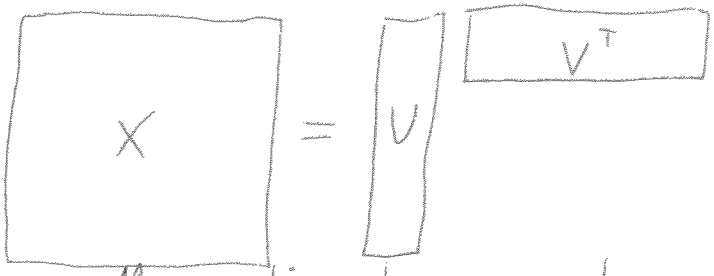
July 21, 2010

① Low-rank matrices, nuclear-norm regularization
② Cauchy bound optization algorithm and singular value thresholding
③ Beyond SVT

Low-rank matrix:

$$X = UV^T$$



Many matrices have a good Low-rank approximation:

$$X \simeq UV^T$$

Advantages of low-rank approximation:
- fewer parameters
- less storage
- faster computation: $Xy = UV^Ty = U(V^Ty)$
- reveals structure (eigenfaces, for example)

Finding a low rank approximation:

$$\min_X f(X) + \lambda \operatorname{rank}(X)$$

- PCA is special case
- In general, hard to solve

SVD of low-rank matrix

$$X = \sum_{i=1}^{\hat{r}} \sigma_i u_i v_i^T,$$

$$\sigma_i = (\sigma_1, \sigma_2, \ldots, \sigma_K, 0, 0, \ldots, 0)$$

"sparse" singular values

Convex Relaxation of rank penalty:

$$\min_X \; f(X) + \lambda \|X\|_* \;, \quad \text{where} \quad \|X\|_* = \sum_{i=1}^{n} \sigma_i$$

- $\|X\|_*$ is a matrix norm, $\Rightarrow$ convex
- Called "nuclear" or "trace" norm
- "L_1-regularization of singular values"
- Non-differentiable if any $\sigma_i = 0$

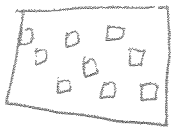
Can generalize to more than one matrix

$$\min_{X_1, X_2, \ldots, X_p} \; f(X_1, X_2, \ldots, X_p) + \sum_{i=1}^{p} \lambda_i \|X\|_*$$

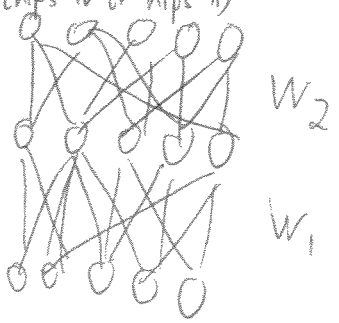
- selects individual matrices ("group" sparsity), since $\text{rank}(X) = 0 \Rightarrow X = 0$
- encourages selected matrices to be low rank

Applications:

- Matrix completion (Netflix)



- Multi-task learning and multinomial classification (low-dimensional feature space shared across tasks)
- Dimensionality reduction
- Transition matrix of HMMs w/ large state-space
- Edge potentials in MRFs ~~(structure learning)~~ (mips '10 or nips '11)
- Weights for layer in DBNs
- Gaussian Graphical models (for $I + X$)



Optimization:

- Can write as semi-definite program
- Solve w/ interior-point methods for ~500 by 500?
- Larger w/ Hessian-free interior-point methods?
- How do we solve Netflix-sized problems?

Cauchy Bound Optimization Algorithm

Consider the unconstrained problem:

$$\min_x \; f(x),$$

where $f(x)$ is twice-differentiable, and eigenvalues of $\nabla^2 f(x)$ are bounded above: $\forall x \; \text{eigs}(\nabla^2 f(x)) \leq L$

(*) Under these conditions, $x^T \nabla^2 f(y) x \leq L x^T x \;\; \forall x \, \forall y$ (e.g. by spectral theorem)

Given iterate x_k, we construct bound function $g(x)$,

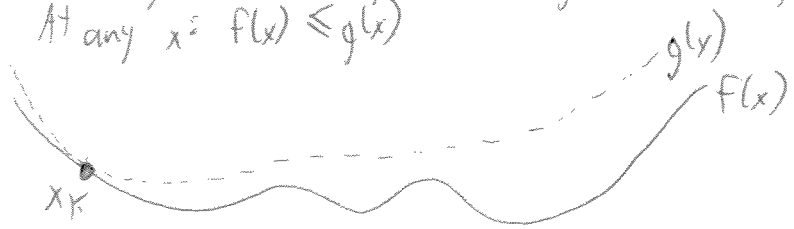
By Taylor's theorem (Lagrange form):

$$f(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k + t(x - x^k))(x - x^k), \quad \text{for } t \in (0, 1)$$

