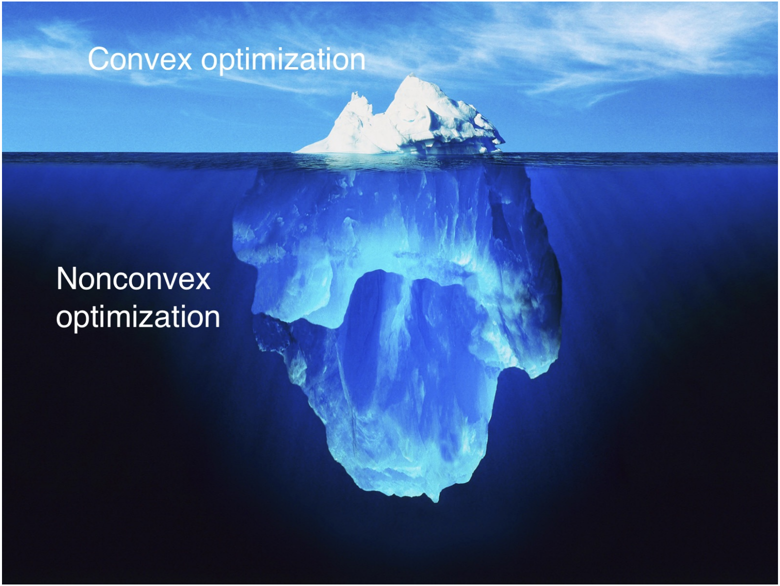
Nonconvex? NP!

(No problem!)

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

Beyond the tip?



Some takeaway points

- If possible, formulate task in terms of convex optimization — typically easier to solve, easier to analyze
- Nonconvex does not necessarily mean nonscientific! However, statistically, it does typically mean high(er) variance
- In more cases than you might expect, nonconvex problems can be solved exactly (to global optimality)

What does it mean for a problem to be nonconvex?

Consider a generic convex optimization problem:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(x) \\
\text{subject to} \quad & h_i(x) \le 0, \;\; i = 1, \dots m \\
& \ell_j(x) = 0, \;\; j = 1, \dots r
\end{aligned}
$$

Here f, h_i, $i = 1, \dots m$ are convex, and ℓ_j, $j = 1, \dots r$ are affine

A nonconvex problem is one of this form, where not all of these assumptions are met on the functions

But trivial modifications of convex problems can lead to nonconvex formulations ... so we really just consider nonconvex problems that are not trivially equivalent to convex ones

What does it mean to solve a nonconvex problem?

Nonconvex problems can have local minima i.e., there can be a feasible x such that

$$f(y) \geq f(x) \quad \text{for all feasible } y \text{ such that } \|x - y\|_2 \leq R$$

but x is still not globally optimal. (Note: we proved that this could not happen for convex problems)

Hence by solving a nonconvex problem, we mean finding the global minimizer

We also implicitly mean doing it efficiently, i.e., in polynomial time

Addendum

This is really about putting together a list of cool problems, that are suprisingly tractable ... hence there will be exceptions about nonconvexity and/or requiring exact global optima

(Also, I'm sure that there are many more examples out there that I'm missing, so I invite you to contribute to the list!)

Outline

Rough categories for today's problems:

- Classical/core nonconvex problems
- Eigen problems
- Graph problems
- Nonconvex proximal operators
- Discrete problems
- Infinite-dimensional problems
- Statistical problems

Classic/core nonconvex problems

Linear-fractional programs

A linear-fractional program is of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{c^T x + d}{e^T x + f} \\ \text{subject to} \quad & Gx \le h, \; e^T x + f > 0 \\ & Ax = b \end{aligned}$$

This is nonconvex (but quasiconvex). Provided that this problem is feasible, it is in fact equivalent to the linear program

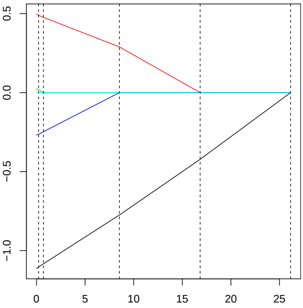
$$\begin{aligned} \min_{y \in \mathbb{R}^n, \, z \in \mathbb{R}} \quad & c^T y + dz \\ \text{subject to} \quad & Gy - hz \le 0, \; z \ge 0 \\ & Ay - bz = 0, \; e^T y + fz = 1 \end{aligned}$$

The link between the two problems is the transformation

$$y = \frac{x}{e^T x + f}, \quad z = \frac{1}{e^T x + f}$$

The proof of their equivlance is very simple; e.g., see Boyd and Vandenberghe (2004), "Convex Optimization", Chapter 4

