Sparse Optimization Methods

Stephen Wright

University of Wisconsin-Madison

Toulouse, Feb 2009

1. Sparse Optimization
   - Motivation for Sparse Optimization
   - Applications of Sparse Optimization
   - Formulating Sparse Optimization Problems
2. Compressed Sensing
3. Matrix Completion
4. Composite Minimization Framework
5. Conclusions

+ Adrian Lewis, Ben Recht, Sangkyun Lee.

Sparse Optimization: Motivation

Look for simple approximate solution of optimization problem, rather than a (more complex) exact solution.

- Occam's Razor: Simple explanations of the observations are preferable to complex explanations.
- Noisy data doesn't justify solving the problem exactly.
- Simple / structured solutions are sometimes more robust to data inexactness.
- Often easier to actuate / implement / store / explain simple solutions.
- May conform better to prior knowledge.

When the solution is represented in an appropriate basis, simplicity or structure may show up as sparsity in x (i.e. few nonzero components).

Sparse optimization does not (necessarily) involve sparse linear algebra!

Example: Compressed Sensing

Given $k \times n$ matrix A and observation vector y, find *sparse* x with

$$Ax \approx y.$$

We can reconstruct x from A and y, even when $k \ll n$ and when noise is present in y, provided:

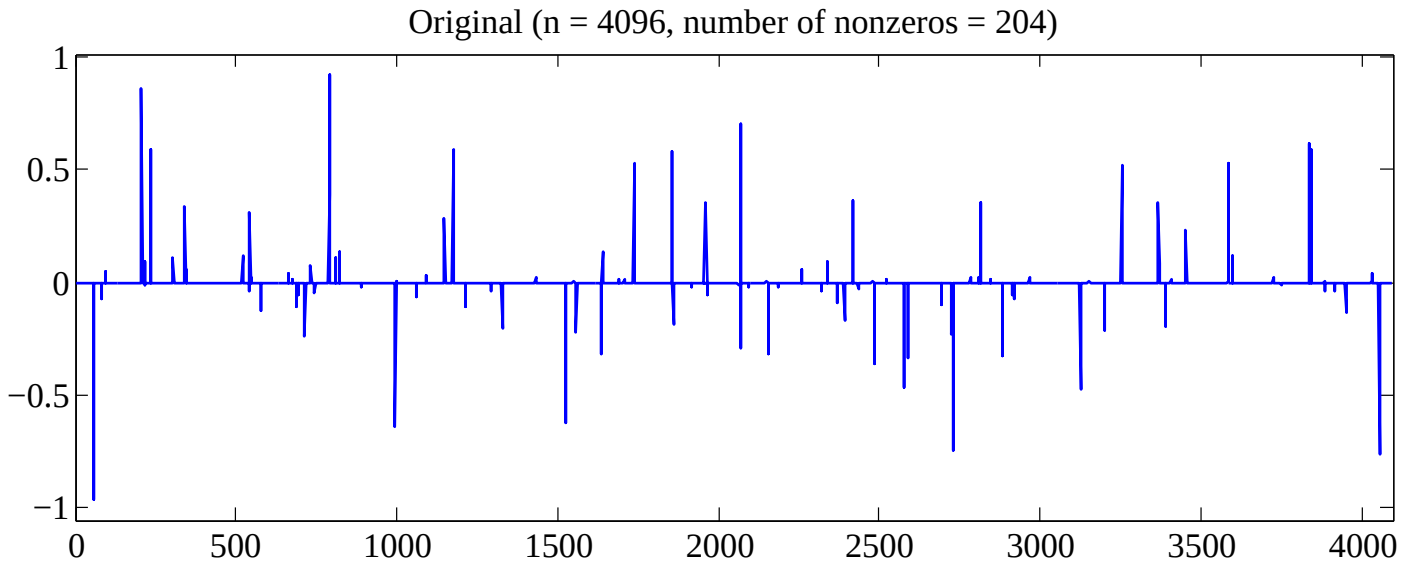
- We know that x is sparse (or nearly so);
- There are enough observations k, relative to sparsity of x;
- A satisfies restricted isometry properties (RIP) that ensure that for all sparse vectors x^1 and x^2, we have $\|A(x^1 - x^2)\|_2 \approx \|x^1 - x^2\|_2$.

If A is a projection from $\mathbb{R}^n$ onto a random k-dimensional subspace, it will have such properties. (Johnson-Lindenstrauss)

Reconstruction: Given A and y, and possibly some knowledge of sparsity level and noise type, recover x.

There are 204 spikes out of 4096 entries.

- Conventional signal processing indicates that you would need at least 4096 measurements (e.g. an FFT, a component-by-component sample) to determine x.
- Using compressed sensing, it can be reconstructed exactly from 1000 random linear combinations of the components of x.



Example: Image Processing

Image Denoising: Given a rectangular array of pixel intensities $f = [f_{ij}]$, $i, j = 1, 2, \ldots, N$, find a "denoised" array $u = [u_{ij}]$ that is close to f but has smaller total variation (more cartoon-like).

Formulate as a data-fitting problem with a regularization term that penalizes the discrete spatial gradient of u:

$$\min_u P(u) := \frac{\lambda}{2}\|u - f\|_2^2 + \sum_{i,j} \left\| \begin{bmatrix} u_{i+1,j} - u_{i,j} \\ u_{i,j+1} - u_{i,j} \end{bmatrix} \right\|_2$$

Tends to filter out random noise in pixels of f. As λ increases, u is closer to the measured image f.

Figure: CAMERAMAN: original (left) and noisy (right)

Figure: Denoised CAMERAMAN: Tol=10^{-2} (left) and Tol=10^{-4} (right).

Example: Matrix Completion

Seek low-rank matrix $X \in \mathbb{R}^{n_1 \times n_2}$ such that $X_{ij} \approx M_{ij}$ for $(i,j) \in \Omega$, where

- Ω is a set of index pairs in $\{1, 2, \ldots, n_1\} \times \{1, 2, \ldots, n_2\}$;
- M_{ij} are given observations.

Example: Netflix Prize, Covariance Estimation.

More general variant: Seek low-rank X such that $\mathcal{A}(X) \approx b$, where $\mathcal{A}$ is a linear mapping on elements of X and b is the vector of observations.

In some sense, extends compressed sensing to matrix variables.

- "Simplicity" $\sim$ "low rank" rather than sparsity.
- Many algorithmic ideas extend, and new ones arise.
- Linear algebra issues are more complicated and more central.

Example: Tensor Decompositions

Given an N-dimensional tensor X, the CP decomposition expresses X approximately as an outer product of F rank-1 tensors:

$$X_{i_1,i_2,\ldots,i_N} \approx \sum_{f=1}^{F} a^{(1)}_{i_1,f} a^{(2)}_{i_2,f} \ldots a^{(N)}_{i_N,f}.$$