(using (*)) $$\leq \underbrace{f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{L}{2} (x - x^k)^T (x - x^k)}_{g(x)}$$

Note: At x_k, bound is tight: $f(x^k) = g(x^k)$ and gradients match: $\nabla f(x^k) = \nabla g(x^k)$

At any x: $f(x) \leq g(x)$



So if $g(x) < g(x^k)$, then $f(x) \leq g(x) < f(x^k)$
(minimizer of bound improves objective unless $\nabla f(x^k) = 0$)

Bound Optimization:

1) compute $\nabla f(x^k)$ to form $g(x)$
2) set $x^{k+1} \leftarrow \min_x g(x)$

At minimizer, $\nabla g(x) = \nabla f(x^k) + L(x - x^k) = 0$

so $x^{k+1} = x^k - \frac{1}{L} \nabla f(x^k)$

- gradient descent w/ constant step size
- guaranteed to improve objective at every iteration (if $\nabla f(x^k) \neq 0$)

Assumptions can be relaxed:
- can replace twice-differentiable and bounded eigenvalues w/ "gradient is Lipschitz continuous w/ constant L"
- L only needs to hold on sub-level set $\{x \mid f(x) \leq f(x^0)\}$, or only locally
- don't need to know L:
  - diminishing step size $L^k = O(k)$
  - evaluate $f(x^{k+1})$ and ~~decrease~~ increase L if it is too ~~large~~ small
- don't even need exact value of $\nabla f(x^k)$

[redacted]

We can re-write as projected gradient

$$\underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\ q(x) = \underbrace{\frac{1}{2L^2}\nabla f(x^k)^T \nabla f(x^k)}_{\text{constant}} + \frac{1}{L}\nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T(x - x^k)$$

$$= \frac{1}{2}\left\| (x - x^k) + \frac{1}{L}\nabla f(x^k) \right\|_2^2$$

$$= \frac{1}{2}\left\| x - \left(x^k - \frac{1}{L}\nabla f(x^k)\right) \right\|_2^2$$

Ways to speed it up:
- Better approximation $q(x)$ (Newton) [redacted]
- Homotopy (regularization path)
- Extrapolation (Nesterov)

Splitting Methods:
- [Lions and Mercier, 1979], [Moreau, 1962]
- "[Proximal] Forward-Backward Splitting", "iterative soft-thresholding", "composite gradient method", [redacted] "separable approximation", "thresholded Landweber", "truncated gradient", etc.

Proximal Splitting Optimization Algorithm

Consider the unconstrained problem:
$\min_x \; f(x) + \Omega(x)$, where $\Omega(x)$ is "simple" (might be non-smooth, extended real-valued, discontinuous)

and where $f(x)$ is twice-differentiable, and eigenvalues of $\nabla^2 f(x)$ are bounded above. $\forall x \; \text{eigs}(\nabla^2 f(x)) \leq L$

(*) Under these conditions, $x^T \nabla^2 f(y) x \leq L x^T x \;\; \forall x \forall y$ (e.g. by spectral theorem)

Given iterate x_k, we construct bound function $g(x)$,

By Taylor's theorem (Lagrange form):

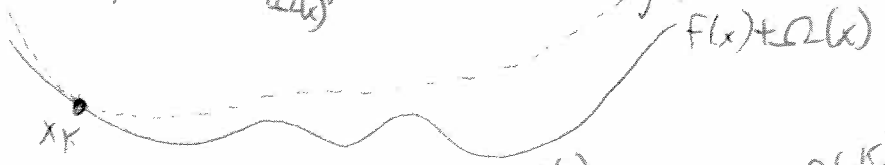
$$f(x) + \Omega(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k + t(x - x^k))(x - x^k) + \Omega(x)$$
for $t \in (0, 1)$

(using (*)) $$\leq f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{L}{2}(x - x^k)^T (x - x^k) + \Omega(x)$$

$g(x)$

Note: At x_k, bound is tight: $f(x^k) + \Omega(x^k) = g(x^k)$

At any x: $f(x) + \Omega(x) \leq g(x)$



So if $g(x) < g(x^k)$, then $f(x) + \Omega(x) \leq g(x) < f(x^k) + \Omega(x^k)$

(improving bound improves objective)

Round Optimization:
1) compute $\nabla f(x^k)$ to form $g(x)$
2) set $x^{k+1} \leftarrow \min_x g(x)$

generalization of
- gradient descent w/ constant step size
- guaranteed to improve objective ~~at every iteration if $\nabla f(x^k) \neq 0$~~ if you improve bound