Linear-fractional problems arise in statistics in the study of the solutions paths for many common adaptive estimation problems



E.g., the knots in the lasso path can be seen as the optimal values of linear-fractional programs

See Taylor et al. (2013), "Tests in adaptive regression via the Kac-Rice formula"

Geometric programs

A monomial is a function $f : \mathbb{R}^n_{++} \to \mathbb{R}$ of the form

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

for some $c > 0$ and $a_1, \ldots a_n \in \mathbb{R}$. A posynomial is a sum of monomials

$$f(x) = \sum_{k=1}^{p} c_k x_1^{a_{k1}} x_2^{a_{k2}} \cdots x_n^{a_{kn}}$$

The nonconvex problem

$$\begin{aligned} \min_{x>0} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \le 1, \;\; i = 1, \ldots m \\ & \ell_j(x) = 1, \;\; j = 1, \ldots r \end{aligned}$$

with f, h_i, $i = 1, \ldots m$ posynomials and h_j, $j = 1, \ldots r$ monomials is called a geometric program

This is equivalent to a convex problem, via a simple transformation. If $f(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$, then we can transform $y_i = \log x_i$, and

$$f(x) = c(e^{y_1})^{a_1}(e^{y_2})^{a_2} \cdots (e^{y_n})^{a_n} = e^{a^T y + b}$$

where $b = \log c$. Similarly, we can write any posynomial as

$$\sum_{k=1}^{p} e^{a_k^T y + b_k}$$

Hence we can express any geometric program as

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & \sum_{k=1}^{p_0} e^{a_{0k}^T y + b_{0k}} \\ \text{subject to} \quad & \sum_{k=1}^{p_i} e^{a_{ik}^T y + b_{ik}} \leq 1, \quad i = 1, \ldots m \\ & e^{g_j^T y + h_j} = 1, \quad j = 1, \ldots r \end{aligned}$$

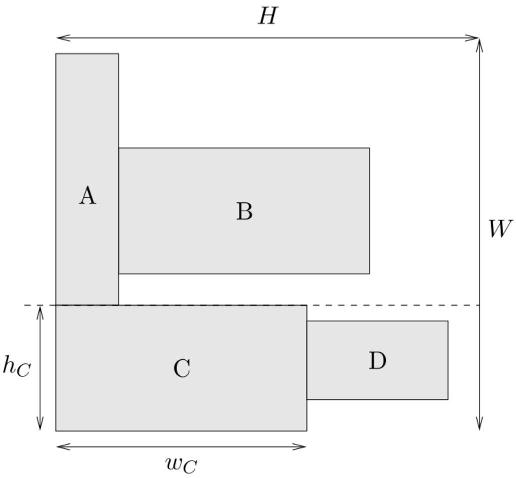
Taking logs, we recover an equivalent convex problem

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & \log\Big(\sum_{k=1}^{p_0} e^{a_{0k}^T y + b_{0k}}\Big) \\ \text{subject to} \quad & \log\Big(\sum_{k=1}^{p_i} e^{a_{ik}^T y + b_{ik}}\Big) \le 0, \;\; i = 1, \ldots m \\ & g_j^T y + h_j = 0, \;\; j = 1, \ldots r \end{aligned}$$

See Boyd and Vandenberghe (2004), Chapter 4; see also Boyd et al. (2007), "A tutorial on geometric programming"

Geometric programming for matrices: Sra and Hosseini (2013), "Geometric optimization on positive definite matrices with application to elliptically contoured distributions"

Many interesting problems are geometric programs, e.g.,



Floor planning can be done with geometric programming (see Boyd et al. (2007), see also Boyd and Vandenberghe (2004), Chapter 8)

Handling convex equality constraints

Given convex f, h_i, $i = 1, \ldots m$, the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \;\; & f(x) \\ \text{subject to} \;\; & h_i(x) \leq 0, \;\; i = 1, \ldots m \\ & \ell(x) = 0 \end{aligned}$$

is nonconvex when ℓ is convex but not affine. A convex relaxation of this problem is

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \;\; & f(x) \\ \text{subject to} \;\; & h_i(x) \leq 0, \;\; i = 1, \ldots m \\ & \ell(x) \leq 0 \end{aligned}$$

If we can ensure that $\ell(x^*) = 0$ at any solution x^* of the above problem, then the two are equivalent

