psi^k))}_{\geq 0 \text{ (Def EM)}} + \underbrace{(H(\psi^k, \psi^k) - H(\psi^{k+1}, \psi^k))}_{\geq 0 \text{ (next slide)}}$$

- Hence, $\log g(\mathbf{y}; \psi^k) \nearrow$ as $k \to \infty$.

Monotonicity of the EM Algorithm

- Then $\log g(\mathbf{y}; \psi) = \log(g_c(\mathbf{x}; \psi)) - \log k(\mathbf{x}|\mathbf{y}; \psi)$.
- Take expectations with density $k(\mathbf{x}|\mathbf{y}; \psi)$

$$\log g(\mathbf{y}; \psi) = Q(\psi, \psi^k) - \underbrace{\mathrm{E}[\log k(\mathbf{X}|\mathbf{y}; \psi)|\mathbf{Y} = \mathbf{y}; \psi^k]}_{=:H(\psi,\psi^k)}$$

- Thus

$$\begin{aligned}&\log g(\mathbf{y}; \psi^{k+1}) - \log g(\mathbf{y}; \psi^k) = \\ &= \underbrace{(Q(\psi^{k+1}, \psi^k) - Q(\psi^k, \psi^k))}_{\geq 0 \text{ (Def EM)}} + \underbrace{(H(\psi^k, \psi^k) - H(\psi^{k+1}, \psi^k))}_{\geq 0 \text{ (next slide)}}\end{aligned}$$

- Hence, $\log g(\mathbf{y}; \psi^k) \nearrow$ as $k \to \infty$.

$\mathbf{y}$ observed, $\mathbf{z}$ missing, $\mathbf{x} = (\mathbf{y}, \mathbf{z})$, $Q(\psi, \psi^k) = \mathrm{E}[\log g_c(\mathbf{X}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]$
g density of $\mathbf{y}$, g_c density of $\mathbf{x}$, $k = g_c/g$ density of $\mathbf{z}|\mathbf{y}$

Monotonicity of the EM Algorithm (cont)

- $H(\psi, \psi^k) = \mathrm{E}[\log k(\mathbf{X}|\mathbf{y}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]$ is maximised at $\psi = \psi^k$.
- Indeed,

$$
\begin{aligned}
H(\psi^k, \psi^k) - H(\psi, \psi^k) &= \mathrm{E}[-\log \frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} | \mathbf{Y} = \mathbf{y}; \psi^k] \\
&\geq -\log \mathrm{E}[\frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} | \mathbf{Y} = \mathbf{y}; \psi^k] \quad \text{(Jensen's inequality)} \\
&= -\log \int \frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} k(\mathbf{X}|\mathbf{y}; \psi^k) dx \\
&= -\log \int k(\mathbf{x}|\mathbf{y}; \psi) d\mathbf{x} = -\log(1) = 0
\end{aligned}
$$

- Thus $H(\psi^k, \psi^k) - H(\psi^{k+1}, \psi^k) \geq 0$.

Monotonicity of the EM Algorithm (cont)

- $H(\psi, \psi^k) = \mathrm{E}[\log k(\mathbf{X}|\mathbf{y}; \psi) | \mathbf{Y} = \mathbf{y}; \psi^k]$ is maximised at $\psi = \psi^k$.
- Indeed,

$$\begin{aligned}
H(\psi^k, \psi^k) - H(\psi, \psi^k) &= \mathrm{E}[-\log \frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} | \mathbf{Y} = \mathbf{y}; \psi^k] \\
&\geq -\log \mathrm{E}[\frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} | \mathbf{Y} = \mathbf{y}; \psi^k] \quad \text{(Jensen's inequality)} \\
&= -\log \int \frac{k(\mathbf{X}|\mathbf{y}; \psi)}{k(\mathbf{X}|\mathbf{y}; \psi^k)} k(\mathbf{X}|\mathbf{y}; \psi^k) dx \\
&= -\log \int k(\mathbf{x}|\mathbf{y}; \psi) d\mathbf{x} = -\log(1) = 0
\end{aligned}$$

- Thus $H(\psi^k, \psi^k) - H(\psi^{k+1}, \psi^k) \geq 0$.

The inqeuality for h H is a special form of the following general inequality:
Let X be a r.v. with density g. Let f be any other density. Then

$$E[\log(f(X))] \leq E[\log(g(X))]$$

Proof: Jensen's inequality.

"Information inequality"

Generalised EM algorithm(GEM)

- The M-step may not have a close-form solution.
- It may not be feasible to find a global maximum of $Q(\cdot, \psi^k)$
- Replace M-step by:

choose ψ^{k+1} such that

$$Q(\psi^{k+1}, \psi^k) \geq Q(\psi^k, \psi^k)$$

Mixture Distribution

- Consider a mixture distribution
  - $\psi_1, \dots, \psi_d \geq 0$, mixing proportions, $\sum_{i=1}^d \psi_i = 1$.
  - $f_1, \dots, f_d$ component densities.

