Optimization in Machine Learning

Daniel L Boley
University of Minnesota

How Convex Optimization plays a big role in Computer Science.

NSF Grant 1319749

Discovery Problems

- Many large pattern discovery problems depend on representing each data sample as a vector in high-dimensional euclidean space.
  - Text documents (news, laws, WWW documents).
  - Gene expression profiles
  - Attributes for individual people, transactions, locations, ecosystems, ...
  - Images

  } tabular

  - Gene-gene or protein-protein interaction networks
  - WWW connectivity graph
  - Computer inter-connect in Internet
  - People-people affinities in Social Media

  } graph

- Many example datasets can easily have up to $O(10^{9+})$ data points.
- Many datasets have much noise or many attributes.
- Many example datasets are sampled, subject to sampling bias.

Tools to Explore

- Dimensionality Reduction
  - Represent each data sample with a reduced set of attribute values
  - Minimize loss of information
  - Implicit assumption: data is subject to some level of noise.
  - Want to preserve some structure (e.g., certain entries known exactly).
- Clustering
  - unsupervised: no labeled training set.
  - group together items that "close" to each other in some sense
  - separate items that are "far" from each other
  - might have some known constraints.
- Build Classifier
  - Use fully or partially labeled training set to build classifier
  - a classifier is just a function mapping a vector of attributes to a class identifier.
  - example: a nearest neighbor classifer takes an attribute vector input, finds the closest vector in the training set, and assigns the latters label as a class ID.

Tools for Other Kinds of Data.

- Graph Properties
  - represent connectivity between entities with links.
  - identify important nodes or links
  - partitioning: cut graph into cohesive chunks.
  - aggregrate properties: volume, distribution of properties across nodes.
- Sparse Representation
  - Have derived a dictionary of data components.
  - Seek to represent each data sample as a combination of only a few components.
  - Hard to interpret individual components in traditional dimensionality reduction methods.
  - Possibly also seek to represent each component as a combination of only a few original attributes.
  - Maintain desire for small approximation error.

Outline

- Sparse Representation – Examples
  - almost shortest path routing.
  - constrained clustering.
  - image/vision,
  - Graph Connection Discovery.
- Finding Sparse Representation
  - Set up as an optimization problem
  - Mathematical formulation: relax to convex problem.
- Solvers
  - Alternating Direction Method of Multipliers
  - Alternatives for L1 Regularized Least Squares

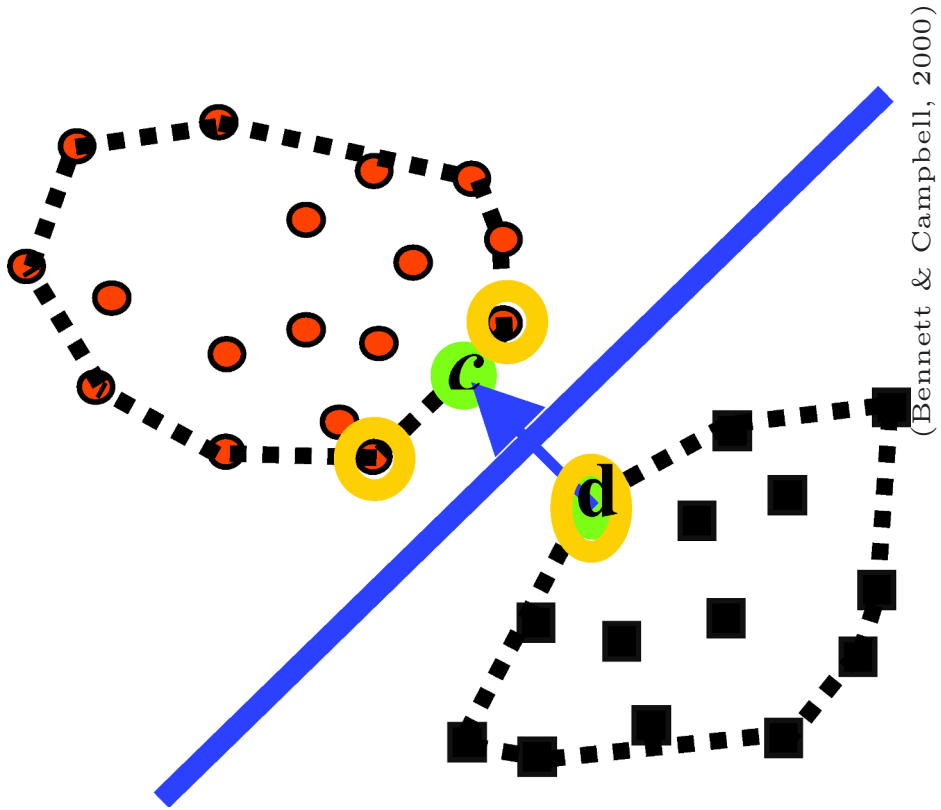
Outline

- Sparse Representation – Examples
  - almost shortest path routing.
  - constrained clustering.
  - image/vision,
  - Graph Connection Discovery.
- Finding Sparse Representation
  - Set up as an optimization problem
  - Mathematical formulation: relax to convex problem.
- Solvers
  - Alternating Direction Method of Multipliers
  - Alternatives for L1 Regularized Least Squares

SVM: Maximum Margin Separator

- $\mathbf{x}_i = i$-th training attribute vector, $y_i = \pm 1 =$ corresponding label.
- Support Vector Machines (Bennett & Campbell, 2000)