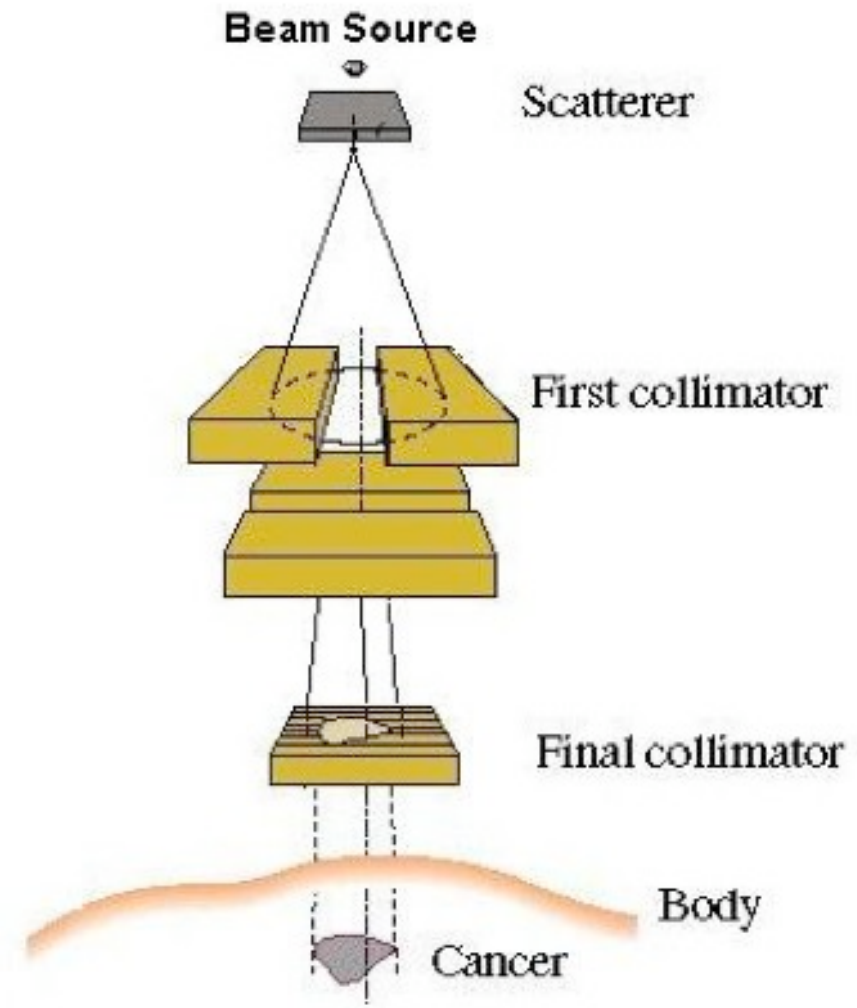
Rank of a tensor is the smallest F for which exact equality holds. However things are much more complicated than in the matrix case ($N = 2$):

- F may be different over $\mathbb{R}$ and $\mathbb{C}$.
- Finding F is NP-hard.
- *Maximum* and *typical* ranks of random tensors may be different.
- Minimum-rank decompositions are nonunique for matrices, but often unique for tensors.
- Can have a sequence of rank-F tensors approaching a rank-$(F+1)$ tensor.

There is interest in solving "tensor completion" problems where we find a rank-F tensor that closely approximates the observations in a given tensor.

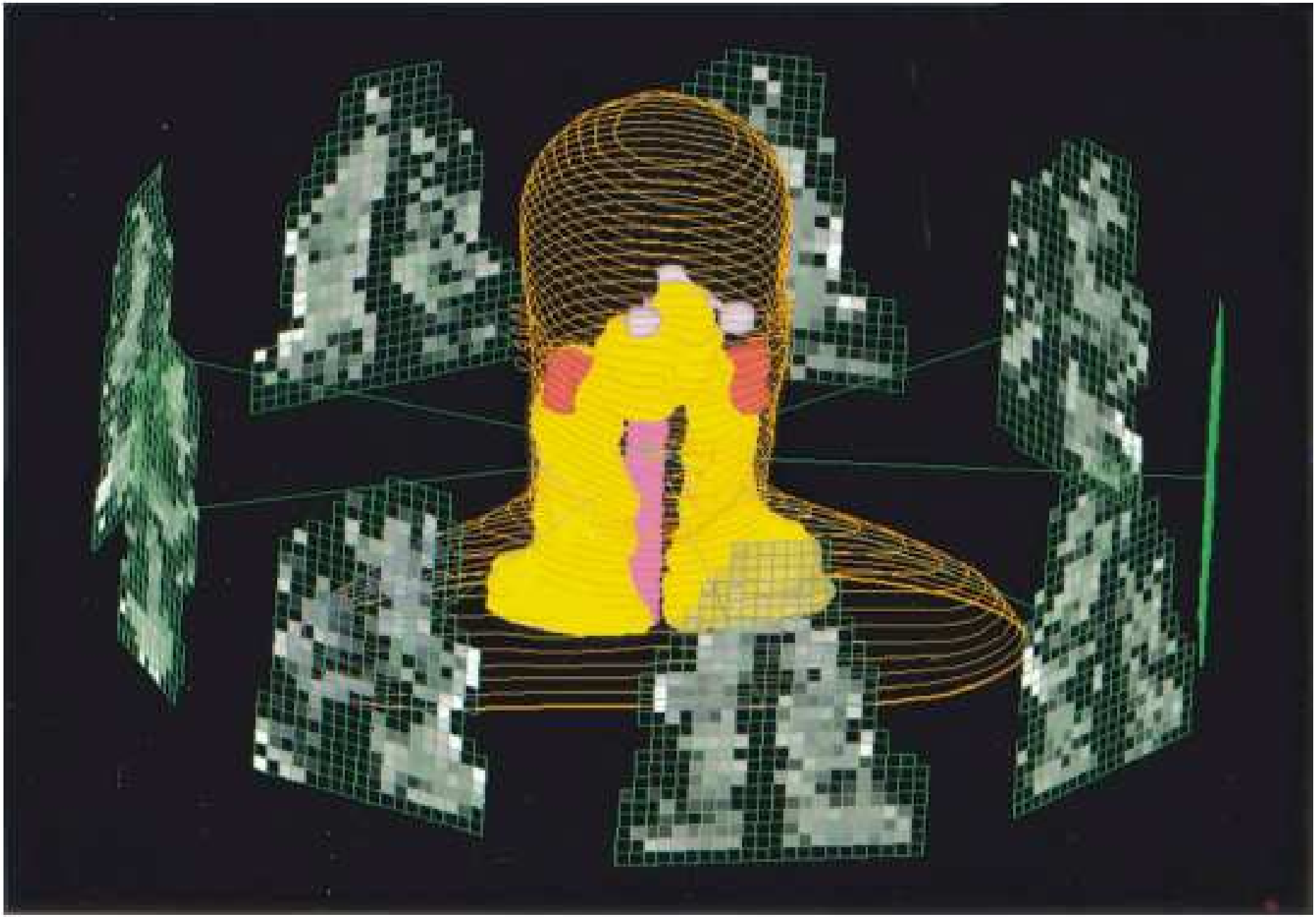
Example: Radiotherapy for Cancer

- Deliver radiation from an external device to an internal tumor.
- Shape radiation beam, choose angles of delivery so as to deliver prescribed radiation dose to tumor while avoiding dose to surrounding tissue and organs.
- Use just a *few* different beam shapes and angles, to simplify the treatment, avoid spending too much time on the device, hopefully reduce the likelihood of treatment errors.