See Boyd and Vandenberghe (2004), Exercises 4.6 and 4.58. E.g., consider the maximum utility problem

$$\begin{aligned} \max_{\substack{x_0,\dots x_T \in \mathbb{R} \\ a_1,\dots a_{T+1} \in \mathbb{R}}} \quad & \sum_{t=0}^{T} \alpha_t u(x_t) \\ \text{subject to} \quad & a_{t+1} = a_t + f(a_t) - c_t, \;\; t = 0, \dots T \\ & 0 \le x_t \le a_t, \;\; t = 0, \dots T \end{aligned}$$

where $a_0 \ge 0$ is fixed. Interpretation: x_t is the amount spent of your total available money a_t at time t; concave function u gives utility, concave function f measures investment return

This is not a convex problem, because of the equality constraint; but can relax to

$$a_{t+1} \le a_t + f(a_t) - c_t, \;\; t = 0, \dots T$$

without changing solution (why? think about throwing out money)

Problems with two quadratic functions

Consider the problem involving two quadratics

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \;\; & x^T A_0 x + 2b_0^T x + c_0 \\ \text{subject to} \;\; & x^T A_1 x + 2b_1^T x + c_1 \le 0 \end{aligned}$$

Here A_0, A_1 need not be positive definite, so this is nonconvex. The dual problem can be cast as

$$\begin{aligned} \max_{u\in\mathbb{R},\, v\in\mathbb{R}} \;\; & u \\ \text{subject to} \;\; & \begin{bmatrix} A_0 + vA_1 & b_0 + vb_1 \\ (b_0 + vb_1)^T & c_0 + vc_1 - u \end{bmatrix} \succeq 0 \\ & v \ge 0 \end{aligned}$$

and (as always) is convex. Furthermore, strong duality holds

See Boyd and Vandenberghe (2004), see also Beck and Eldar (2006), "Strong duality in nonconvex quadratic optimization with two quadratic constraints"

Eigen problems

Principal component analysis

Given a matrix $Y \in \mathbb{R}^{n \times p}$, consider the nonconvex problem

$$\min_{X \in \mathbb{R}^{n \times p}} \|Y - X\|_F^2 \ \text{ subject to } \ \text{rank}(X) = k$$

for some fixed k. The solution here is given by the singular value decomposition of Y: if $Y = UDV^T$, then

$$\hat{X} = U_k D_k V_k^T,$$

where U_k, V_k are the first k columns of U, V, and D_k is the first k diagonal elements of D. I.e., $\hat{X}$ is the reconstruction of Y from its first k principal components

This is often called the Eckart-Young Theorem, established in 1936, but was probably known even earlier — see Stewart (1992), "On the early history of the singular value decomposition"

Fantope

Another characterization of the SVD is via the following nonconvex problem, given a symmetric matrix $S \in \mathbb{R}^{p \times p}$:

$$\min_{Z \in \mathbb{R}^{p \times p}} \|S - Z\|_F^2 \;\; \text{subject to} \;\; \mathrm{rank}(Z) = k, \; Z \text{ is a projection}$$

The solution here is $\hat{Z} = V_k V_k^T$, where the columns of $V_k \in \mathbb{R}^{p \times k}$ give the first k eigenvectors of S

This is equivalent to a convex problem. Start by expressing the constraint set C as

$$\begin{aligned} C &= \Big\{ Z \in \mathbb{R}^{p \times p} : \mathrm{rank}(Z) = k, \; Z \text{ is a projection} \Big\} \\ &= \Big\{ Z \in \mathbb{R}^{p \times p} : Z = Z^T, \; \lambda_i(Z) \in \{0, 1\} \text{ for } i = 1, \ldots p, \\ &\qquad\qquad \mathrm{tr}(Z) = k \Big\} \end{aligned}$$

Now consider the convex hull $\mathcal{F}_k = \mathrm{conv}(C)$:

$$\mathcal{F}_k = \Big\{ Z \in \mathbb{R}^{p\times p} : Z = Z^T,\ \lambda_i(Z) \in [0,1],\ i = 1, \ldots p,\ \mathrm{tr}(Z) = k \Big\}\Big)$$
$$= \Big\{ Z \in \mathbb{R}^{p\times p} : Z = Z^T,\ 0 \preceq Z \preceq I,\ \mathrm{tr}(Z) = k \Big\}$$

This is called the Fantope of order k. Further, the convex problem

$$\min_{Z \in \mathbb{R}^{p\times p}} \|S - Z\|_F^2 \ \text{ subject to } \ Z \in \mathcal{F}_k$$

admits the same solution as the original one, i.e., $\hat{Z} = V_k V_k^T$

See Fan (1949), "On a theorem of Weyl conerning eigenvalues of linear transformations", and Overton and Womersley (1992), "On the sum of the largest eigenvalues of a symmetric matrix"