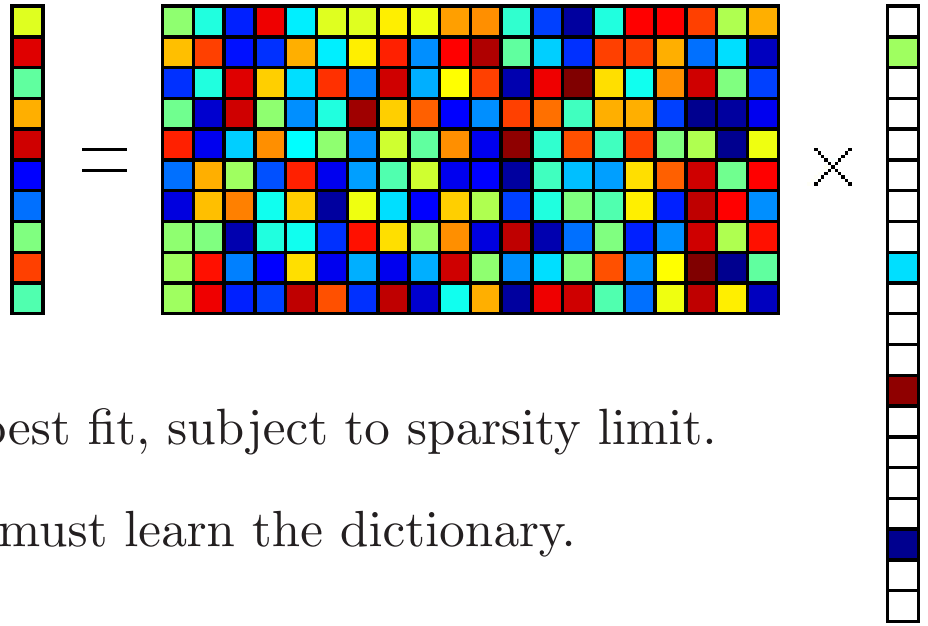
$$\begin{array}{ll} \text{minimize} & \|\mathbf{c} - \mathbf{d}\|_2^2 \\ \text{subject to} & \mathbf{c} = \sum_{i:y_i=1} \alpha_i \mathbf{x}_i, \ \mathbf{d} = \sum_{i:y_i=-1} \alpha_i \mathbf{x}_i \\ & \sum_{i:y_i=1} \alpha_i = 1, \ \sum_{i:y_i=-1} \alpha_i = 1, \ \alpha_i \geq 0 \end{array}$$



Sparse Representation

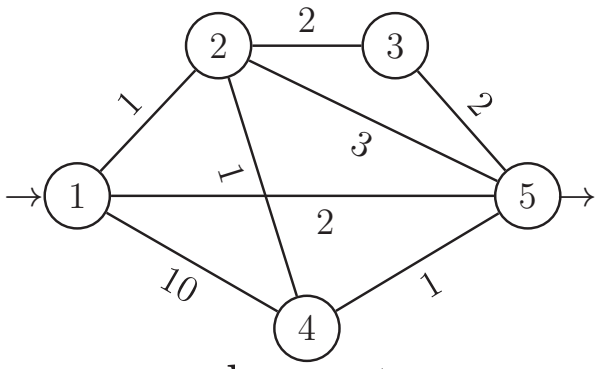
- Many machine learning algorithms can explore massive data:
  K-nearest Neighbors, Kernal-SVM, Boosting, Metric Learning, . . .
- All can benefit from denoising by finding a sparse representation:

raw datum = dictionary atoms × sparse representation

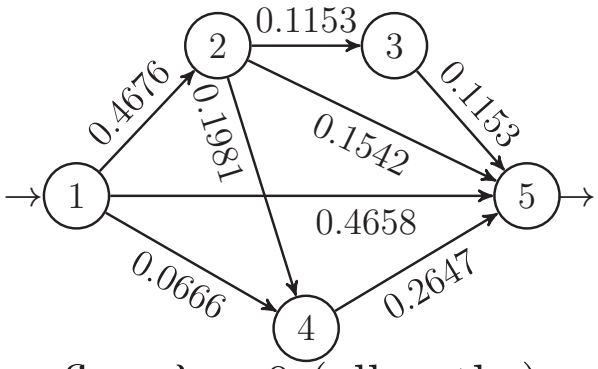


- Must find best fit, subject to sparsity limit.
- Optionally must learn the dictionary.

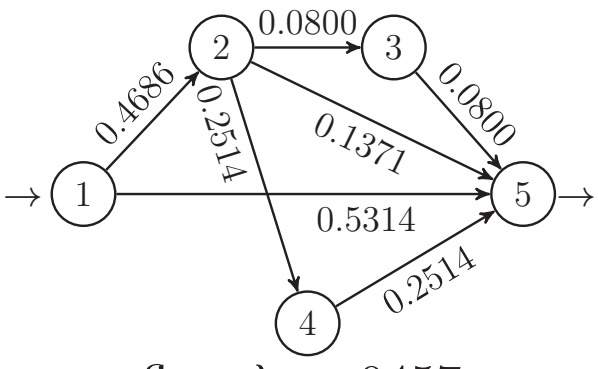
Almost Shortest Path Routing



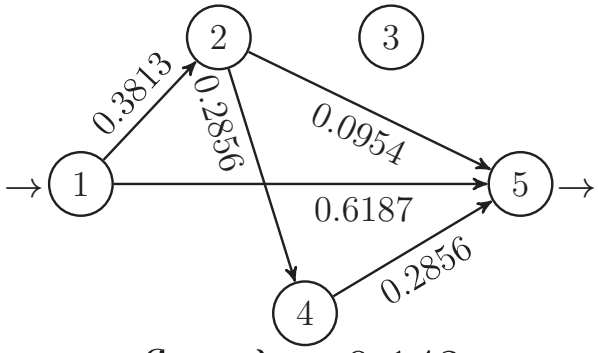
edge costs



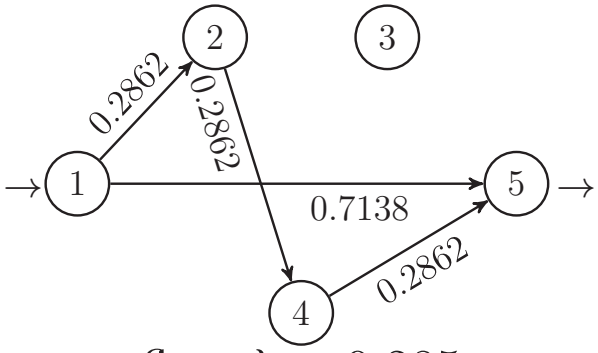
flow $\lambda = 0$ (all-paths)