Assumptions can be relaxed:
- can replace twice-differentiable and bounded eigenvalues w/ "gradient is Lipschitz continuous w/ constant L"
- L only needs to hold on sub-level set $\{x \mid f(x) \leq f(x^0)\}$, or only locally
- don't need to know L:
  - diminishing step size $L^k = O(k)$
  - evaluate $f(x^{k+1})$ and ~~decrease~~ increase L if it is too ~~big~~ small
- don't even need exact value of $\nabla f(x^k)$

~~Reduce to 0~~

We can re-write as proximal gradient

$$\arg\min_{x \in \mathbb{R}^n} g(x) = \underbrace{\frac{1}{2L^2}\nabla f(x^k)^T \nabla f(x^k)}_{\text{constant}} + \frac{1}{L}\nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T(x - x^k) + \frac{1}{L}\Omega(x)$$

$$= \frac{1}{2}\left\| (x - x^k) + \frac{1}{L}\nabla f(x^k) \right\|_2^2 + \frac{1}{L}\Omega(x)$$

$$= \frac{1}{2}\left\| x - (x^k - \frac{1}{L}\nabla f(x^k)) \right\|_2^2 + (\frac{1}{L})\Omega(x)$$

Ways to speed it up:
- Better approximation $g(x)$ (Newton) ~~Barrier Methods~~
- Homotopy (regularization path)
- Extrapolation (Nesterov)

We call $\Omega(x)$ 'simple' if you can efficiently solve proximal problem:

$$P[v] \triangleq \min_u \frac{1}{2}\|u - v\|_2^2 + (\frac{1}{L})\Omega(u)$$

Examples:

- L_1-regularization: $\Omega(x) = \lambda \|x\|_1$ (non-smooth)
- Group L_1-regularization: $\Omega(x) = \lambda \sum_g \|w_g\|_2$
- Overlapping Group-L_1 w/ tree-structured groups
- L_0-regularization: $\Omega(x) = \lambda \|x\|_0$ (discontinuous)

[redacted]

- Any separable regularizer
- Indicator for simple convex set: $\Omega(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{if } x \notin C \end{cases}$

(in this case we obtain projected gradient)

L_1-regularization

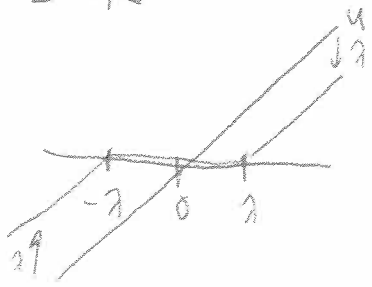
$$\min_v \|u - v\|_2^2 + \lambda \|v\|_1$$

$$\partial |v| = \begin{cases} 1 & \text{if } v > 0 \\ -1 & \text{if } v < 0 \\ [-1, 1] & \text{if } v = 0 \end{cases}$$

v_* optimal $\iff 0 \in (u - v) + \lambda \partial [\|v\|_1]$

Solution:

$$\begin{cases} v_i = u_i - \lambda & \text{if } u > \lambda \\ v_i = u_i + \lambda & \text{if } u < -\lambda \\ v_i = 0 & \text{if } -\lambda \le u \le \lambda \end{cases}$$



$$v_i = \text{sgn}(u_i)[|u_i| - \lambda]^+$$

This is the 'soft-threshold' operator.

In general, sparse regularizers lead to sparse iterations.

Group L_1-regularization

$$v_g = \text{sgn}(u_g)[\|u\|_2 - \lambda]^+$$

Nuclear Norm

$$X = U \, \text{diag}([\sigma - \lambda]^+) V^T$$

"Singular value thresholding"

Beyond basic SVT

Recent papers include tricks to scale to larger problems.

[Cai, Candes, Shen, 2008]: only need singular values $> \tau$, so use Lanczos (power method + re-orthogonalization)

[Goldfarb, Ma, Shen, 2009]: randomized linear algebra for linear-time SVD, acceleration using homotopy

[Mazumder, Hastie, Tibshirani, 2009]: consider different bound for quadratic case that doesn't need step size

[Toh and Yun, 2009] [Ji and Ye, 2009]: acceleration using Nesterov's extrapolation method.

[Tomioka, Suzuki, Sugiyama, Kashima, 2010]: Dual augmented Lagrangian to speed convergence.

[Jaggi, Sulovsky, 2010]: Core-set approach for approximate SDP (Hazan), only requires largest $\hat{\sigma}_i$.