Linear accelerator, showing cone and collimators

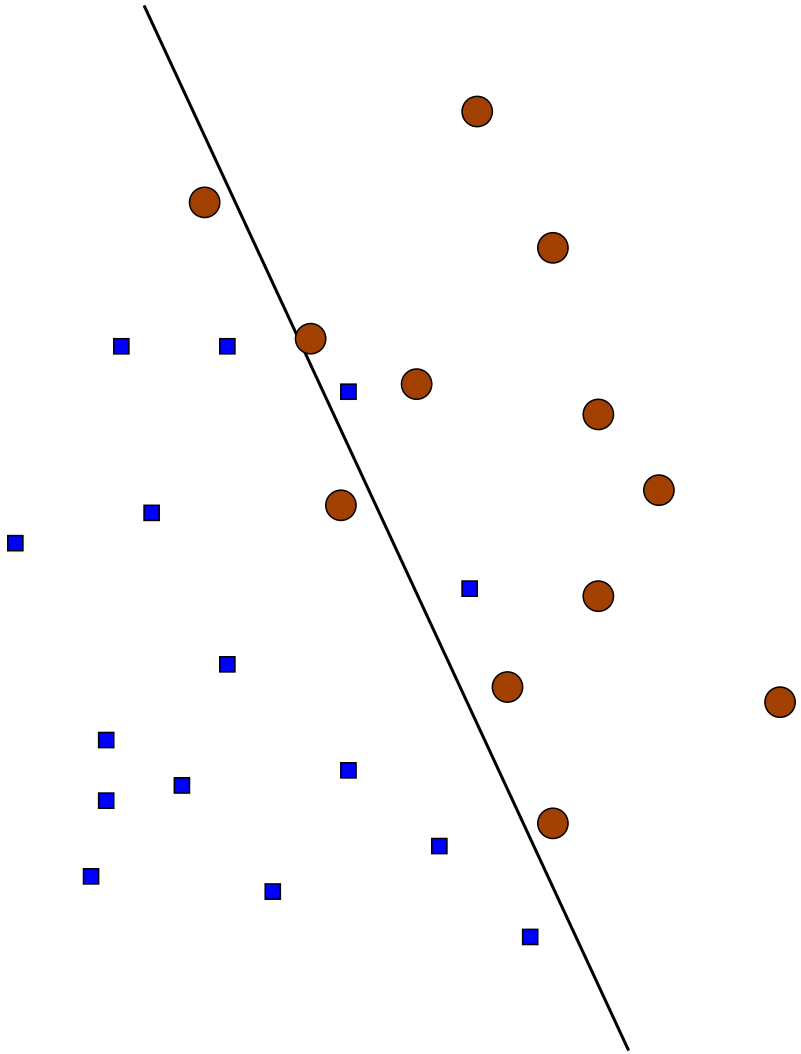
Multileaf collimator. Leaves move up and down to shape the beam.



Class of Examples: Extracting Information from Data

- We are drowning in data!
- Key challenge: Extract salient information from large data sets efficiently.
- What's "Salient"
  - Main effects — the essence — not minor effects that possibly overfit the observations
  - The main effects are sometimes complex combinations of the basic ones — that is, we are looking for a small number from a potentially huge set — needle in a haystack.
  - The problem is sparse, by our definition.
- A few specific instances follow...

Example: Support Vector Machines (Linear)



- Have attribute vectors $x_1, x_2, \ldots, x_m$ (real vectors) and labels $y_1, y_2, \ldots, y_m$ (binary ± 1).
- Seek a hyperplane $w^T x - b$ defined by (w, b) that separates the points according to their classification:

$$w^T x_i - b >= 1 \Rightarrow y_i = 1, \qquad w^T x_i - b <= -1 \Rightarrow y_i = -1$$

(for most i).

- Obtain (w, b) from a function that penalizes incorrect classifications with a loss function, and also keeps $\|w\|_2$ small:

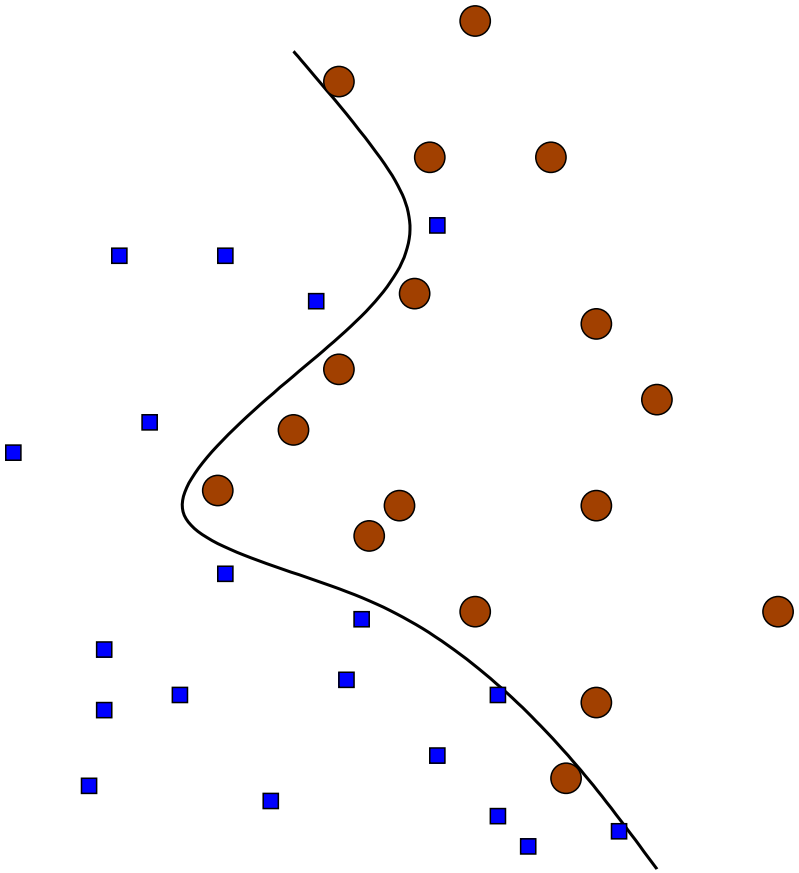
$$\min_{(w,b)} \frac{\lambda}{2} w^T w + \sum_{i=1}^{m} \max\left(1 - y_i[w^T x_i - b], 0\right).$$

- Dual formulation:

$$\max_{\alpha} e^T \alpha - \frac{1}{2} \alpha^T Y^T K Y \alpha \ \text{ subject to } \ \alpha^T y = 0, \ \ 0 \leq \alpha \leq C\mathbf{1},$$

where $y = (y_1, y_2, \ldots, y_m)^T$, $Y = \mathrm{diag}(y)$, $K_{ij} = x_i^T x_j$ is the kernel.

Example: Support Vector Machines (Nonlinear)



When a hyperplane is inadequate for separating the vectors, can find a nonlinear classifier by mapping the x_i into a higher-dimensional space (via a function $\phi : \mathbb{R}^n \to \mathcal{H}$) and doing linear classification there. When the hyperplane is projected into the original space, it gives a nonlinear surface.

Don't need to define ϕ explicitly! Instead define the kernel function $k(s,t)$ to be a measure of closeness of s and t. Implicitly, $k(s,t) = \langle \phi(s), \phi(t) \rangle$.

Can define the *dual* SVM optimization problem and the classifier function in terms of k alone — no need for ϕ.

$$\max_{\alpha} \; e^T\alpha - \frac{1}{2}\alpha^T Y^T K Y \alpha \;\; \text{subject to} \;\; \alpha^T y = 0, \;\; 0 \leq \alpha \leq (1/\lambda)\mathbf{1},$$

where $K_{ij} = k(x_i, x_j)$ is the kernel. (Can get a primal formulation too.)

Where does sparsity come in? Can formulate approximate versions of these problems in which few of the α are allowed to be nonzero. (In fact, these are more tractable when m is very large.)

Example: Regularized Logistic Regression

Have attribute vectors $x_1, x_2, \ldots, x_m$ (real vectors) and labels $y_1, y_2, \ldots, y_m$ (binary 0/1).

Instead of a classifier, want to construct a function p that will give the probability of a given vector X having label $Y = 1$.