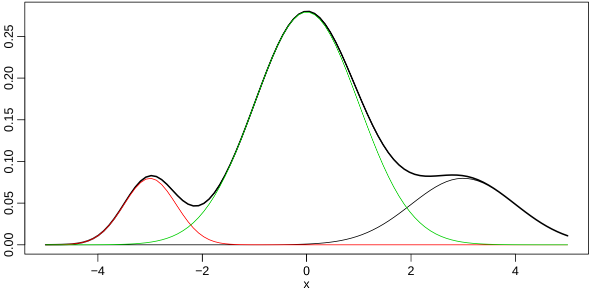
  With probability ψ_i sample from f_i.
  Resulting density:

$$f(x) = \sum_{i=1}^{d} \psi_i f_i(x)$$

- We will assume that $f_1, \dots, f_d$ are known, but that $\boldsymbol{\psi} = (\psi_1, \dots, \psi_d)$ is unknown.

Mixture of Normals

- $d = 3$
- $f_1 =$ pdf of $N(3, 1)$
- $f_2 =$ pdf of $N(-3, 0.5)$
- $f_3 =$ pdf of $N(0, 1)$
- $\psi = (0.2, 0.1, 0.7)$



Mixture Distributions (cont.)

- Let $Y_1, \dots, Y_n$ be an iid sample from the mixture distribution.
- The likelihood of the incomplete data is

$$g(\mathbf{y};\psi) = \prod_{i=1}^{n} \sum_{j=1}^{d} \psi_j f_j(y_i)$$

- Missing data: Z_{ij} indicator variables of chosen component
- Complete density:

$$g_c(\mathbf{y},\mathbf{z};\psi) = \prod_{i=1}^{n} \prod_{j=1}^{d} (\psi_j f_j(y_i))^{z_{ij}}$$

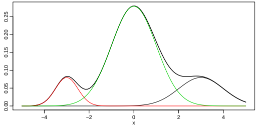
Hence, the log-likelihood for the full data is

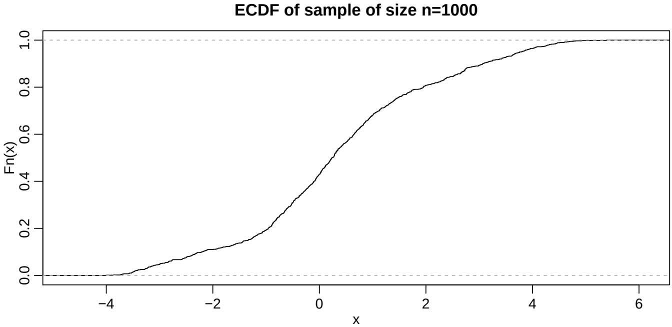
$$\log g_c(\mathbf{y},\mathbf{z};\psi) = \sum_{i=1}^{n} \sum_{j=1}^{d} z_{ij} \log(\psi_j) + C,$$

where C does not depend on ψ.

Mixture of Normals (cont.)

A sample





Mixture Distributions (cont.)

$$Q(\psi, \psi^k) = \mathrm{E}[\log g_c(\mathbf{y}, \mathbf{z}; \psi); \mathbf{y}, \psi^k] = \sum_{i=1}^{n} \sum_{j=1}^{d} \log(\psi_j)\, \mathrm{E}[z_{ij}; \mathbf{y}, \psi^k] + C,$$

where

$$\mathrm{E}[z_{ij}; \mathbf{y}, \psi^k] = \frac{\psi_j^k f_j(y_i)}{\sum_\nu \psi_\nu^k f_\nu(y_i)} =: a_{ij}$$

We want to maximise

$$Q(\psi, \psi^k) = \sum_{j=1}^{d} (\sum_{i=1}^{n} a_{ij}) \log(\psi_j)$$

subject to $\sum \psi_j = 1$. Using e.g. Lagrange multipliers and $\sum_j a_{ij} = 1$ one can see that the optimum is at

$$\psi_j^{k+1} = \frac{1}{n} \sum_{i=1}^{n} a_{ij}, \quad j = 1, \dots, d$$

Note: a_{ij} depends on ψ^k

Mixture Distributions (cont.)

$$Q(\boldsymbol{\psi},\boldsymbol{\psi}^k) = \mathrm{E}[\log g_c(\mathbf{y},\mathbf{z};\boldsymbol{\psi});\mathbf{y},\boldsymbol{\psi}^k] = \sum_{i=1}^{n}\sum_{j=1}^{d}\log(\psi_j)\,\mathrm{E}[z_{ij};\mathbf{y},\boldsymbol{\psi}^k] + C,$$

where

$$\mathrm{E}[z_{ij};\mathbf{y},\boldsymbol{\psi}^k] = \frac{\psi_j^k f_j(y_i)}{\sum_\nu \psi_\nu^k f_\nu(y_i)} =: a_{ij}$$

We want to maximise

$$Q(\boldsymbol{\psi},\boldsymbol{\psi}^k) = \sum_{j=1}^{d}(\sum_{i=1}^{n} a_{ij})\log(\psi_j)$$

subject to $\sum \psi_j = 1$. Using e.g. Lagrange multipliers and $\sum_j a_{ij} = 1$ one can see that the optimum is at

$$\psi_j^{k+1} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad j = 1,\dots,d$$

Note: a_{ij} depends on ψ^k

Mixture Distributions (cont.)