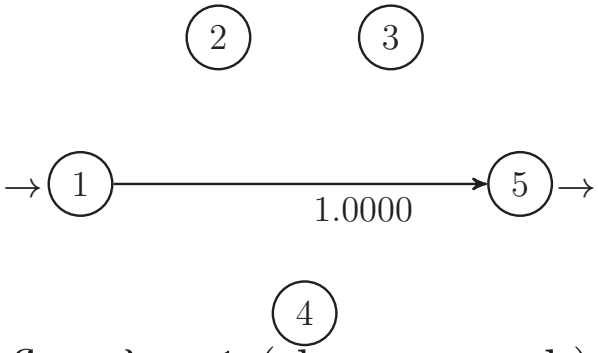
flow $\lambda = .0457$



flow $\lambda = 0.143$



flow $\lambda = 0.285$



flow $\lambda = 1$ (shortest path)

$\min_{\mathbf{x}} \mathbf{x}^T\mathbf{W}\mathbf{x} + \lambda\|\mathbf{x}\|_1 = \sum_{ij\in E} X_{ij}^2 w_{ij} + \lambda|X_{ij}|$ minimize total flow energy

s.t. $\sum_{i:\, ik\in E} X_{ik} = \sum_{j:\, kj\in E} X_{kj} \quad \forall k$ flow in = flow out at every node k

(Li et al., 2011)

Constrained Clustering

- Graph Clustering with *Must-link* and *Cannot-link* constraints.
- Let $\mathbf{x}$ be the indicator vector for the clustering
  ($x_i = +\alpha \,|\, -\beta$ depending on membership in $\mathcal{C}_+ \,|\, \mathcal{C}_-$).
- Spectral Graph Cut: $= \mathbf{x}^T\mathbf{L}\mathbf{x}$ [where $\mathbf{L}$ = Laplacian].
- Constraints represented by a subgraph with Incidence matrix $\mathbf{C_c}$ and Laplacian $\mathbf{L_c} = \mathbf{C_c}^T\mathbf{C_c}$.
- Previous approach: minimize $\mathbf{x}^T\mathbf{L}\mathbf{x} + \lambda\mathbf{x}^T\mathbf{L_c}\mathbf{x}$ s.t. $\mathbf{x}^T\mathbf{x} = 1$ (Shi et al., 2010).
- Our approach: minimize cut with $L1$ penalty on constraint violations:
  $\mathbf{x}^T\mathbf{L}\mathbf{x} + \lambda\|\mathbf{C_c}\mathbf{x}\|_1$ s.t. $\mathbf{x}^T\mathbf{x} = 1$ [Kawale et al].

=== Fused LASSO:

- Analogous: $\|A\mathbf{x} - \mathbf{b}\|^2 + C\mathbf{x}$, where C is a 1st or 2nd difference operator.

Image Descriptors

Image Descriptor

- Pixel Descriptors: for i-th pixel $z_i = \phi(x_i, y_i)$ is a vector of descriptors for the pixel at point (x_i, y_i) in the image.
- Example, could use $z_i = (I_x, I_y, |\mathsf{grad} I|, \angle_{\mathsf{grad}} I, I_{xx}, I_{xy}, I_{yy})$ where I is the intensity value. Could also incorporate color information.

Covariance Descriptor (Tuzel et al., 2006)

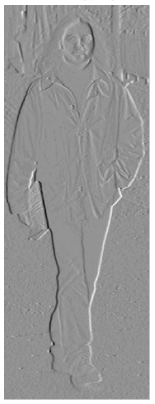
- Within each small patch around each pixel compute the covariance C_i of the pixel descriptors.
- Covariance descriptors eliminate differences due to scaling, brightness, large shadows, but enhance local features.
- Use for object detection, tracking, recognition, and more ...
- Each C_i is a small positive semi-definite matrix (7×7 in this example).
- Regularize each C_i by adding a small multiple of the identity.

Covariance Descriptor Example

Raw Image

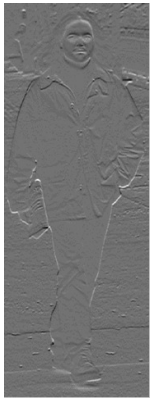
Image

first derivatives



x-grad

grad-mag



y-grad

grad-dir

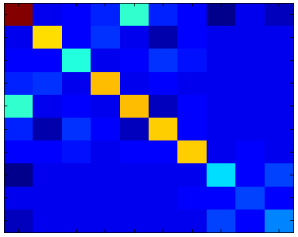
second derivatives

Dxx

Dxy

Dyy

pixel by pixel descriptor



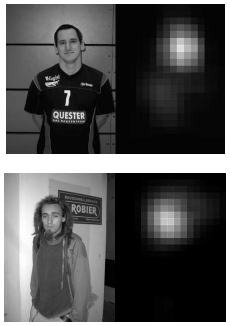
Covariance descriptor

Covariance Descriptor Usage

- **Object Detection and Tracking in Image.**

Object Detection

face

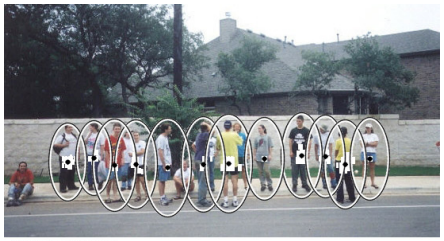


(Opelt et al., 2004; Sivalingam et al., 2011)

license plate

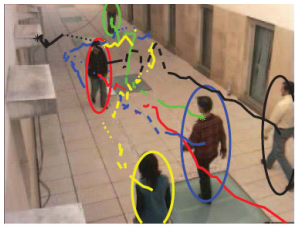
(Porikli & Kocak, 2006)

human



(Tuzel et al., 2007)

Object Tracking



(Palaio et al., 2009)

Object Recognition

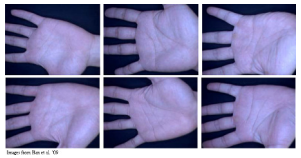
face

(Pang et al., 2008)

action

KTH dataset

palmprint



(Han et al., 2009)

Optimization Setup for Covariances

Notation: (Sivalingam et al., 2010; Sivalingam et al., 2011)

- S = a raw covariance matrix,
  $\mathbf{x}$ = vector of unknown coefficients.
  $\mathcal{A} = (A_1, A_2, \ldots, A_k)$ = collection of dictionary atoms.
  $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ = vector of unknown coefficients.

- Goal: Approximate $S \approx A_1 x_1 + \cdots + A_k x_k = \mathcal{A} \cdot \mathbf{x}$.