Model *log odds* or *logit* function as linear combination of basis functions $B_l(x)$, $l = 1, 2, \ldots, N$ (N may be huge):

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \sum_{l=0}^{N} a_l B_l(x),$$

Define a log-likelihood function (of the coefficients $a_1, a_2, \ldots, a_N$):

$$\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log p(x_i) + (1-y_i)\log(1-p(x_i))\right].$$

Choose coefficients $(a_1, a_2, \ldots, a_N)$ sparsely to approximately maximize this function.

Example: Regularized Regression

(Lasso: Tibshirani, 1997) Want to find a sparse least-squares solution to an overdetermined problem $Ax \approx b$. Solve:

$$\min_x \|Ax - b\|_2^2 \text{ s.t. } \|x\|_1 \leq T$$

for some parameter $T > 0$. In fact, can trace the solution x as a function of T. Generally higher T leads to less sparse x.

Can extend to *group* lasso, where x is broken into disjoint subvectors $x_{[l]}$, $l = 1, 2, \ldots, K$, and we impose the constraint:

$$\sum_{l=1}^{K} \|x_{[l]}\|_\infty \leq T \qquad \text{or} \qquad \sum_{l=1}^{K} \|x_{[l]}\|_2 \leq T.$$

That is, each subvector $x_{[l]}$ is "turned on or off" as a group, not by individual components, e.g. [Turlach, Venables, Wright, 2005].

Can also have non-disjoint subvectors, i.e. when the components are arranged in a tree (e.g. wavelet coefficients), sometimes wish to turn subtrees on and off, not individual nodes.

Formulating Sparse Optimization Problems

Two basic ingredients:

- An underlying optimization problem — often of data-fitting or max-likelihood type
- Regularization term or constraints or imposed structure to encourage sparsity / structure — usually nonsmooth.

Usually large, computationally demanding. Need techniques from

- Large-scale optimization
- Nonsmooth optimization
- Conic programming
- Computational linear algebra
- Statistics
- Heuristics

Also a lot of domain-specific knowledge.

Nonsmooth Norms are Useful!

Consider first a scalar function $f : \mathbb{R} \to \mathbb{R}$. Want to find x that approximately minimizes f, but accept 0 as an approximate solution provided it's not too far off.

One approach is to add the nonsmooth regularizer $|x|$ with parameter $\lambda > 0$, and solve

$$\min_x f(x) + \lambda |x|$$

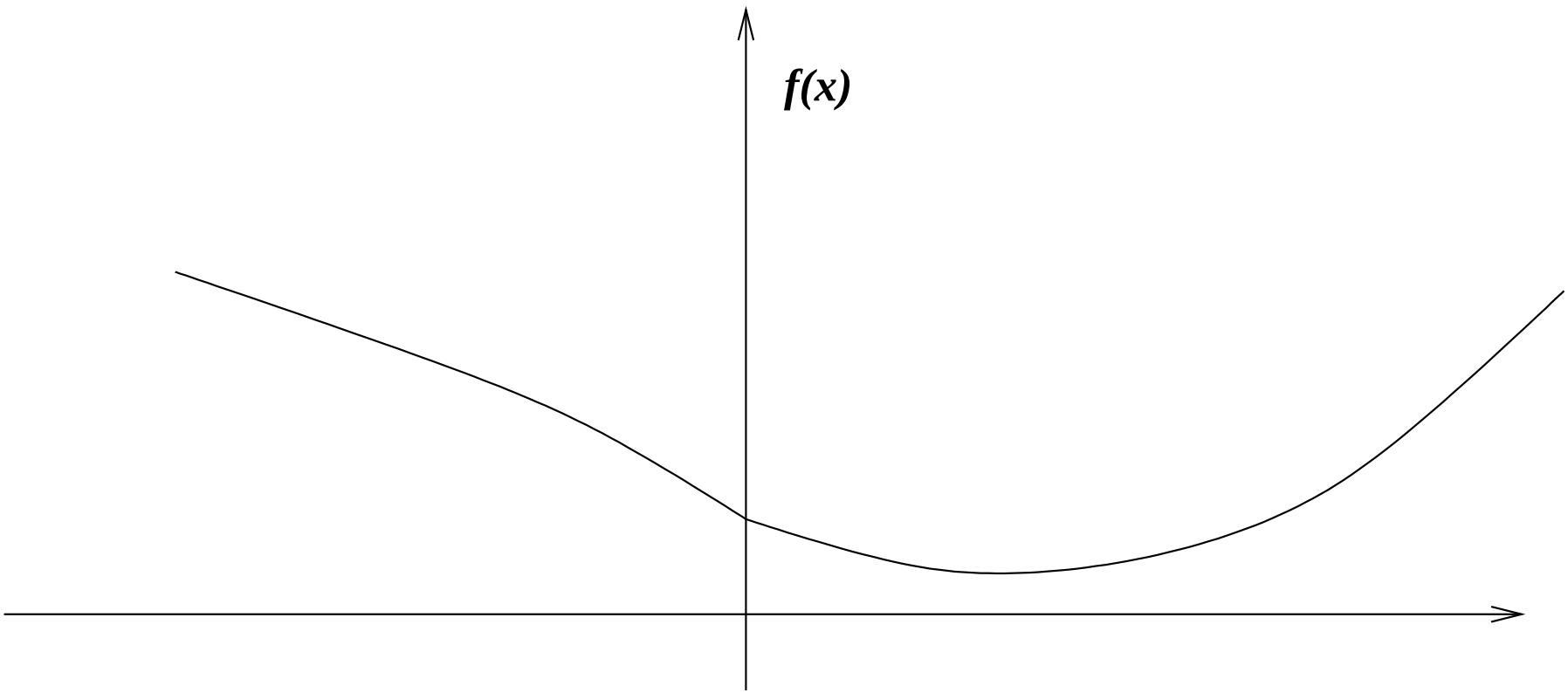
First-order optimality conditions are $0 \in \partial f(x)$, where

$$\partial f(x) = \begin{cases} f'(x) - \lambda & \text{if } x < 0 \\ f'(0) + \lambda[-1, 1] & \text{if } x = 0 \\ f'(x) + \lambda & \text{if } x > 0. \end{cases}$$