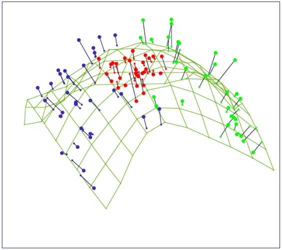
Sparse PCA extension: Vu et al. (2013), "Fantope projection and selection: near-optimal convex relaxation of sparse PCA"

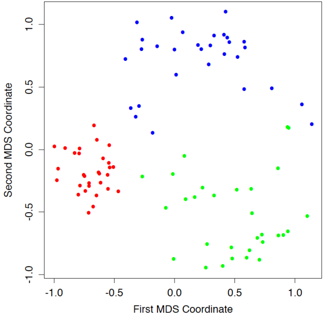
Classical multidimensional scaling

Given $x_1, \dots x_n \in \mathbb{R}^p$, and similarities $s_{ij} = (x_i - \bar{x})^T (x_j - \bar{x})$, classical multidimensional scaling solves the nonconvex problem

$$\min_{z_1,\dots z_n \in \mathbb{R}^k} \sum_{i,j} \Big(s_{ij} - (z_i - \bar{z})^T (z_j - \bar{z}) \Big)^2$$

for a fixed k.





From Hastie et al. (2009), "The elements of statistical learning"

Let S be the similarity matrix (entries S_{ij})

The classical MDS problem has an exact solution in terms of the eigendecomposition $S = UD^2U^T$:

$$\hat{z}_1, \ldots \hat{z}_n \text{ are the rows of } U_k D_k$$

where U_k is the first k columns of U, and D_k the first k diagonal entries of D

Note: other very similar forms of MDS are not convex, and not directly solveable, e.g., least squares scaling, with $d_{ij} = \|x_i - x_j\|_2$:

$$\min_{z_1,\ldots z_n \in \mathbb{R}^k} \sum_{i,j} \big(d_{ij} - \|z_i - z_j\|_2\big)^2$$

See Hastie et al. (2009), Chapter 14

Generalized eigenvalue problems

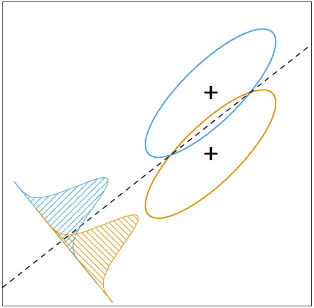
Given $B, W \in \mathbb{R}^{p \times p}$, $B, W \succeq 0$, consider the nonconvex problem

$$\max_{v \in \mathbb{R}^n} \frac{v^T B v}{v^T W v}$$

This is a generalized eigenvalue problem, with exact solution given by the top eigenvector of $W^{-1}B$

This is important, e.g., in Fisher's discriminant analysis, where B is the between-class covariance matrix, and W the within-class covariance matrix

See Hastie et al. (2009), Chapter 4

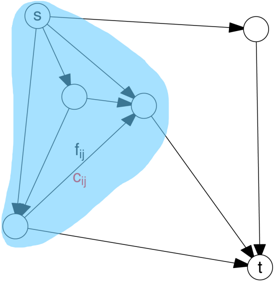


Graph problems

Min cut

Given a graph $G = (V, E)$ with $V = \{1, \dots n\}$, two nodes $s, t \in V$, and costs $c_{ij} \geq 0$ on edges $(i, j) \in E$. Min cut problem:

$$\begin{array}{ll} \min_{b \in \mathbb{R}^{|E|}, x \in \mathbb{R}^{|V|}} & \sum_{(i,j) \in E} b_{ij} c_{ij} \\ \text{subject to} & b_{ij} \geq x_i - x_j \\ & b_{ij}, x_i, x_j \in \{0, 1\} \\ & \text{for all } i, j, \\ & x_s = 0, \; x_t = 1 \end{array}$$



Think of b_{ij} as the indicator that the edge (i, j) traverses the cut from s to t; think of x_i as an indicator that node i is grouped with t. This nonconvex problem can be solved exactly using max flow (max flow/min cut theorem)

A relaxation of min cut

$$\begin{aligned}\min_{b\in\mathbb{R}^{|E|},x\in\mathbb{R}^{|V|}} \quad & \sum_{(i,j)\in E} b_{ij}c_{ij} \\ \text{subject to} \quad & b_{ij} \geq x_i - x_j, \;\; \text{all } i,j \\ & b \geq 0 \\ & x_s = 0, \;\; x_t = 1\end{aligned}$$