$$Q(\boldsymbol{\psi},\boldsymbol{\psi}^k) = \mathrm{E}[\log g_c(\mathbf{y},\mathbf{z};\boldsymbol{\psi});\mathbf{y},\boldsymbol{\psi}^k] = \sum_{i=1}^{n}\sum_{j=1}^{d}\log(\psi_j)\,\mathrm{E}[z_{ij};\mathbf{y},\boldsymbol{\psi}^k] + C,$$

where

$$\mathrm{E}[z_{ij};\mathbf{y},\boldsymbol{\psi}^k] = \frac{\psi_j^k f_j(y_i)}{\sum_\nu \psi_\nu^k f_\nu(y_i)} =: a_{ij}$$

We want to maximise

$$Q(\boldsymbol{\psi},\boldsymbol{\psi}^k) = \sum_{j=1}^{d}(\sum_{i=1}^{n} a_{ij})\log(\psi_j)$$

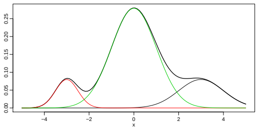
subject to $\sum \psi_j = 1$. Using e.g. Lagrange multipliers and $\sum_j a_{ij} = 1$ one can see that the optimum is at

$$\boldsymbol{\psi}_j^{k+1} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad j = 1,\dots,d$$

Note: a_{ij} depends on $\boldsymbol{\psi}^k$

Mixture of Normals

Applying the EM algorithm



k	ψ_1^k	ψ_2^k	ψ_3^k
1	0.333	0.333	0.333
2	0.261	0.115	0.624
3	0.225	0.097	0.678
4	0.216	0.094	0.69
5	0.214	0.094	0.692
6	0.213	0.094	0.693
7	0.213	0.094	0.693
8	0.213	0.094	0.693
9	0.213	0.094	0.693
10	0.213	0.094	0.693

Convergence Results

- We have already seen that $L(\psi^k)$ is increasing in k.
- Thus, if L is bounded from above, $L(\psi^k)$ converges to some L^*.
- In almost all applications, L^* is a stationary value, i.e. $L^* = L(\psi^*)$ for some ψ^* such that

$$\frac{\partial L(\psi)}{\partial \psi}\Big|_{\psi=\psi^*} = \mathbf{0}$$

- Want L^* to be a global maximum.
- However, general theorems will only guarantee that L^* is a stationary point or a local maximum.
- There are some theorems that ensure convergence to a global maximum (assuming unimodality of L).
- Main reference for convergence results: Wu (1983). (see also McLachlan & Krishnan (2008))

EM-Algorithm - Some Warnings

- There are (pathological?) examples, where the (Generalised) EM-algorithm does not work as expected, e.g. where there may
  - convergence to a saddle point,
  - convergence to a local MINIMUM,
  - $L(\psi^k)$ converges, but ψ^k does not.

  (see (McLachlan & Krishnan, 2008, Section 3.6))
- Don't trust the output of the EM result blindly!
  The very least you can do is try using different starting values.

Comments

- If the E-step cannot be computed analytically then Monte-Carlo techniques can be used. The resulting algorithm is often called "MCEM" algorithm.
  MCMC techniques (e.g. Gibbs sampling) can come into play here.
- For an overview of theoretical work concerning the convergence rate of the EM-algorithm see (McLachlan & Krishnan, 2008, Chapter 4).

Part III

LASSO and related algorithms

LASSO

Penalised Regression

LARS algorithm

Comments

Ordinary least squares (OLS)

- Linear Model:

$$\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$\mathbf{Y}$	vector of responses (n-dimensional)
$X \in \mathbb{R}^{n\times p}$	matrix of covariates
$\boldsymbol{\beta} \in \mathbb{R}^p$	vector of regression coefficients (unknown)
$\boldsymbol{\epsilon}$	vector of errors, $\mathrm{E}\,\boldsymbol{\epsilon} = \mathbf{0}$, $\mathrm{Cov}\,\boldsymbol{\epsilon} = \sigma^2 I_n$

- Classical approach (if $n > p$):
  $\boldsymbol{\beta}$ is chosen as minimiser of the Sum of squares

$$\underline{\hat{\beta} = (X^T X)^{-1} X^T y}$$

$$S(\boldsymbol{\beta}) = \|\mathbf{Y} - X\boldsymbol{\beta}\|^2 = \sum_{i=1}^{n} (Y_i - (X\boldsymbol{\beta})_i)^2,$$

  where $\|\mathbf{a}\|^2 = \sum_i a_i^2$.
- Many modern datasets (e.g. microarrays):
  high-dimensional covariates, even $n << p$ (large p small n)
  $\Rightarrow \hat{\boldsymbol{\beta}}$ is not uniquely identified!

Lasso

Lasso ('least absolute shrinkage and selection operator') (Tibshirani, 1996)

$\hat{\beta}$ solution of

$$\begin{cases} \|\mathbf{Y} - X\boldsymbol{\beta}\|_2^2 \to \min \\ \sum_{i=1}^{d} |\beta_i| \leq c \end{cases}$$

where $c \in \mathbb{R}$ is a constant.