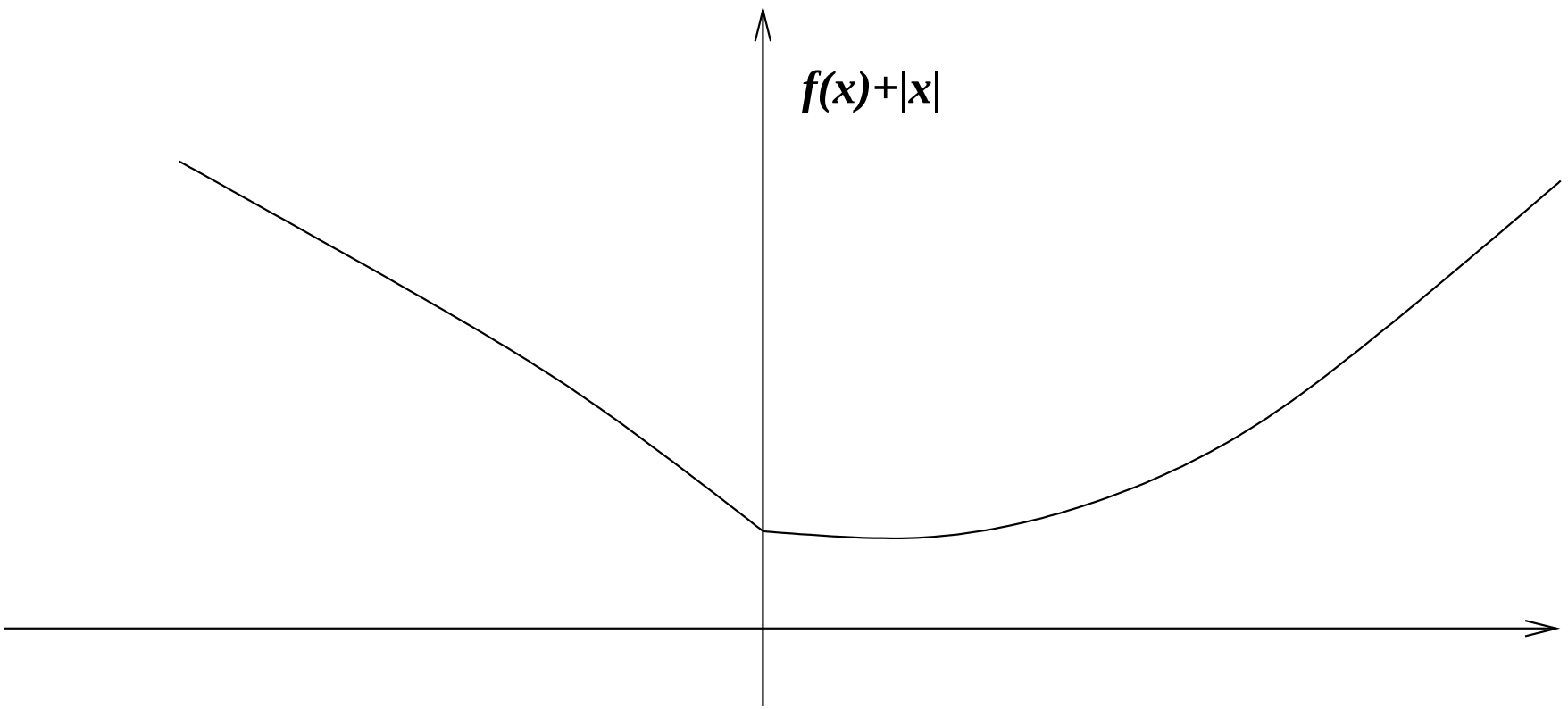
Introduces nonsmoothness at the kink $x = 0$, making it more "likely" that 0 will be chosen as the solution.

The "likelihood" increases as λ increases.

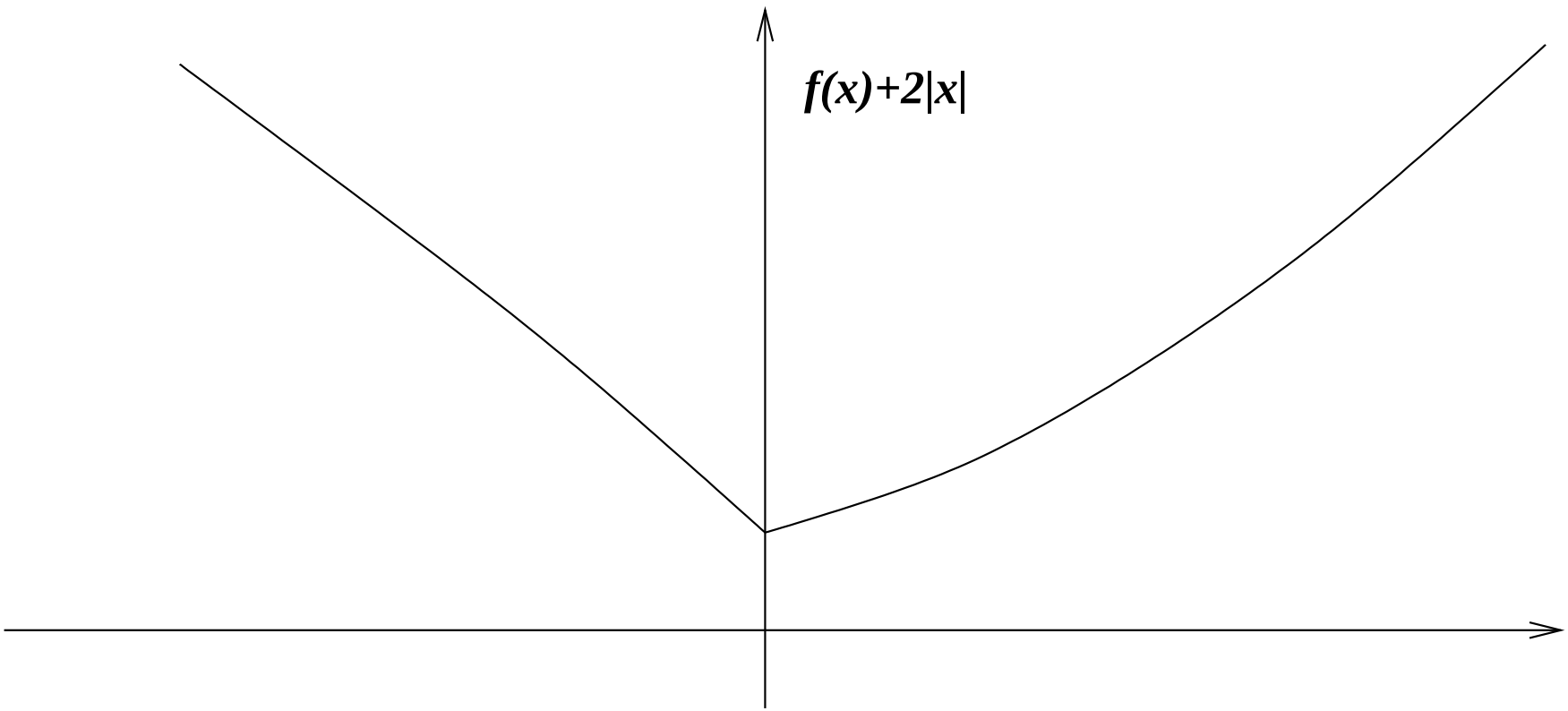
Effect of λ



Effect of λ



Effect of λ



Higher Dimensions

In higher dimensions, if we design nonsmooth functions $c(x)$ that have their "kinks" at points that are "sparse" according to our definition, then they are suitable regularizers for our problem.

Examples:

- $c(x) = \|x\|_1$ will tend to produce x with few nonzeros.
- $c(x) = \|x\|_1$ is less interesting — kink only when all components are zero (all or nothing).
- $c(x) = \|x\|_\infty$ has kinks where components of x are equal — also may not be interesting for sparsity.
- $c(x) = \sum_{l=1}^{K} \|x_{[l]}\|_2$ has kinks where $x_{[l]} = 0$ for some l – suitable for group sparsity.
- Total Variation norm: Has kinks where $u_{i,j} = u_{i+1,j} = u_{i,j+1}$ for some i, j, i.e. where spatial gradient is zero.

Compressed Sensing

$$Ax \approx y, \qquad A \in \mathbb{R}^{n \times k}, \quad n \ll k.$$

Given sparsity level $S \leq k$, A satisfies RIP with isometry constant $\delta_S < 1$ if for any column submatrix $A_{\cdot T}$ of A with at most S columns, we have

$$(1 - \delta_S)\|c\|_2^2 \leq \|A_{\cdot T} c\|_2^2 \leq (1 + \delta_S)\|c\|_2^2, \qquad \text{for all } c \in \mathbb{R}^S.$$

That is, each column submatrix with k columns is nearly orthonormal.

If δ_{2S} is somewhat less than 1, then A can distinguish clearly between any two vectors in $\mathbb{R}^n$ with sparsity level S or below.

Random matrices with good RIP include:

- elements of A drawn i.i.d. from $N(0, 1)$;
- columns of A uniformly distributed on the unit sphere in $\mathbb{R}^k$;
- row submatrix of discrete cosine transform.