- Use "logdet" divergence as measure of discrepancy:
  $D_{\mathrm{ld}}(\mathcal{A} \cdot \mathbf{x}, S) = tr((\mathcal{A} \cdot \mathbf{x})S^{-1}) - \log\det((\mathcal{A} \cdot \mathbf{x})S^{-1}) - n.$

- Logdet divergence measures relative entropy between two different zero-mean multivariate Gaussians.

Optimization Problem for Covariances

(Sivalingam et al., 2010; Sivalingam et al., 2011)

- Leads to optimization problem

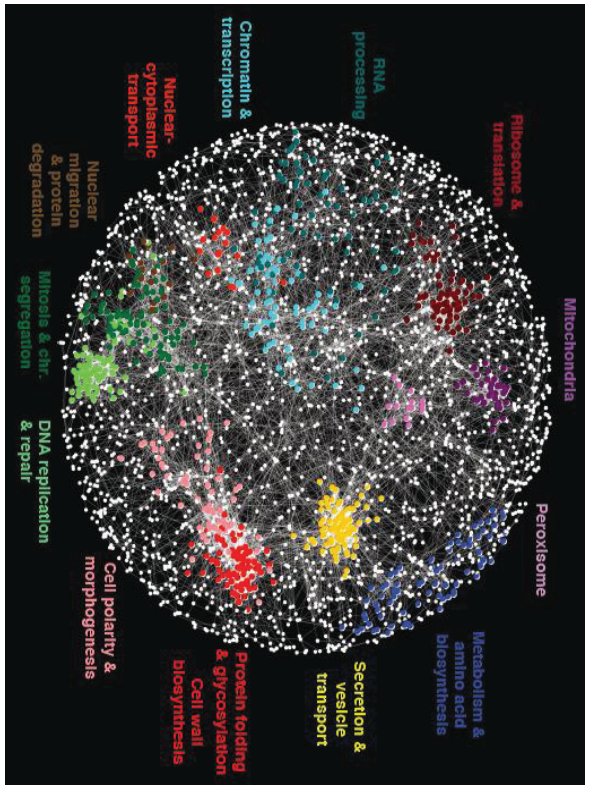
$$\begin{array}{ll} \min_{\mathbf{x}} & \underbrace{\sum_i x_i tr(A_i) - \log\det\left[\sum_i x_i A_i\right]}_{\text{Dist}(\mathcal{A}\cdot\mathbf{x},S)} + \underbrace{\lambda \sum_i x_i}_{\text{sparsity}} \\ \text{s.t.} & \mathbf{x} \geq 0 \\ & \sum_i x_i A_i \succeq 0 \quad \text{(positive semi-definite)} \\ & \sum_i x_i A_i \preceq S \quad \text{(residual positive semi-def.)} \end{array}$$

- This is in a standard form for a MaxDet problem.
- The sparsity term is a relaxation of true desired penalty: # nonzeros in $\mathbf{x}$.
- Convex problem solvable by many solvers.

Graph Connections Discovery

- Signal at node i is gaussian & correlated to neighbors, but conditionally independent of unconnected node j.
- Statistical Theory $\Longrightarrow (\text{Covariance})^{-1}_{ij} = 0$.
  $(\text{Covariance})^{-1}$ is called the Precision Matrix.
- If graph is sparse, expect $(\text{Covariance})^{-1}$ to be sparse.
- Problem: Graph connections are unknown.
- Task: Given signals at each node, recover graph edges.
- Applications: biology, climate modelling, social networks.

[Myers]



- Method:
  - Compute sample precision matrix from signals.
  - Find best ***sparse*** approximation to sample precision matrix.
  - Use previous log-det divergence to measure discrepancy between covariance matrices.

Sparse Inverse of Positive Definite Matrix

- Seek sparse approximation X to inverse of positive definite S.
- Use measure $\phi(X) = \mathrm{Tr}(SX - I) - \log\det(SX) = \mathrm{Tr}(SX) - \log\det X -$ constants.
- $\phi(X) = 0$ when $X = S^{-1}$.

=== Properties: First Variation:

- $\phi(X + \Delta) = \phi(X) + \mathrm{Tr}((S - X^{-1})\Delta) - \mathrm{Tr}(X^{-1}\Delta X^{-1}\Delta) + O((\Delta^3)$.
- Linear term is $\langle (S - X^{-1})^T, \Delta \rangle$, zero when $S = X^{-1}$.
- If X pos. def., Quadratic term is $\mathrm{Tr}\left(X^{-1/2}\Delta X^{-1}\Delta X^{-1/2}\right) > 0$.
- For sparse inverse, minimize $\phi(X) + \lambda|X|_1$ (where $|X|_1 = \sum |x_{ij}|$).
- Newton's method: around iterate X, minimize *wrt* Δ.
- Each inner iteration is an ℓ_1 regularized quadratic $\Longrightarrow$ LASSO.