Remark:

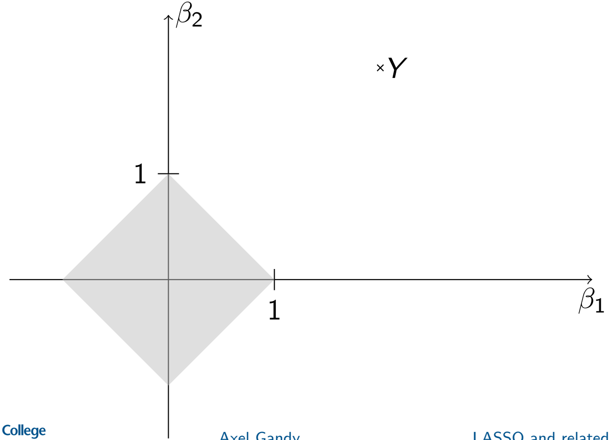
Instead of side condition, can use L_1-penalty

$$\|\mathbf{Y} - X\boldsymbol{\beta}\|^2 + \lambda \sum_{i=1}^{d} |\beta_i| \to \min$$

with a constant $\lambda > 0$.

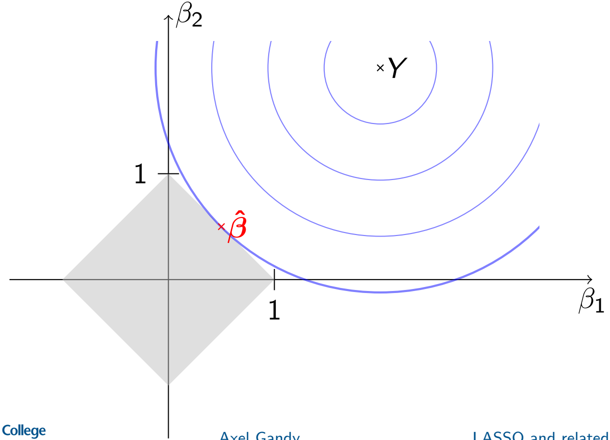
Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \min \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



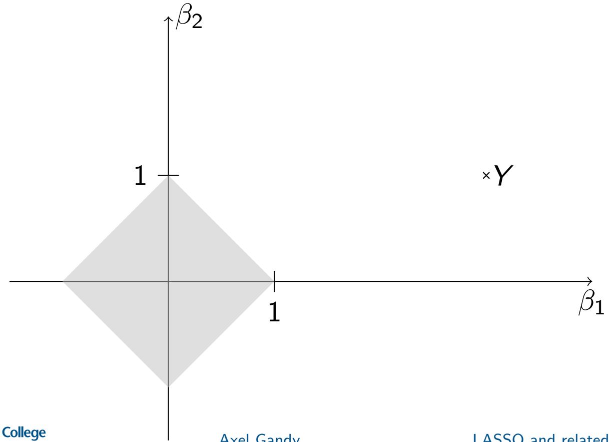
Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \min \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



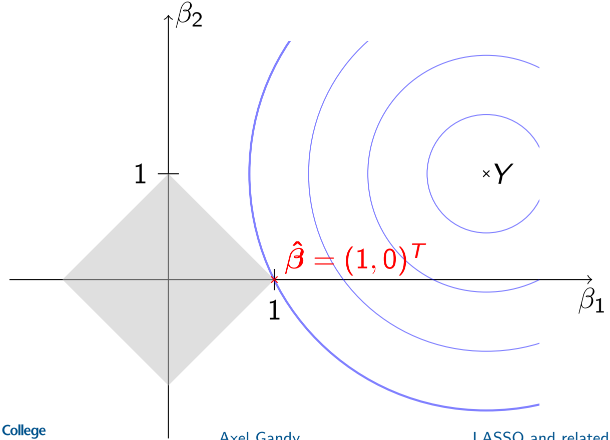
Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \min \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



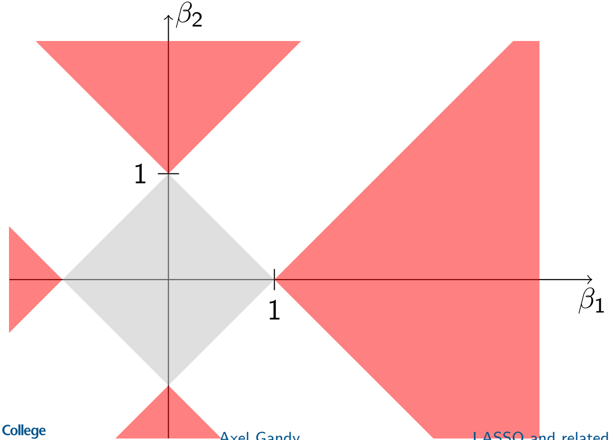
Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \min \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

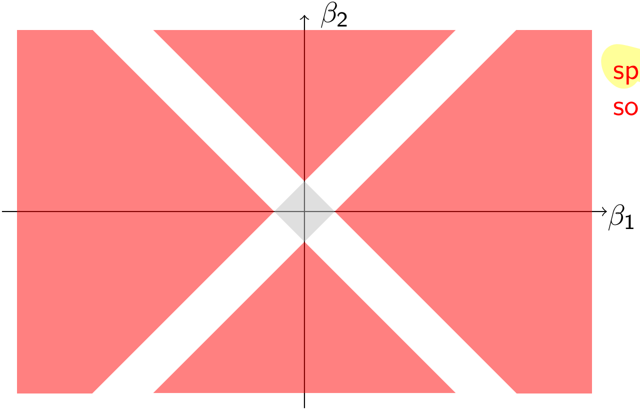
$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \min \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



sparse solutions

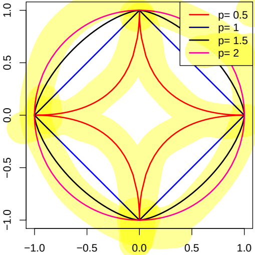
Example: $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$. Using $c = 1$,

$$\begin{cases} (Y_1 - \beta_1)^2 + (Y_2 - \beta_2)^2 \to \text{min} \\ |\beta_1| + |\beta_2| \leq 1 \end{cases}$$



sparse solutions

Penalised Regression



add regularity conditions on $\boldsymbol{\beta}$:

$$p(\boldsymbol{\beta}) \leq t \quad \text{for a constant } t$$

Examples:

- $p(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_0 = \#\{i : \beta_i \neq 0\}$ (best subset selection)
- $p(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1 = \sum_{i=1}^p |\beta_i|$ (LASSO, 'least absolute shrinkage and selection operator', see Tibshirani (1996))
- $p(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^p |\beta_i|^2$ (ridge regression)
- Bridge Regression - families of penalties, e.g.:
  - $p_d(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^{\cancel{d}\,\rho} = \sum_{i=1}^p |\beta_i|^{\cancel{d}\,\rho}$ where $0 \leq \underset{\rho}{\cancel{d}} \leq 2$
  - elastic net

Thus overall:

$$S(\boldsymbol{\beta}) \to \min \text{ subject to } p(\boldsymbol{\beta}) \leq t$$

Alternatively: For some constant λ,

$$S(\boldsymbol{\beta}) + \lambda p(\boldsymbol{\beta}) \to \min$$

Finding the Solution of Penalised Regression

closed form solution exists

- Best subset regression: NP hard problem
- Convex optimisation problem for e.g. LASSO, Ridge → standard optimisation techniques could be used to find a solution.
- LARS/LASSO algorithm: faster algorithm for $p(\beta) = \sum_{j=1}^{p} |\beta_j|$.
- How to choose the threshold t (or λ)? Use cross-validation.

Least Angle Regression

- Introduced in Efron et al. (2004).
- Efficient stepwise algorithm.
- LASSO modification of the LARS algorithm: generates LASSO solutions for all thresholds t.

Assumptions

Will assume that

- response has mean 0, i.e.

$$\sum_{i=1}^{n} Y_i = 0$$

- covariates have mean 0 and length 1, i.e.

$$\sum_{i=1}^{n} X_{ij} = 0 \text{ and } \sum_{i=1}^{n} X_{ij}^2 = 1 \text{ for } j = 1, \dots, p$$

LARS algorithm

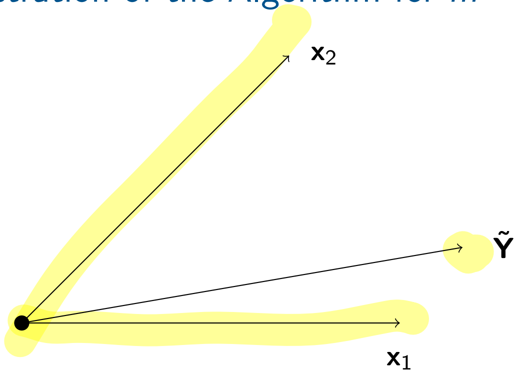
Least Angle Regression (Efron et al., 2004)

A rough description:

Let $\mathbf{x}_1, \dots, \mathbf{x}_p$ be the predictors, i.e. the columns of X.

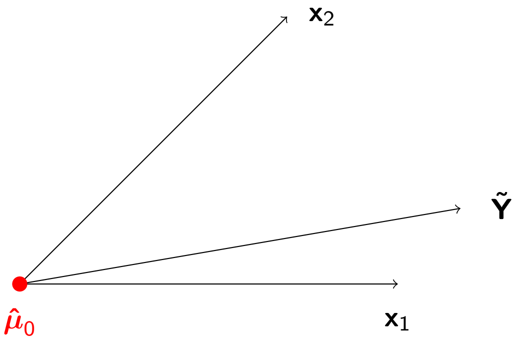
- Start with all coefficient vectors equal to 0, i.e. $\beta_1 = 0, \dots, \beta_p = 0$
- Let $\mathcal{A}$ be the set of covariates that are most correlated with the current residual (initially the residual is the response).
- Initially, $\mathcal{A} = \{\mathbf{x}_{j_1}\}$.
- take the largest step possible in the direction of $\mathbf{x}_{j_1}$ until another predictor $\mathbf{x}_{j_2}$ enters $\mathcal{A}$
- continue in the direction equiangular between $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ until a third predictor $\mathbf{x}_{j_3}$ enters $\mathcal{A}$
- continue in the direction equiangular between $\mathbf{x}_{j_1}$, $\mathbf{x}_{j_2}$, $\mathbf{x}_{j_3}$ until a fourth predictor $\mathbf{x}_{j4}$ enters the most correlated set
- ...

Illustration of the Algorithm for $m = 2$ Covariates



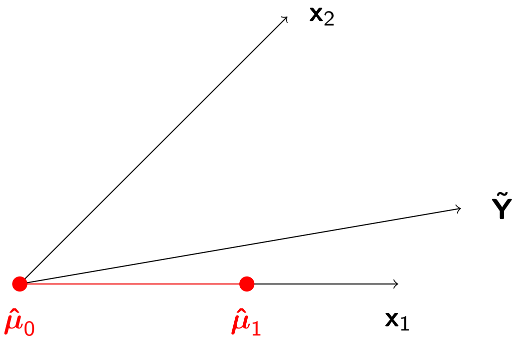
- $\tilde{\mathbf{Y}}$ projection of $\mathbf{Y}$ onto the plane spanned by $\mathbf{x}_1, \mathbf{x}_2$.
- $\hat{\mu}_j$ estimate after j-th step.