RIP and Tractability

A natural formulation for the recovery problem might be:

$$\min \|x\|_0 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \epsilon,$$

where ϵ is related to expected noise in measurements. This is intractable for general A.

However when RIP is good, greedy methods can succeed in recovering sparse signals.

In addition, fundamental theory in compressed sensing [Candes, Romberg, Tao 05], [Donoho 04] shows that when RIP or similar properties hold, $\|\cdot\|_1$ can be used as a surrogate for $\|\cdot\|_0$.

This observation leads to convex optimization formulations.

Optimization Formulations of the Recovery Problem

LASSO with parameter $\beta > 0$:

$$\min \frac{1}{2}\|Ax - y\|_2^2 \qquad \text{subject to } \|x\|_1 \leq \beta.$$

Reconstruction with noise bound ϵ:

$$\min \|x\|_1 \qquad \text{subject to } \|Ax - y\|_2 \leq \epsilon.$$

Unconstrained nonsmooth formulation with regularization $\tau > 0$.

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1.$$

By varying their parameters, all three formulations generally lead to the same path of solutions.

Algorithms

- Interior-point:
  - Primal-dual: SparseLab / PDCO [Saunders et al 98, 02] `l1_ls` [Kim et al 07]
  - SOCP: ℓ_1-magic [Candès, Romberg 05]
- Gradient projection on QP formulation: GPSR [Figueiredo, Nowak, Wright 07].
- Pivoting / Homotopy a la LARS: SparseLab / SolveLasso
- Iterative shrinking-thresholding / Forward-backward splitting / Fixed-point: [Daubechies, Defriese, DeMol 04], [Combettes, Wajs 05], FPC [Hale, Yin, Zhang 07], SpaRSA [Wright, Figueiredo, Nowak 08].
- Augmented Lagrangian / Bregman [Yin et al 08] SALSA [Afonso et al 09]
- Matching pursuit: OMP [Pati, Rezaiifar, Krishnaprasad 93] [Davis, Mallat, Avellaneda 97], CoSaMP [Needell, Tropp 08].
- Optimal first-order: [Nesterov 07], FISTA [Beck, Teboulle 08].

Orthogonal Matching Pursuit (OMP)

$$q(x) := \frac{1}{2}\|Ax - y\|_2^2, \qquad \nabla q(x) = A^T r, \ \text{ where } r := Ax - b.$$

OMP chooses elements of ∇q one at a time, allowing the corresponding components of x to move away from 0 and adjust r accordingly.

Given A, y, set $t = 1$, $r_0 = 0$, and $\Omega_0 = \emptyset$.

1. Define n_t to be largest compoment of $A^T r_{t-1}$ and set $\Omega_t = \Omega_{t-1} \cup \{n_t\}$;
2. Solve reduced least squares problem $u_t := \min_u \|y - A_{\cdot\Omega_t} u\|_2^2$ and define $r_t = y - A_{\cdot\Omega_t} u_t$;
3. Repeat until termination test satisfied.

Main costs per iteration are multiplications by A and A^T.

OMP and Descendants

OMP is fundamental, extremely simple, and cheap, but theoretical guarantees are not too strong, and practical performance varies.

Can form the basis of more sophisticated algorithms (e.g. CoSaMP) that have more complex strategies for updating Ω_t and make bigger changes to the reduced least-squares method at each iteration.

In all these methods, if RIP holds, the matrix $A_{\cdot\Omega_t}$ is well conditioned provided $|\Omega_t|$ is not much bigger than the true sparsity of x.

SpaRSA

$$\min \phi(x) := \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1.$$

Define $q(x) := (1/2)\|Ax - y\|_2^2$. From iterate x^k, get step d by solving

$$\min_d \nabla q(x^k)^T d + \frac{1}{2}\alpha_k d^T d + \tau\|x^k + d\|_1.$$

Can view the α_k term as an approximation to the Hessian: $\alpha_k I \approx \nabla^2 q = A^T A$. (When RIP holds, this approximation is good, for small principal submatrices of AA^T.)

Subproblem is trivial to solve in $O(n)$ operations, since it is separable in the components of d. Equivalent to the shrinkage operator:

$$\min_z \frac{1}{2}\|z - u^k\|_2^2 + \frac{\tau}{\alpha_k}\|z\|_1, \qquad \text{with } u^k := x^k - \frac{1}{\alpha_k}\nabla q(x^k).$$

Choosing α_k

- By choosing α_k greater than a threshold value $\bar{\alpha}$ at every iteration, can guarantee convergence, but slowly.
- Can use a Barzilai-Borwein (BB) strategy: choose α_k it to mimic the true Hessian $A^T A$ over the step just taken. e.g. do a least squares fit to:

$$[x^k - x^{k-1}] \approx \alpha_k^{-1}[\nabla q(x^k) - \nabla q(x^{k-1})].$$

  Generally non-monotone.
- Cyclic BB variants: e.g. update α_k only every 3rd iteration.
- Get monotone variants by backtracking: set $\alpha_k \leftarrow 2\alpha_k$ repeatedly until a decrease in objective is obtained.

SpaRSA Implementation and Properties

- Exploits warm starts well.
- Problem is harder to solve for smaller τ (corresponding to more nonzeros in x). Performance improved greatly by continuation:
  - Choose initial $\tau_0 \leq \|A^T y\|_\infty$ and decreasing sequence $\tau_0 > \tau_1 > \tau_2 > \ldots > \tau_{\text{final}} > 0$, where τ_{final} is the target final value.
  - Solve for τ equal to each element in sequence, using previous solution as the warm start.
- Debiasing: After convergence of the main algorithm, fix nonzero set (support) in x and minimize $\|Ax - b\|_2^2$ over this reduced set.
- Can make large changes to the active manifold on a single step (like interior-point, unlike pivoting).
- Each iteration is cheap: one multiplication each with A or A^T

Matrix Completion

Seek low-rank matrix $X \in \mathbb{R}^{n_1 \times n_2}$ such that $\mathcal{A}(X) \approx b$, where $\mathcal{A}$ is a linear mapping on elements of X and b is the vector of observations.

In some sense, extends matrix completion is compressed sensing on matrix variables. Linear algebra is more complicated.

Can formulate as

$$\min_X \text{rank}(X) \;\; \text{s.t.} \;\; \mathcal{A}(X) = b$$

for exact observations, or

$$\min_X \text{rank}(X) \;\; \text{s.t.} \;\; \|\mathcal{A}(X) - b\| \leq \epsilon$$

for noisy observations.

Matrix Completion Formulations

To get a convex optimization formulation, replace rank(X) by its *convex envelope* on the set $\{X \mid \|X\|_2 \leq 1\}$, which is the nuclear norm $\|X\|_*$ defined by

$$\|X\|_* = \sum_{i=1}^{n_2} \sigma_i(X),$$

where $\sigma_i(X)$ is the ith singular value of X. This is a nonsmooth convex function of X.

Obtain formulations

$$\min_X \|X\|_* \text{ s.t. } \mathcal{A}(X) = b \tag{1}$$

and

$$\min_X \tau\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2.$$

Algorithms like SpaRSA

Obtain an extension of the SpaRSA approach by using $\alpha_k I$ to approximate $\mathcal{A}^*\mathcal{A}$. Obtain steps from the shrinkage operator by solving:

$$\min_Z \frac{\tau}{\alpha_k}\|Z\|_* + \frac{1}{2}\|Z - Y^k\|_F^2,$$

where

$$Y^k := X^k - \frac{1}{\alpha_k}\mathcal{A}^*(\mathcal{A}(X^k) - b).$$

e.g. [Ma, Goldfarb, Chen, 08]. Can prove convergence for α_k sufficiently large (uniformly greater than $\lambda_{\max}(\mathcal{A}^*\mathcal{A})/2$).