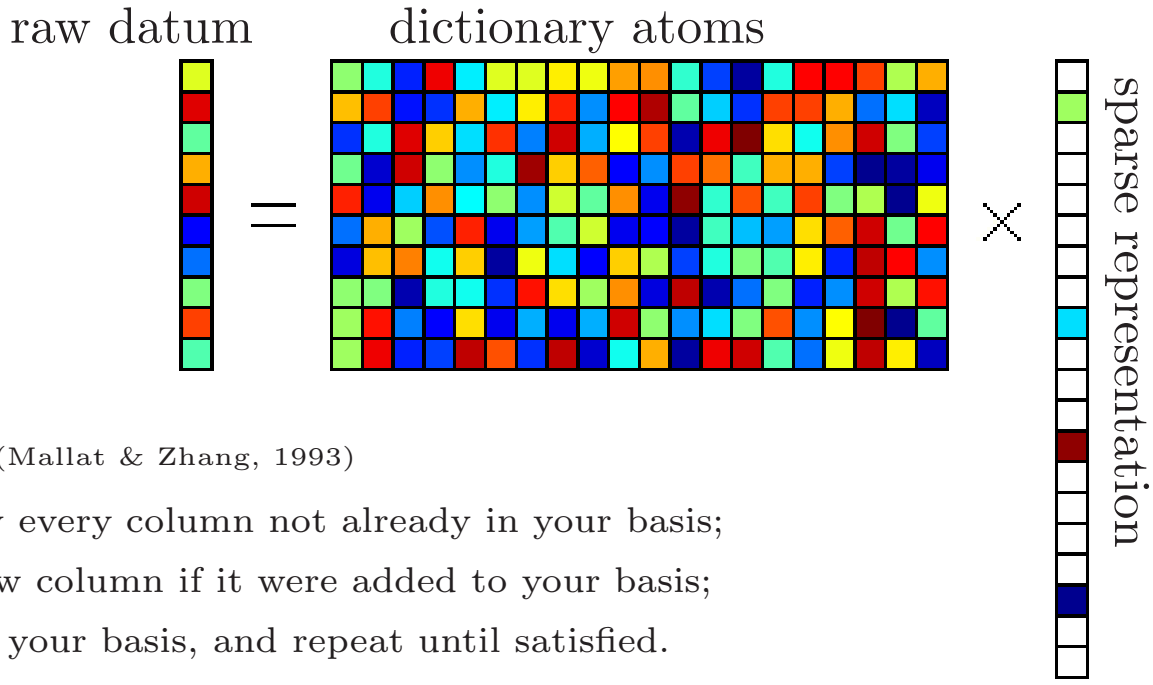
Matrix Completion

- Application: Recommend movies for users by filling in matrix of movie ratings (Netflix problem), Missing answers in a questionnaire.
- Application: Low dimensional embedding: have partial table of distances between wireless sensors, want to fill in missing distances.
- Model: Assume true matrix is low rank: all user behavior can be grouped into a combination of a small number of distinct primitive behaviors.
- Problem: Find low rank matrix X whose entries matches the known entries in the data matrix M.
- Leads to: minimize $\text{rank}(X)$ subject to $X_{ij} = M_{ij}$, for $(i,j) \in \Omega$ = set of known entries.
- Relax to: minimize $\|X\|_*$ subject to $X_{ij} = M_{ij}$, for $(i,j) \in \Omega$, where $\|\cdot\|_*$ is the *nuclear norm* (sum of singular values). (Candés & Recht, 2008)

Outline

- Sparse Representation – Examples
  - almost shortest path routing.
  - constrained clustering.
  - image/vision,
  - Graph Connection Discovery.
- Finding Sparse Representation
  - Set up as an optimization problem
  - Mathematical formulation: relax to convex problem.
- Solvers
  - Alternating Direction Method of Multipliers
  - Alternatives for L1 Regularized Least Squares

Constructing Sparse Basis



- Matching Pursuit: (Mallat & Zhang, 1993)
  - Greedy algorithm: try every column not already in your basis;
  - evaluate quality of new column if it were added to your basis;
  - add "best" column to your basis, and repeat until satisfied.
- Basis Pursuit (Chen et al., 2001)
  - Minimize $\|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0$.
  - Difficulty: this is a NP-hard combinatorial problem.
  - Relax to $\|\mathbf{b} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$.
  - Relaxed problem is convex, so solvable more efficiently.
  - LASSO: Solve for all λ fast (Tibshirani, 1996).
- Non-linear Problem
  - Use Newton's method: inner loop $\equiv$ LASSO Problem.

Convex Relaxation $\Longrightarrow$ LASSO

- Known as Basis Pursuit, Compressed Sensing, "small error + sparse".
- Add penalty for number of nonzeros with weight λ:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_0.$$

- Convert hard combinatorial problem into easier convex optimization problem.
- Relax previous $||\mathbf{x}||_0$ to convex problem:

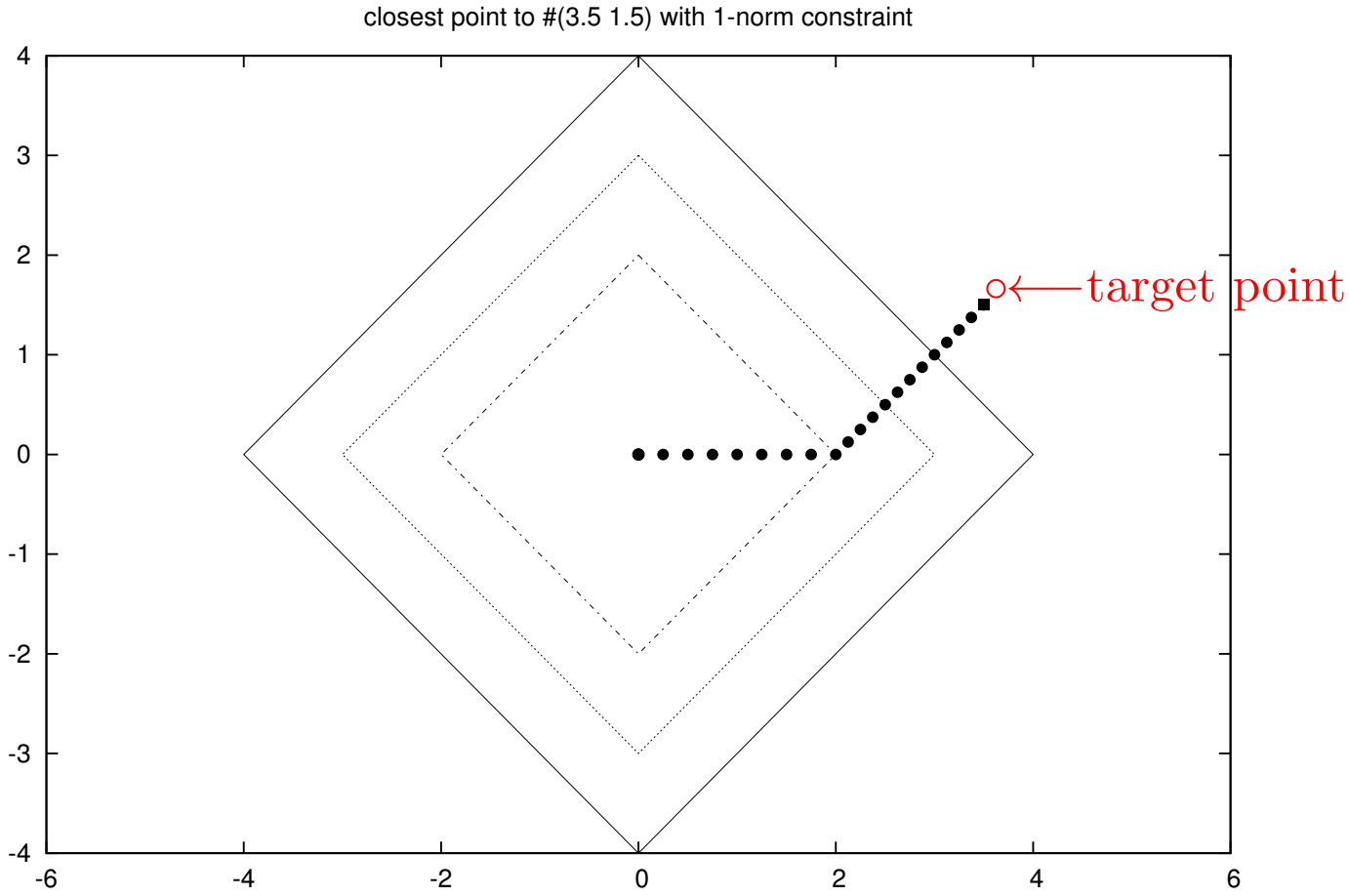
$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1,$$