Illustration of the Algorithm for $m = 2$ Covariates



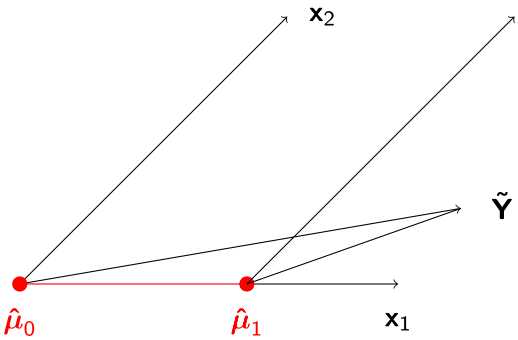
- $\tilde{\mathbf{Y}}$ projection of $\mathbf{Y}$ onto the plane spanned by $\mathbf{x}_1, \mathbf{x}_2$.
- $\hat{\mu}_j$ estimate after j-th step.

Illustration of the Algorithm for $m = 2$ Covariates



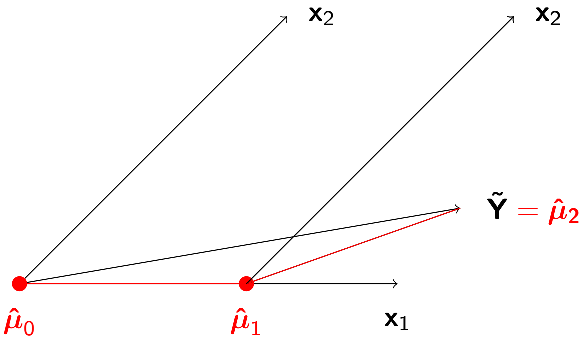
- $\tilde{\mathbf{Y}}$ projection of $\mathbf{Y}$ onto the plane spanned by $\mathbf{x}_1, \mathbf{x}_2$.
- $\hat{\mu}_j$ estimate after j-th step.

Illustration of the Algorithm for $m = 2$ Covariates



- $\tilde{\mathbf{Y}}$ projection of $\mathbf{Y}$ onto the plane spanned by $\mathbf{x}_1, \mathbf{x}_2$.
- $\hat{\mu}_j$ estimate after j-th step.

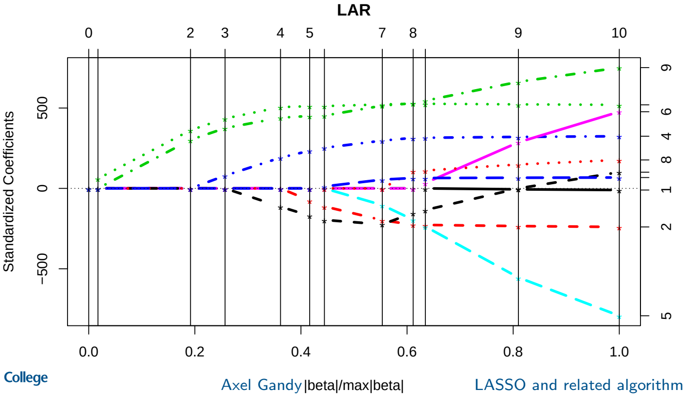
Illustration of the Algorithm for $m = 2$ Covariates


$\mathbf{x}_2$
$\mathbf{x}_2$
$\tilde{\mathbf{Y}} = \hat{\boldsymbol{\mu}}_2$
$\hat{\boldsymbol{\mu}}_0$
$\hat{\boldsymbol{\mu}}_1$
$\mathbf{x}_1$

- $\tilde{\mathbf{Y}}$ projection of $\mathbf{Y}$ onto the plane spanned by $\mathbf{x}_1, \mathbf{x}_2$.
- $\hat{\mu}_j$ estimate after j-th step.

LARS - Diabetes Data

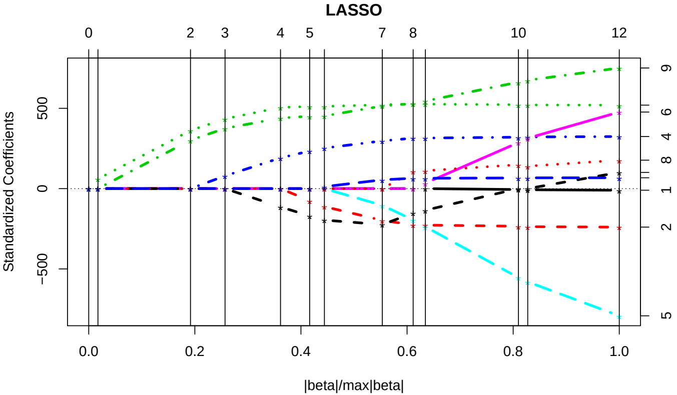
- from Efron et al. (2004)
- 442 patients
- covariates: age, sex, BMI, blood pressure, 6 blood serum measurements
- Response: “a measure of disease progression”



LASSO Modification of the LARS Algorithm

- LARS algorithm needs to be modified to yield all LASSO solutions
- essentially a modification is needed when a β_j crosses 0.