This is an LP; it is the dual of the max flow LP (see lecture 12):

$$\begin{aligned}\max_{f\in\mathbb{R}^{|E|}} \quad & \sum_{(s,j)\in E} f_{sj} \\ \text{subject to} \quad & f_{ij} \geq 0, \;\; f_{ij} \leq c_{ij} \quad \text{for all } (i,j)\in E \\ & \sum_{(i,k)\in E} f_{ik} = \sum_{(k,j)\in E} f_{kj} \quad \text{for all } k \in V \setminus \{s,t\}\end{aligned}$$

Max flow min cut theorem tells us that the relaxed min cut is tight

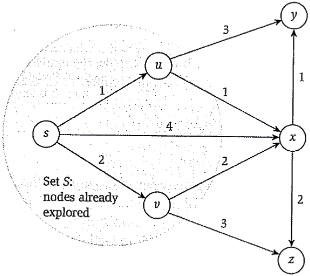
Shortest paths

Given a graph $G = (V, E)$, with edge costs c_e, $e \in E$, consider the shortest path problem, between two nodes $s, t \in V$

$$\min_{\text{Paths } P} \sum_{e \in P} c_e \quad \Longleftrightarrow \quad \begin{aligned} \min_{P=(e_1,\dots e_r)} \quad & \sum_{e \in P} c_e \\ \text{subject to} \quad & e_{1,1} = s,\ e_{r,2} = t \\ & e_{i,2} = e_{i+1,1},\ \ i = 1, \dots r-1 \end{aligned}$$

Dijkstra's algorithm solves this problem (and more), from Dijkstra (1959), "A note on two problems in connexion with graphs"

Clever implementations run in $O(|E| \log |V|)$ time; e.g., see Kleinberg and Tardos (2005), "Algorithm design", Chapter 5



Nonconvex proximal operators

Hard-thresholding

One of the simplest nonconvex problems, given $y \in \mathbb{R}^n$:

$$\min_{\beta \in \mathbb{R}^n} \sum_{i=1}^n (y_i - \beta_i)^2 + \sum_{i=1}^n \lambda_i 1\{\beta_i \neq 0\}$$

Solution is given by hard-thresholding y,

$$\beta_i = \begin{cases} y_i & \text{if } y_i^2 > \lambda_i \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, \ldots n$$

and can be seen by inspection. Special case $\lambda_i = \lambda$, $i = 1, \ldots n$,

$$\min_{\beta \in \mathbb{R}^n} \|y - \beta\|_2^2 + \lambda \|\beta\|_0$$

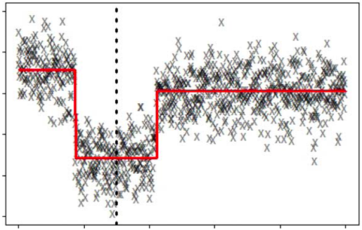
Compare to soft-thresholding, prox operator for ℓ_1 penalty. Note: changing the loss to $\|y - X\beta\|_2^2$ gives best subset selection, which is NP hard for general X

1-dimensional ℓ_0 segmentation

Consider the nonconvex 1-dimensional segmentation problem

$$\min_{\beta \in \mathbb{R}^n} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-1} 1\{\beta_i \neq \beta_{i+1}\}$$

Can be solved exactly using dynamic programming, in two ways: Bellman (1961), "On the approximation of curves by line segments using dynamic programming", and Johnson (2013) "A dynamic programming algorithm for the fused lasso and L_0-segmentation"



Johnson: more efficient, Bellman: more general

Worst-case $O(n^2)$, but with practical performance more like $O(n)$

Tree-leaves projection
(Credit: Miguel Carreira-Perpinan)

Given target $u \in \mathbb{R}^n$, tree g on $\mathbb{R}^n$, and label $y \in \{0,1\}$, consider

$$\min_{z \in \mathbb{R}^n} \|u - z\|_2^2 + \lambda \cdot 1\{g(z) \neq y\}$$

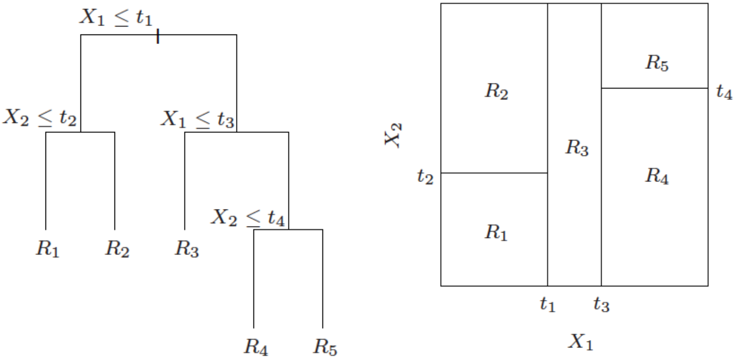
Interpretation: find z close to u, whose label under g is not unlike y. Argue directly that solution is either $\hat{z} = u$ or $\hat{z} = P_S(u)$, where