Can enhance by similar strategies as in compressed sensing:

- Continuation
- Barzilai-Borwein α_k, nonmonotonicity
- Debiasing.

Implementing Shrinking Methods

Main operation is the shrinkage operator:

$$\min_Z \nu\|Z\|_* + \frac{1}{2}\|Z - Y\|_F^2,$$

which can be solved via an SVD of Y. Calculate $Y = U\Sigma V^T$, then define diagonal matrix $\Sigma(\nu)$ by

$$\Sigma(\nu)_{ii} = \max(\Sigma_{ii} - \nu, 0),$$

and set $Z = U\Sigma(\nu)V^T$. Expensive for problems of interesting size.

Need for approximate SVD strategies.

- possibly based on sampling;
- possibly using Lanczos iterations;
- possibly exploiting the fact that we often need only a few leading singular values and vectors

See [Halko, Martinsson, Tropp 09] for a review of sampling-based approximate factorizations.

Explicit Parametrization of X

From [Recht, Fazel, Parrilo 07] and earlier work of Burer, Monteiro, Choi in SDP setting.

Choose target rank r and approximate X by LR^T, where $L \in \mathbb{R}^{n_1 \times r}$ and $R \in \mathbb{R}^{n_1 \times r}$ are the unknowns in the problem. For the formulation:

$$\min_X \|X\|_* \text{ s.t. } \mathcal{A}(X) - b,$$

we have the following *equivalent* formulation:

$$\min_{L,R} \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right) \text{ s.t. } \mathcal{A}(LR^T) = b.$$

A nonconvex minimization problem. Local solutions can be found by e.g. the method of multipliers [RFP 07], [Recht 08] using nonlinear conjugate gradient (modified Polak-Ribière) for the subproblems.

Can perform exact line search with a quartic rootfinder [Burer, Choi 06].

Explicit Parametrization: Noisy Formulation

$$\min_{L,R} \tau \left(\|L\|_F^2 + \|R\|_F^2 \right) + \frac{1}{2} \left\| \mathcal{A}(LR^T) - b \right\|_2^2.$$

Again, can use nonlinear conjugate gradient with exact line search, and can do continuation on τ.

- No need for SVD. Implementations are easy.
- Local minima? [Burer 06] shows that (in an SDP setting) provided r is chosen large enough, the method should not converge to a local solution — only the global solution.
- Performance degrades when rank is overestimated, probably because of degeneracy.

Investigations of this approach are ongoing.

Summarizing: Tools Used for Matrix Completion

- Formulation Tools:
  - Nuclear norm as a proxy for rank.
  - Lagrangian theory (equivalence of different formulations).
  - No local solutions, despite nonconvexity.
- Optimization Tools:
  - Operator splitting (the basis of IST)
  - Gradient projection
  - (Optimal) gradient and subgradient methods
  - Augmented Lagrangian
  - Algorithms for large-scale nonlinear unconstrained problems (nonlinear CG, L-BFGS)
  - Semidefinite programming
  - Handling of degeneracy
- Linear Algebra Tools:
  - SVD
  - Approximate SVD via sampling
  - Lanczos

Composite Minimization Framework

[Lewis, Wright 08] Develop a general algorithmic framework and supporting theory, for extension of SpaRSA-like approaches to a much wider class of problems.

$$\min_x \; h(c(x))$$

- vector function $c : \mathbb{R}^n \to \mathbb{R}^m$ is smooth;
- scalar function $h : \mathbb{R}^m \to \mathbb{R}$ usually nonsmooth.

In most of the analysis, we allow h to be

- extended-valued (to enforce some constraints explicitly)
- subdifferentially regularity or prox-regular.

Many applications have h convex — the analysis is much simpler in this case.

Examples

Compressed Sensing

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

where $A \in \mathbb{R}^{m\times n}$ with $m \ll n$. The second term induces sparsity in the optimal x, generally more sparse as λ increases. Composite formulation has $m = n + 1$ and

$$c(x) = \begin{bmatrix} f(x) \\ x \end{bmatrix}, \qquad h(c) = c_1 + \tau\|c_{2:n+1}\|_1.$$

Regularized Logistic Regression and Group-Regularized Regression problems can be framed similarly.

ℓ_1 Penalty Function:

$$\min f(x) \;\; s.t. \;\; r(x) \leq 0, \;\; x \in \mathcal{C}$$

ℓ_1 penalty is

$$\min_{x \in \mathcal{C}} f(x) + \tau \|r(x)_+\|_1$$

Set

$$c(x) = \begin{bmatrix} f(x) \\ r(x) \\ x \end{bmatrix}, \qquad h(c) = c_1 + \tau \sum_{j=2}^{n_c+1} \max(c_j, 0) + \delta_{\mathcal{C}}(c_{n_c+2:n_c+n+1}).$$

Nonlinear Approximation

$$\min \|c(x)\|,$$

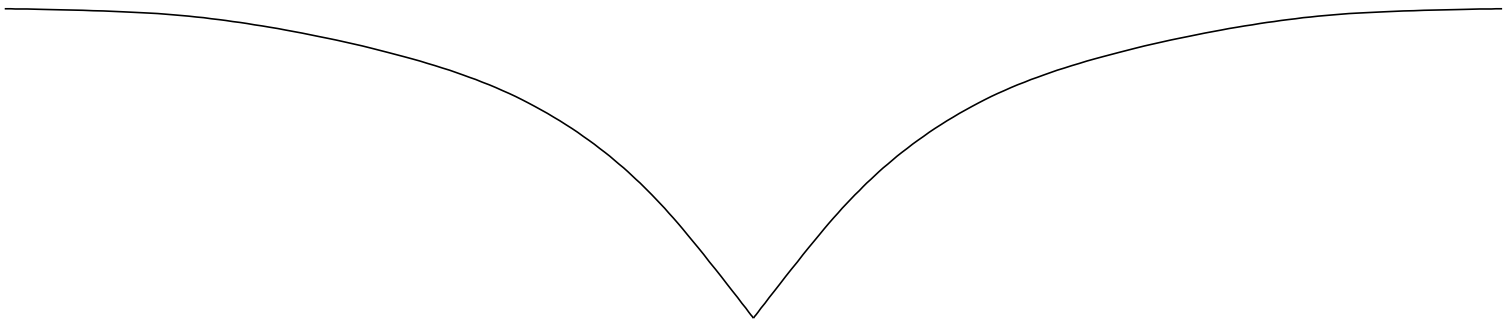
where $\|\cdot\|$ is ℓ_1, ℓ_2, ℓ_∞, or Huber function, for example.

Nonconvex Examples

Alternative to ℓ_1 regularization whre the penalty for large $|x_i|$ is attenuated:

$$\min_x f(x) + \lambda |x|_*, \quad \text{where} \quad |x|_* = \sum_{i=1}^{n} (1 - e^{-\alpha |x|_i}),$$

for some $\alpha > 0$. [Mangasarian, 1999], [Jokar and Pfetsch, 2007]



A similar regularization term is used in Zhang et al (2006) in a support-vector-machines objective.

Proximal Linearized Step

Obtain step d by solving a prox-linearized subproblem:

$$\text{PLS}(x, \mu): \qquad \min_d \; h(c(x) + \nabla c(x) d) + \frac{\mu}{2}|d|^2,$$

for some $\mu > 0$.