LASSO - Diabetes Data



Note: now 12 steps instead of 10 with the LARS algorithms

Comments

- R-package: *lars*
- A LASSO fit has no more than $n-1$ (centred) predictors with nonzero coefficient
- Number of operations needed:

  $p < n$: $O(p^3 + np^2)$
  $p > n$: $O(n^3 + n^2 p)$
- Other algorithm: coordinate descent

Further recent approaches

- Group Lasso

$$\|\mathbf{Y} - X\boldsymbol{\beta}\|^2 + \lambda \sum_j \left(\sum_{\nu \in K_j} |\beta_\nu|^2 \right)^{1/2} \to \text{min}$$

  where K_j are disjoint groups of variables and $\lambda > 0$.

- Fused Lasso

$$\|\mathbf{Y} - X\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{(i,j) \in A} |\beta_i - \beta_j| \to \text{min}$$

  where $A \subset \{1, \dots, n\}^2$ and $\lambda_1, \lambda_2 > 0$.

- Recent "hot" topics: compressed sensing, matrix completion, stability selection.

Part IV

NP complete problems

NP-complete Problems I

- Concerns decision problems
  - Input: 0-1 sequence of length n
  - Output: “yes” or “no”
- P= class of all decision problems that can be solved in at most *polynomial* time in n (on a Turing machine)
- NP is the set of decision problems for which a solution can be verified in polynomical time with some additional input of polynomial size.

  As a consequence: all problems in NP can be solved in *exponential* time.
- A decision problem is NP-complete if any other decision problem in NP can be reduced to it in polynomial time.

NP-complete Problems II

- There is a large number of NP-complete problems, e.g.
  - Travelling Salesman Problem
    Phrased as decision problem:
    Let x be some fixed length. Is there a roundtrip for the salesman of length $\leq x$?
  - Best subset regression: (phrased as decision problem)
  - ....

  (see `http://en.wikipedia.org/wiki/List_of_NP-complete_problems` for a long list)
- It is not clear if $P \neq NP$. This is one of the Millennium Prize Problems with a \$1,000,000 prize, see `http://www.claymath.org/millennium/P_vs_NP/`

Part V

Stochastic Approximation

The Robbins-Monro Algorithm

Example

Stochastic Approximation

Robbins-Monro/Kiefer-Wolfowitz algorithm

- Want to minimise $z(\boldsymbol{\theta})$ over $\boldsymbol{\Theta} \subset \mathbb{R}^d$
  e.g.: $z(\boldsymbol{\theta}) = \mathrm{E}(f(X, \boldsymbol{\theta}))$, where X is a random vector with known distribution and f is a known function.
- Iterative algorithm: successive approximations $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots$
- Standard approach - Gradient Descent:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \epsilon_n \nabla z(\boldsymbol{\theta}_n)$$

  for some deterministic sequence ϵ_n.
- Assume that we cannot evaluate $\nabla z(\boldsymbol{\theta})$ directly.
- Available $\mathbf{Y}_n$ "close to" $\nabla z(\boldsymbol{\theta})$.
  In the Robbins-Monro-algorithm, see Robbins & Monro (1951), one assumes

$$\mathbf{Y}_n = \nabla z(\theta) + \epsilon$$

  with $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$.
  Iteration:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \epsilon_n \mathbf{Y}_{n+1},$$

How to choose ϵ_n?

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \epsilon_n \mathbf{Y}_{n+1}$$

Requirements on ϵ_n

- To be able to reach any point:

$$\sum_{n=0}^{\infty} \epsilon_n = \infty$$

(assuming $\mathrm{E}\,\mathbf{Y}_n$ is bounded)

- To get convergence of $\boldsymbol{\theta}_n$, need

$$\epsilon_n \to 0$$

(assuming $\mathrm{Var}(Y_n) \not\to 0$):

Canonical choice: $\epsilon_n = an^{-\delta}$ for some $0 < \delta \leq 1$ and some $a > 0$.

How can one obtain Y_n?

Some options for $z(\boldsymbol{\theta}) = \mathrm{E}(f(\mathbf{X}, \boldsymbol{\theta}))$:

- finite differences (Kiefer-Wolfowitz algorithm, Kiefer & Wolfowitz (1952)): Let $M(\theta)$ be such that $\mathrm{E}(M(\theta)) = z(\theta)$

$$Y_{n,i} = \frac{M(\boldsymbol{\theta} + c_n) - M(\boldsymbol{\theta} - c_n)}{2c_n}$$

- Infinitesimal Perturbation Analysis (IPA)
  Main Idea: often $\frac{\partial}{\partial\theta} z(\boldsymbol{\theta}) = \frac{\partial}{\partial\theta} \mathrm{E}(f(\mathbf{X}, \boldsymbol{\theta})) = \mathrm{E}(\frac{\partial}{\partial\theta} f(\mathbf{X}, \boldsymbol{\theta}))$.
  Define Y_n as Monte Carlo estimate of the RHS:

$$Y_n = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial\theta} f(\mathbf{X}^i, \boldsymbol{\theta})$$

where $\mathbf{X}, \mathbf{X}^1, \dots, \mathbf{X}^m$ is iid.