- or convert to constrained problem:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq \texttt{tol}.$$

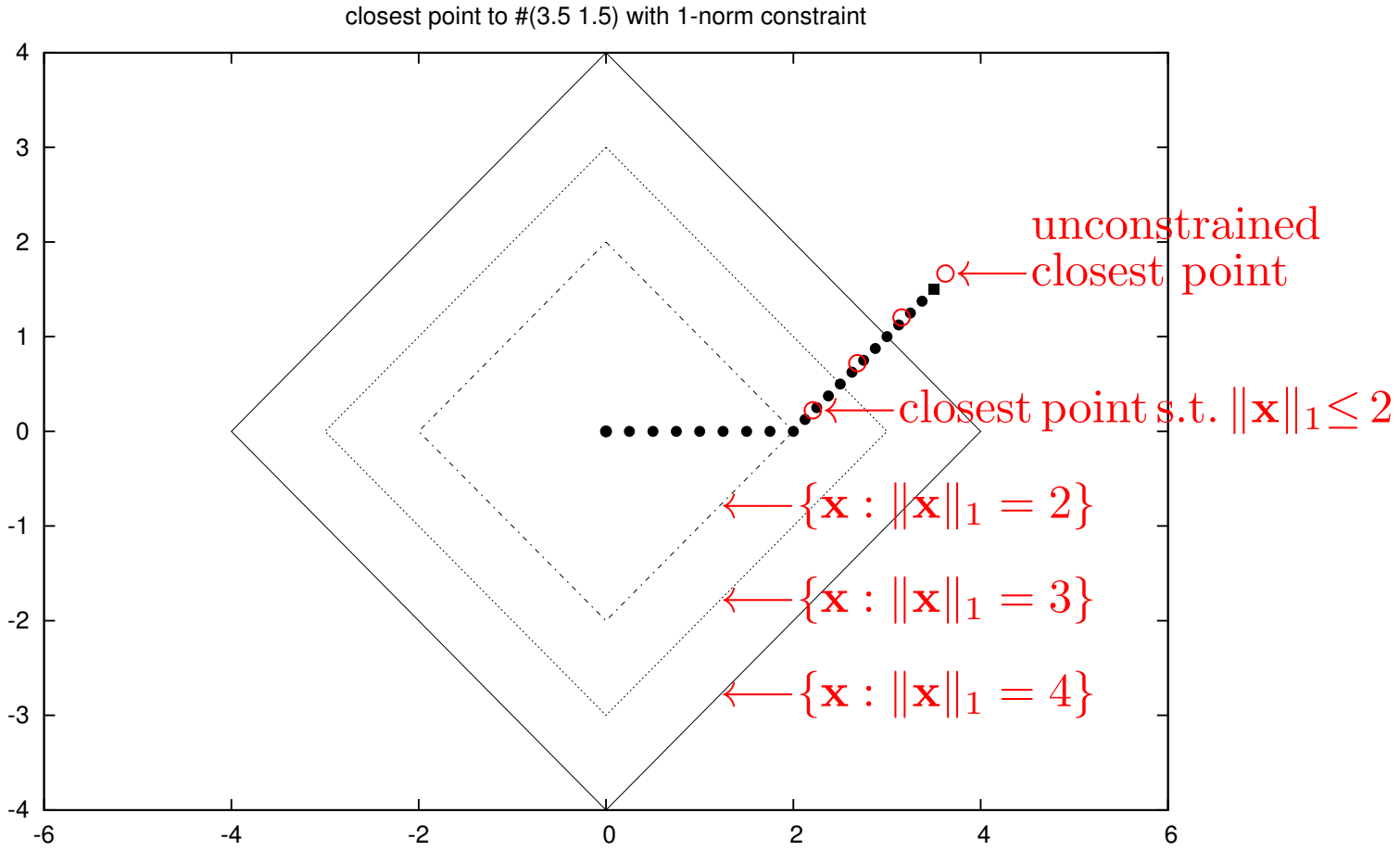
- Vary parameter λ or `tol`, to explore the trade-off between "small error" and "sparse".