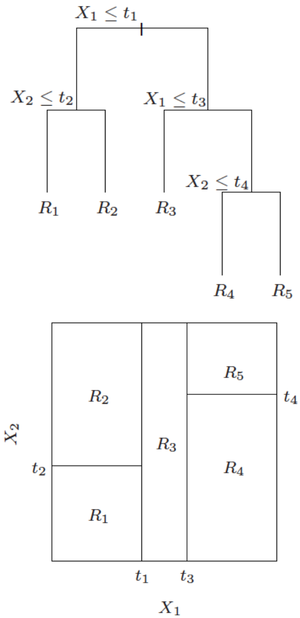
$$S = g^{-1}(1) = \{z \in \mathbb{R}^n : g(z) = y\}$$

the set of leaves of g assigned label y. We simply compute both options for $\hat{z}$ and compare costs. Therefore problem reduces to computing $P_S(y)$, the projection onto a set of tree leaves, a highly nonconvex set

This appears as a subroutine of a broader algorithm for nonconvex optimization; see Carreira-Perpinan and Wang (2012), "Distributed optimization of deeply nested systems"

The set S is a union of axis-aligned boxes; projection onto any one box is fast, $O(n)$ operations





To project onto S, could just scan through all boxes, and take the closest

Faster: Decorate each node of tree with labels of its leaves, and bounding box. Perform depth-first search, pruning nodes

- that do not contain a leaf labeled y, or
- whose bounding box is farther away than the current closest box

Discrete problems

Binary graph segmentation

Given $y \in \mathbb{R}^n$, and a graph $G = (V, E)$, $V = \{1, \dots n\}$, consider binary graph segmentation:

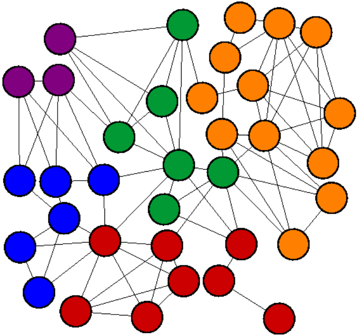
$$\min_{\beta \in \{0,1\}^n} \sum_{i=1}^n (y_i - \beta_i)^2 + \sum_{(i,j) \in E} \lambda_{ij} 1\{\beta_i \neq \beta_j\}$$

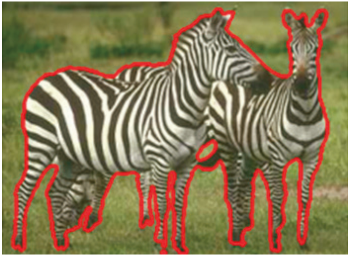
Simple manipulation brings this problem to the form

$$\max_{A \subseteq \{1,\dots n\}} \sum_{i \in A} a_i + \sum_{j \in A^c} b_j - \sum_{(i,j) \in E,\, |A \cap \{i,j\}| = 1} \lambda_{ij}$$

which is a segmentation problem that can be solved exactly using min cut/max flow. E.g., Kleinberg and Tardos (2005), "Algorithm design", Chapter 7

E.g., apply recursively to get a verison of graph hierarchical clustering (divisive)





E.g., take the graph as a 2d grid for image segmentation (From `http://ailab.snu.ac.kr`)

Discrete 1-dimensional ℓ_0 segmentation

Now consider discrete 1-dimensional segmentation:

$$\min_{\beta \in \{b_1, \dots b_k\}^n} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-1} 1\{\beta_i \neq \beta_{i+1}\}$$

where $\{b_1, \dots b_k\}$ is some fixed discrete set. This can be efficiently solved using classic (discrete) dynamic programming