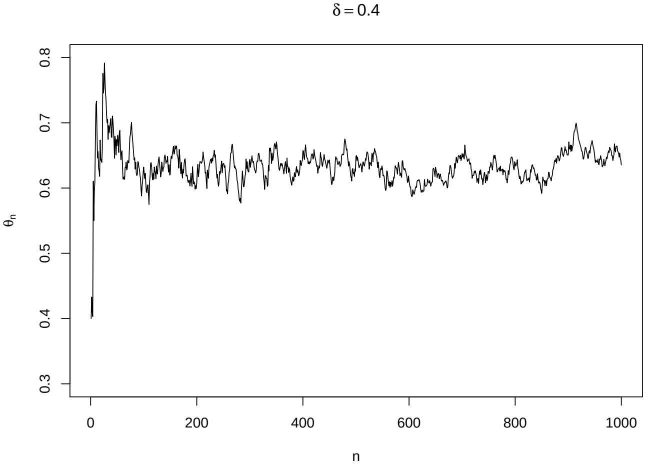
Stochastic-Approximation - Example

based on (Asmussen & Glynn, 2007, Section VIII 5a)

- Minimise
$$z(\theta) = \mathrm{E}[\max(\theta X_1 + X_2, (1-\theta)X_3)],$$
where $X_i \sim \mathit{Gamma}(2, 2/i), i = 1, \dots, 3$ are independent.
(the correct minimiser is 0.625)
- Estimate $z'(\boldsymbol{\theta}_n)$ by MC simulation:
Note $z'(\boldsymbol{\theta}) = \mathrm{E}[g(X_1, X_2, X_3, \theta)]$, where
$$g(x_1, x_2, x_3, \theta) = \begin{cases} x_1 & \theta x_1 + x_2 \geq (1-\theta)x_3 \\ -x_3 & \text{otherwise} \end{cases}$$
Use the estimator
$$Y_n = \frac{1}{m}\sum_{i=1}^{m} g(X_1^i, X_2^i, X_3^i, \theta)$$
where $X_j^i \sim X_j$, $j = 1, \dots, 3$, $i = 1, \dots, m$ are independent

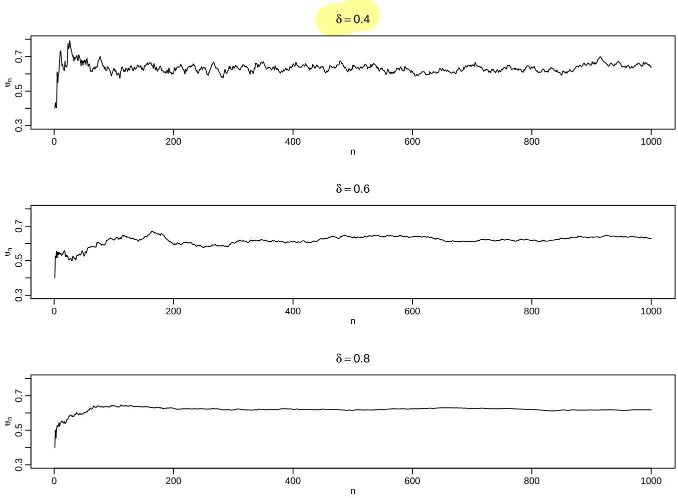
Stochastic-Approximation - one run

$m = 10$, $\epsilon_n = n^{-\delta}/10$, $\theta_0 = 0.4$



Stochastic-Approximation - Sensitivity to θ

Same parameters as before



Stochastic Approximation - Comments

- Very general class of algorithms - related to stochastic control.
- Several Parameters need tuning (best done on a case by case basis)
  - How many samples m to take at each step? Should m depend n?
  - What ϵ_n to use?
- A lot of theoretical work has been concerned with establishing theoretical properties of these algorithms.
  Main idea:
  - Relate the sequence the sequence θ_n to the solution $\theta(t)$ of the deterministic dynamical system

    $$\frac{\partial}{\partial t}\boldsymbol{\theta}(t) = -\nabla z(\boldsymbol{\theta}(t))$$

    and use martingale theory to analyse the differences.
    See e.g. Kushner & Yin (2003) for details.
- A shorter introduction can be found in e.g. Asmussen & Glynn (2007).

Part VI

Appendix

Topics in the coming lectures:

- MCMC methods
- Bootstrap
- Particle Filtering

References I

Asmussen, S. & Glynn, P. W. (2007). *Stochastic Simulation - Algorithms and Analysis*. Springer.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39**, 1–38.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32**, 407–451.

Kiefer, J. & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* **23**, 462–466.

Kushner, H. J. & Yin, G. (2003). *Stochastic Approximation and Recursive Algorithms and Recursive Algorithms and Applications*. Springer.

McLachlan, G. J. & Krishnan, T. (2008). *The EM Algorithm and Extensions*. Second ed., Wiley.

Ng, S. K., Krishnan, T. & McLachlan, G. J. (2004). The EM algorithm. In *Computational Statistics* (eds. J. Gentle, W. Härdle & Y. Mori), chap. II.5, 137–168, Springer.

References II

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* **22**, 400–407.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58**, 267–288.

Wu, C. F. J. (1983). On the convergence properties of the em algorithm. *The Annals of Statistics* **11**, 95–103.