Motivation: find closest sparse point



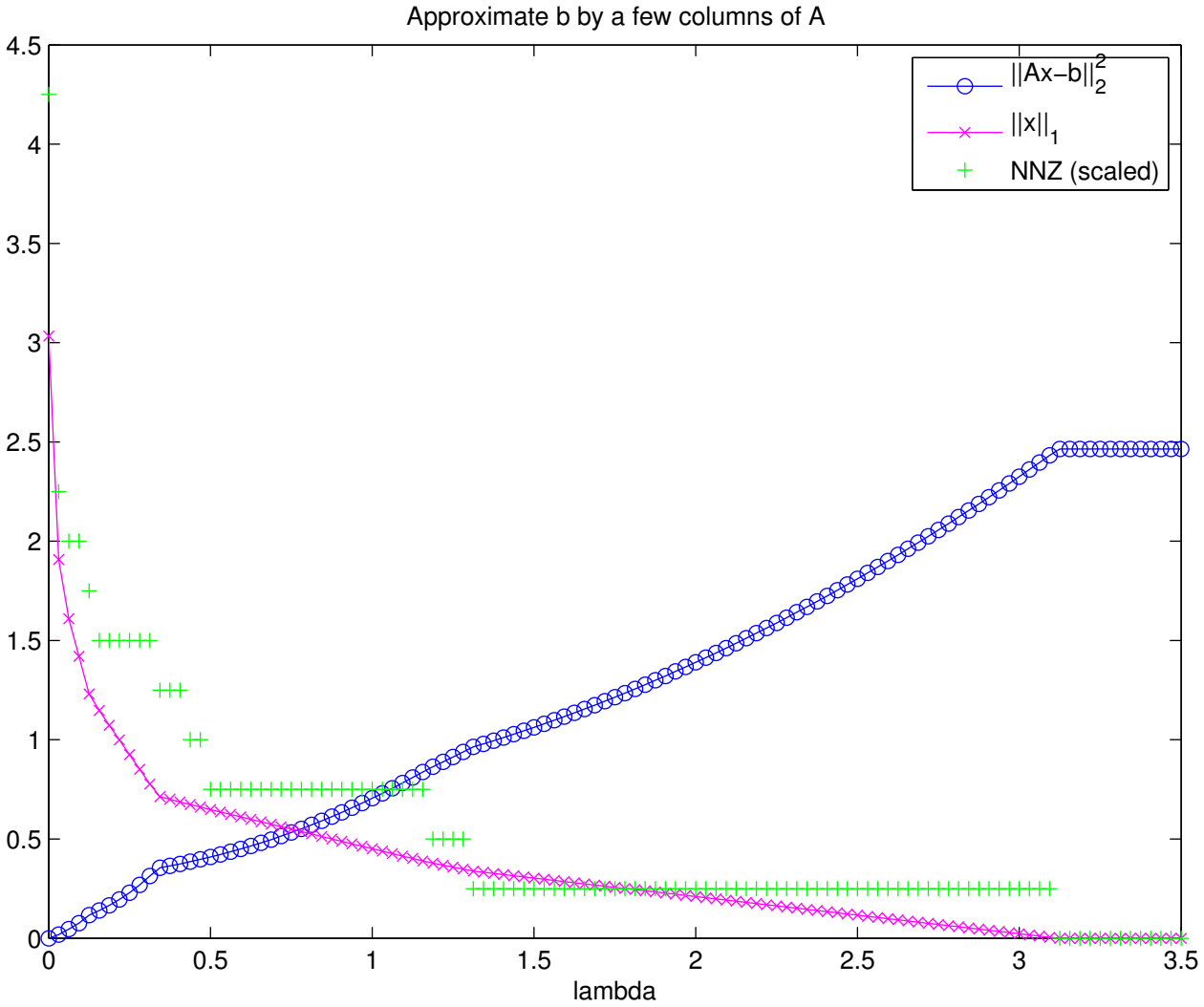
- Find closest point to target . . . subject to ℓ_1 norm constraint.

Motivation: find closest sparse point



- As limit on $\|\mathbf{x}\|_1$ is tightened, the coordinates are driven toward zero.
- As soon as one coordinate reaches zero, it is removed, and the remaining coordinates are driven to zero.

Example: 17 signals with 10 time points



- As λ grows, the error grows, fill (#non-zeros) shrinks.

Methods

- All problems are convex.
- Must work exists on software for convex programming problems
- YALMIP is a front end with links to many solver packages (Löfberg, 2004).
- CVX is a free package of convex solvers with easy matlab interface (Grant & Boyd, 2010).
- ADMM is a paradigm for a simple iterative solver especially adapted for very large but separable problems (Boyd et al., 2011).

Outline

- Sparse Representation – Examples
  - almost shortest path routing.
  - constrained clustering.
  - image/vision,
  - Graph Connection Discovery.
- Finding Sparse Representation
  - Set up as an optimization problem
  - Mathematical formulation: relax to convex problem.
- Solvers
  - Alternating Direction Method of Multipliers
  - Alternatives for L1 Regularized Least Squares

Local Linear Convergence of ADMM

Model QP/LP: $\min \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, (1)

Lagrangian: $\mathcal{L}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{x} - \mathbf{v}^T(A\mathbf{x} - \mathbf{b})$, (2)

where $\mathbf{y} \geq 0$ is the vector of Lagrange multipliers for the inequality constraints $\mathbf{x} \geq 0$.

Previous Convergence Theory

- Very abstract theory based on monotone linear operators.
- Recent results are of the form $O(k)$ or $O(k^2)$, where k = iteration number.
- Bounds are far from actual behavior.

Dual Ascent Method

Model QP/LP: $\min \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, (1)

Lagrangian: $\mathcal{L}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{x} - \mathbf{v}^T(A\mathbf{x} - \mathbf{b})$, (2)

where $\mathbf{y} \geq 0$ = Lagrange multipliers for the constraints $\mathbf{x} \geq 0$.

Primal Problem: $\min_{\mathbf{x}} \boxed{\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})}$: $\boxed{\cdots} = \infty$ when constraints violated.

Dual Problem: $\max_{\mathbf{y}} \boxed{\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y})}$: $\boxed{\textit{boxed expr}}$ is relatively easy to solve.

Dual Ascent Method: solve $\boxed{\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y})}$ in dual problem exactly, take small gradient ascent steps on dual variable $\mathbf{y}$.

Split Primal variables into x, z:

$$\min \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + g(\mathbf{z}) \text{ s.t. } A\mathbf{x} = \mathbf{b},\ \mathbf{x} = \mathbf{z}, \tag{3}$$

where $g(\mathbf{z})$ is the indicator function for the non-negative orthant:

$$g(\mathbf{z}) = \begin{cases} 0 & \text{if } \mathbf{z} \geq 0 \\ \infty & \text{if any component of } \mathbf{z} \text{ is negative.} \end{cases}.$$

$g(\mathbf{z})$ is a non-smooth convex function encoding the inequality constraints.

Partially augmented Lagrangian

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{x} - \mathbf{z}) + \tfrac{1}{2}\rho\|\mathbf{x} - \mathbf{z}\|_2^2, \quad \text{s.t. } A\mathbf{x} = \mathbf{b}, \tag{4}$$

where $\mathbf{y}$ is now the vector of Lagrange multipliers for the additional equality constraint $\mathbf{x} - \mathbf{z} = 0$, ρ is a proximity penalty parameter chosen by the user.