- Perturb d if necessary to nearby $\tilde{d}$ to restore finiteness of $h(c(x + \tilde{d}))$.
- Set $x \leftarrow x + \tilde{d}$ if sufficient decrease in $h \circ c$ is obtained; otherwise $\mu \leftarrow \tau\mu$ (for some fixed $\tau > 1$) and re-solve PLS.
- After a successful step, set $\mu \leftarrow \max(\mu_{\min}, \mu/\tau)$.

Approach is suitable when PLS(x, μ) is much easier to solve than the original problem.

Similar to Levenberg-Marquardt, in that the regularization parameter μ is manipulated directly to obtain an acceptable step, *not* a trust region.

The Prox-Linear Subproblem

When applied to compressed sensing, logistic regression, and matrix completion, the step PLS(x, μ) is exactly the SpaRSA step.

ℓ_1 Penalty Function: The subproblem is:

$$\min_{d: x+d \in \mathcal{C}} \nabla f(x)^T d + \frac{\mu}{2}\|d\|_2^2 + \tau\|(r(x) + \nabla r(x)^T d)_+\|_1.$$

... similar to an SLP subproblem with an $\|\cdot\|_2$ trust region.

Nonlinear Approximation

$$\min_d \ \|c(x) + \nabla c(x)^T d\| + \frac{\mu}{2}\|d\|_2^2$$

... extending Levenberg-Marquardt.

Assumption: Prox-Regularity of h

For most results assume prox-regular h: "convex to within a fudge term."

h is *prox-regular at* $\bar{c}$ *for subgradient* $\bar{v}$ if h is finite at $\bar{c}$, locally lower semicontinuous at $\bar{c}$, and there exists $\rho > 0$ such that

$$h(c') \geq h(c) + \langle v, c' - c \rangle - \frac{\rho}{2}\|c' - c\|_2^2$$

for all c', c near $\bar{c}$ with $h(c)$ near $h(\bar{c})$ and $v \in \partial h(c)$ near $\bar{v}$.

h is *prox-regular at* $\bar{c}$ if it is prox-regular at $\bar{c}$ for all subgradients $\bar{v} \in \partial h(\bar{c})$.

e.g. Max of quadratic functions (convex and concave) is prox-regular.

Related Work

SLQP. An approach that uses a similar first-order step (with a different trust region e.g. box-shaped) has been proposed for nonlinear programming / composite nonsmooth minimization [Fletcher, Sainz de la Maza, 1989] [Byrd et al., 2004] [Yuan, 1980s].

Proximal Point. Obtain step from

$$\min_d \; h(c(x_k + d)) + \frac{\mu}{2}\|d\|_2^2.$$

[Marinet, 1970] for convex, lower semicontinuous; generalized by [Rockafellar, 1976] and others. (Doesn't linearize c.)

Casting Functions [Burke, 1985].

$\mathcal{VU}$ Theory and Algorithms. [Lemaréchal, Oustry, Sagastizábal, Mifflin, Miller, Malick, Hare, Daniilidis]

Result: Existence of Solution to PLS

Need a regularity (transversality) condition at critical point $\bar{x}$:

$$\partial^\infty h(\bar{c}) \cap \mathrm{null}(\nabla c(\bar{x})^*) = \{0\},$$

where $\partial^\infty h$ is the "horizon subgradient" consisting of directions along which h grows faster than any linear function.

Need μ larger than a threshold $\bar{\mu}$ that quantifies the nonconvexity of h at $\bar{c} = c(\bar{x})$.

Then for x near $\bar{x}$, we have a local solution d of PLS with $d = O(|x - \bar{x}|)$.

If $x_r \to \bar{x}$ and $\mu_r > \bar{\mu}$, and either $\mu_r |x_r - \bar{x}|^2 \to 0$ or $h(c(x_r)) \to h(c(\bar{x}))$, we have

$$h(c(x_r) + \nabla c(x_r) d_r) \to h(c(\bar{x})).$$

Result: Restoring Feasibility

Curvature in c can cause $h(c(x+d))$ to be infinite even when $h(c(x)+\nabla c(x)d)$ is finite. However can do a small perturbation to restore feasibility.

Assume

- regularity: $\partial^\infty h(\bar{c}) \cap \mathrm{null}(\nabla c(\bar{x})^*) = \{0\}$,
- smoothness of c, h lsc,
- x near $\bar{x}$, d near 0, $h(c(x)+\nabla c(x)d)$ near $h(c(x))$.

Then have $\tilde{d}$ with $|d-\tilde{d}| \leq \gamma |d|^2$ and

$$h(c(x+\tilde{d})) \leq h(c(x)+\nabla c(x)d) + \gamma |d|^2$$

for some $\gamma > 0$.

(Like a second-order correction.)

Result: Multiplier Convergence, Uniqueness

The "multipliers" v_r that satisfy

$$0 = \nabla c(x_r)^* v_r + \mu_r d_r$$
$$v_r \in \partial h(c(x_r) + \nabla c(x_r) d_r)$$

are bounded and converge to a unique value when a stronger condition (analogous to LICQ) holds:

$$\operatorname{par} \partial h(c(\bar{x})) \cap \operatorname{Null} \nabla c(\bar{x})^* = \{0\}.$$

When this condition holds, the PLS solution d_r near 0 is unique.

Active Manifold Identification

In constrained optimization it is useful to be able to identify the active constraints at the solution x^*, before x^* itself is known. Can thus accelerate local convergence, improve robustness of algorithms.

In the setting $h(c(x))$, we look for manifolds in c-space along which h is smooth:

$$\mathcal{M} = \{c \mid h|_{\mathcal{M}} \text{is smooth}\}.$$

When x^* is such that $c(x^*)$ lies on such a manifold, and when we replace

$$\begin{aligned} \text{criticality:} \quad & \partial h(\bar{c}) \cap \text{null}\,(\nabla c(\bar{x})^*) \neq \emptyset \\ \text{by } \textit{strict} \text{ criticality:} \quad & \text{ri}\,\partial h(\bar{c}) \cap \text{null}\,(\nabla c(\bar{x})^*) \neq \emptyset, \end{aligned}$$

(like strict complementarity) along with other conditions, then

$$c(x_r) + \nabla c(x_r) d_r \in \mathcal{M}$$

for all r sufficiently large. Also, stay on $\mathcal{M}$ after the "efficient projection."

ProxDescent: A Descent Algorithm Based on PLS

At iteration k:

- Find a local solution of PLS at x_k and the current μ that improves on $d = 0$;
- "efficiently project" $x_k + d$ onto the domain of h to get x_k^+ (require $(x_k^+ - x_k) \approx d$).
- Increase μ as necessary until a sufficient decrease test is satisfied:

$$h(c(x_k)) - h(c(x_k^+)) \geq .01 \left[h(c(x_k)) - h(c(x_k) + \nabla c(x_k)^T d) \right]$$

- Decrease μ (but enforce $\mu \geq \mu_{\min}$) in preparation for next iteration.

Convergence

Roughly:

- The algorithm can step away from a non-stationary point: The solution of PLS(x, μ) is accepted for μ large enough.
- Cannot have accumulation at noncritical points that are nice (i.e. where h is subdifferentially regular and transversality holds).

See paper for details.

Extensions

- Nonmonotone algorithms? Barzilai-Borwein choices of μ.
- Second-order enhancements. Use the PLS problem to identify a surface, then take a step along that surface with "real" second-order information: Newton-like step for $h(c(x))|_{\mathcal{M}}$.
- Inexact variants.
- Regularizers other than $(\mu/2)|d|^2$.

Conclusions

- Sparse optimization draws on many areas of optimization, linear algebra, and statistics as well as the underlying application areas.
- There is some commonality across different areas that can be abstracted and analyzed.
- Much work remains!