Key insight is that the 1-dimensional structure allows us to exactly solve and store

$$\hat{\beta}_1(\beta_2) = \underset{\beta_1 \in \{b_1, \dots b_k\}}{\operatorname{argmin}} \underbrace{(y_1 - \beta_1)^2 + \lambda |\beta_1 - \beta_2|}_{f_1(\beta_1, \beta_2)}$$

$$\hat{\beta}_2(\beta_3) = \underset{\beta_2 \in \{b_1, \dots b_k\}}{\operatorname{argmin}} f_1\big(\hat{\beta}_1(\beta_2), \beta_2\big) + (y_2 - \beta_2)^2 + \lambda |\beta_2 - \beta_3|$$

$$\dots$$

Algorithm:

- Make a forward pass over $\beta_1, \ldots \beta_{n-1}$, keeping a look-up table; also keep a look-up table for the optimal partial criterion values $f_1, \ldots f_{n-1}$
- Solve exactly for β_n
- Make a backward pass $\beta_{n-1}, \ldots \beta_1$, reading off the look-up table

	b_1	b_2	$\ldots$	b_k
β_1				
β_2				
$\ldots$				
β_{n-1}				

	b_1	b_2	$\ldots$	b_k
f_1				
f_2				
$\ldots$				
f_{n-1}				

Requires $O(nk)$ operations

Infinite-dimensional problems

Smoothing splines

Given pairs $(x_i, y_i) \in \mathbb{R} \times \mathbb{R}$, $i = 1, \ldots n$, smoothing splines solve

$$\min_{f:\mathbb{R}\to\mathbb{R}} \sum_{i=1}^n \big(y_i - f(x_i)\big)^2 + \lambda \int_{-\infty}^{\infty} \big(f^{(\frac{k+1}{2})}(t)\big)^2 \, dt$$

for a fixed odd k. The domain of minimization here is all functions f for which $\int_{-\infty}^{\infty} (f^{(\frac{k+1}{2})}(t))^2 \, dt < \infty$. This problem is infinite-dimensional, but convex (in function space)

Can show that the solution $\hat{f}$ to the above problem is unique, and given by a natural spline of order k, with knots at $x_1, \ldots x_n$. This means we can restrict our attention to functions

$$f = \sum_{j=1}^n \theta_j \eta_j$$

where $\eta_1, \ldots \eta_n$ are natural spline basis functions

Plugging in $f = \sum_{j=1}^n \theta_j \eta_j$, transform smoothing spline problem into finite-dimensional form:

$$\min_{\theta \in \mathbb{R}^n} \|y - N\theta\|_2^2 + \lambda \theta^T \Omega \theta$$

where $N_{ij} = \eta_j(x_i)$, and $\Omega_{ij} = \int_{-\infty}^{\infty} \eta_i^{(\frac{k+1}{2})}(t)\, \eta_j^{(\frac{k+1}{2})}(t)\, dt$. The solution is explicitly given by

$$\hat{\theta} = (N^T N + \lambda \Omega)^{-1} N^T y$$

and fitted function is $\hat{f} = \sum_{j=1}^n \hat{\theta}_j \eta_j$. With proper choice of basis function (B-splines), calculation of $\hat{\theta}$ is $O(n)$

See, e.g., Wahba (1990), "Splines models for observational data"; Green and Silverman (1994), "Nonparametric regression and generalized linear models"; Hastie et al. (2009), Chapter 5

Locally adaptive regression splines

Given same setup, locally adaptive regression splines solve

$$\min_{f:\mathbb{R}\to\mathbb{R}} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \cdot \mathrm{TV}(f^{(k)})$$

for fixed k, even or odd. The domain is all f with $\mathrm{TV}(f^{(k)}) < \infty$, and again this is infinite-dimensional but convex

Again, can show that a solution $\hat{f}$ to above problem is given by a spline of order k, but two key differences:

- Can have any number of knots $\leq n-k-1$ (tuned by λ)
- Knots do not necessarily coincide with input points $x_1, \ldots x_n$

See Mammen and van de Geer (1997), "Locally adaptive regression splines"; in short, these are statistically more adaptive but computationally more challenging than smoothing splines

Mammen and van de Geer (1997) consider restricting attention to splines with knots contained in $\{x_1, \ldots x_n\}$; this turns the problem into finite-dimensional form,

$$\min_{\theta \in \mathbb{R}^n} \|y - G\theta\|_2^2 + \lambda \sum_{j=k+2}^{n} |\theta_j|$$

where $G_{ij} = g_j(x_i)$, and $g_1, \ldots g_n$ is a basis for splines with knots at $x_1, \ldots x_n$. The fitted function is $\hat{f} = \sum_{j=1}^n \hat{\theta}_j g_j$

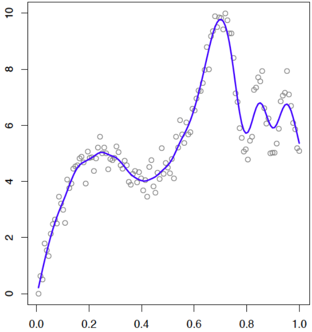
These authors prove that the solution of this (tractable) problem $\hat{f}$ and of the original problem f^* differ by

$$\max_{x \in [x_1, x_n]} |\hat{f}(x) - f^*(x)| \leq d_k \cdot \mathrm{TV}\big((f^*)^{(k)}\big) \cdot \Delta^k$$

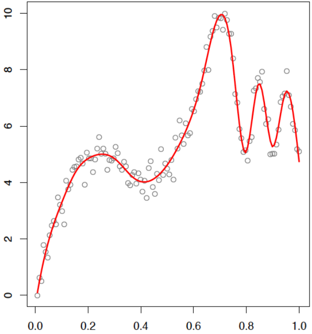
with Δ the maximum gap between inputs. Therefore, statistically it is reasonable to solve the finite-dimensional problem

E.g., a comparison, tuned to the same overall model complexity:

Smoothing spline



Finite-dimensional locally adaptive regression spline



The left fit is easier to compute, but the right is more adaptive

(Note: trend filtering estimates are asymptotically equivalent to locally adaptive regression splines, but with efficiency comparable to smoothing splines; see Tibshirani (2013), "Adaptive piecewise polynomial estimation via trend filtering")

Statistical problems

Sparse underdetermined linear systems

Suppose that $X \in \mathbb{R}^{n \times p}$ has unit normed columns, $\|X_i\|_2 = 1$, for $i = 1, \dots n$. Given y, consider the problem of finding the sparsest sparse linear solution

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_0 \;\; \text{subject to} \;\; X\beta = y$$

This is nonconvex and known to be NP hard, for a generic X. A natural convex relaxation is the ℓ_1 basis pursuit problem:

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_1 \;\; \text{subject to} \;\; X\beta = y$$

It turns out that there is a deep connection between the two; we cite results from Donoho (2006), "For most large underdetermined systems of linear equations, the minimal ℓ_1 norm solution is also the sparsest solution"

As n, p grow large, $p > n$, there exists a threshold ρ (depending on the ratio p/n), such that for most matrices X, if we solve the ℓ_1 problem and find a solution with:

- fewer than ρn nonzero components, then this is the unique solution of the ℓ_0 problem
- greater than ρn nonzero components, then there is no solution of the linear system with less than ρn nonzero components

(Here "most" is quantified precisely in terms of a probability over matrices X, constructed by drawing columns of X uniformly at random over the unit sphere in $\mathbb{R}^n$)

There is a large and fast-moving body of related literature. See Donoho et al. (2009), "Message-passing algorithms for compressed sensing" for a nice review

Nearly optimal K-means

Given data points $x_1, \dots x_n \in \mathbb{R}^p$, the K-means problem solves

$$\min_{c_1,\dots c_K \in \mathbb{R}^p} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \min_{k=1,\dots K} \|x_i - c_k\|_2^2}_{f(c_1,\dots c_K)}$$

This is NP hard, and is usually approximately solved using Lloyd's algorithm, run many times, with random starts

Careful choice of starting positions makes a big impact: running Lloyd's algorithm once, from $c_1 = s_1, \dots c_K = s_K$, for cleverly chosen random $s_1, \dots s_K$, yields estimates $\hat{c}_1, \dots \hat{c}_K$ satisfying

$$\mathbb{E}\big[f(\hat{c}_1, \dots \hat{c}_K)\big] \le 8(\log k + 2) \cdot \min_{c_1,\dots c_K \in \mathbb{R}^p} f(c_1, \dots c_K)$$

See Arthur and Vassilvitskii (2007), "k-means++: The advantages of careful seeding"

In fact, the construction of $s_1, \ldots s_K$ is very simple:

- Begin by choosing s_1 uniformly at random among $x_1, \ldots x_n$
- Compute squared distances

$$d_i^2 = \|x_i - s_1\|_2^2$$

  for all points i not chosen, and choose s_2 by drawing from the remaining points, with probability weights $d_i^2 / \sum_j d_j^2$

- Recompute the squared distances as

$$d_i^2 = \min\left\{\|x_i - s_1\|_2^2, \|x_i - s_2\|_2^2\right\}$$

  and choose s_3 according to the same recipe

- And so on, until $s_1, \ldots s_K$ are chosen