Splitting

Using the common splitting (Boyd et al., 2011), the ADMM method consists of three steps: first minimize Lagrangian with respect to $\mathbf{x}$, then with respect to $\mathbf{z}$, and then perform one ascent step on the Lagrange multipliers $\mathbf{u}$:

$$\begin{array}{lll} 1. \quad \text{Set } \mathbf{x}^{[k+1]} & = & \operatorname{argmin}_{\mathbf{x}} \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + \tfrac{1}{2}\rho\mathbf{x}^T\mathbf{x} + \rho\mathbf{x}^T(\mathbf{u}^{[k]} - \mathbf{z}^{[k]}) \\ & & \text{subject to } A\mathbf{x} = \mathbf{b} \\ 2. \quad \text{Set } \mathbf{z}^{[k+1]} & = & \operatorname{argmin}_{\mathbf{z}} g(\mathbf{z}) + \tfrac{1}{2}\rho\mathbf{z}^T\mathbf{z} - \rho\mathbf{z}^T(\mathbf{x}^{[k+1]} + \mathbf{u}^{[k]}) \\ 3. \quad \text{Set } \mathbf{u}^{[k+1]} & = & \mathbf{u}^{[k]} + \nabla_{\mathbf{u}}\mathcal{L}_\rho(\mathbf{x}^{[k+1]}, \mathbf{z}^{[k+1]}, \mathbf{u}). \end{array} \tag{5}$$

Closed Form

Each step of Alg I can be solved in closed form, leading to the ADMM iteration (with no acceleration) consisting of the following steps repeated until convergence, where $\mathbf{z}^{[k]}, \mathbf{u}^{[k]}$ denote the vectors from the previous pass, and ρ is a given fixed proximity penalty:

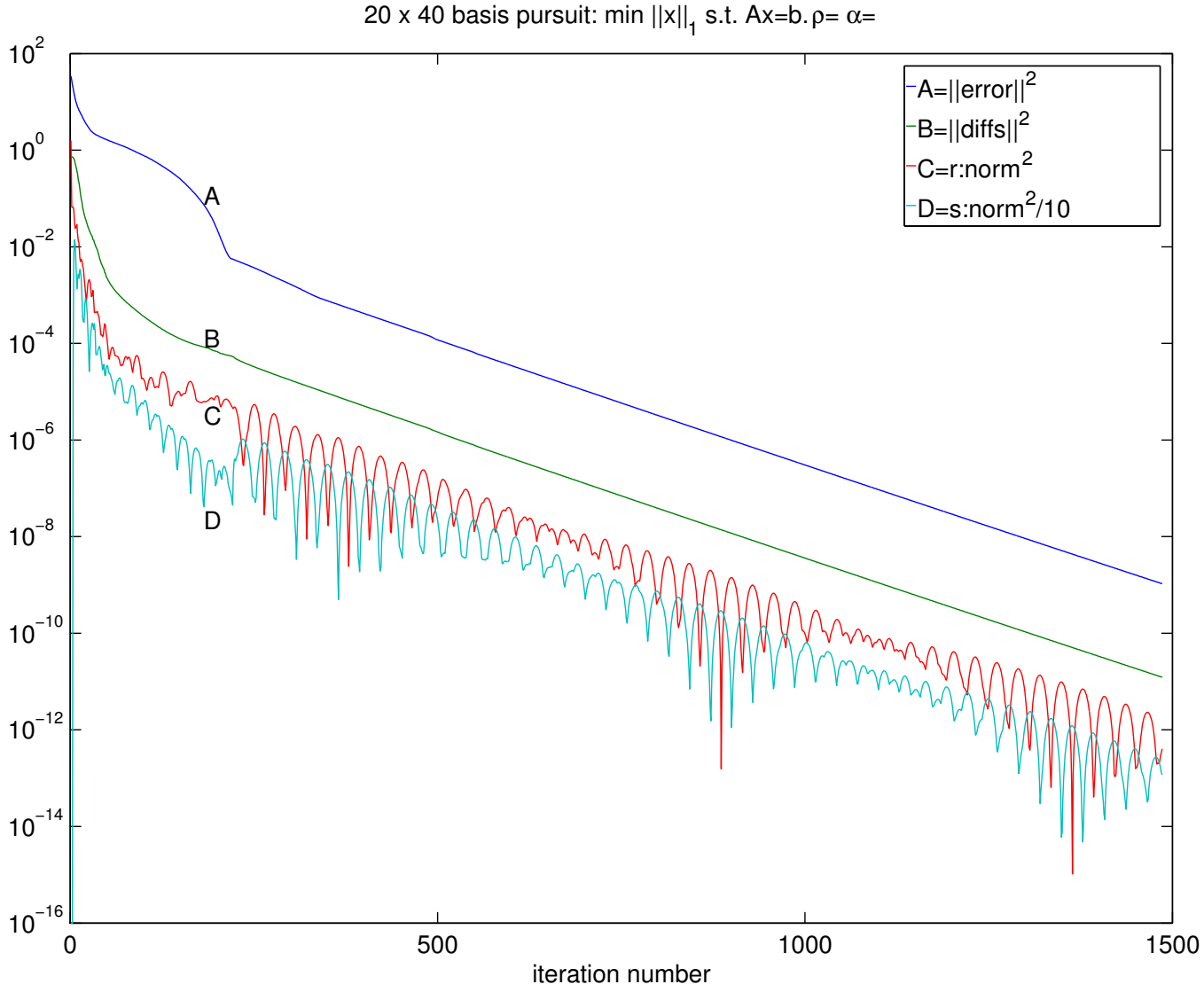
Algorithm 1: One Pass of ADMM

Start with $\mathbf{z}^{[k]}, \mathbf{u}^{[k]}$.

1. Solve $\begin{pmatrix} Q+\rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{[k+1]} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \rho(\mathbf{z}^{[k]} - \mathbf{u}^{[k]}) - \mathbf{c} \\ \mathbf{b} \end{pmatrix}$ for $\mathbf{x}^{[k+1]}, \boldsymbol{\nu}$.
2. Set $\mathbf{z}^{[k+1]} = \max\{0, \mathbf{x}^{[k+1]} + \mathbf{u}^{[k]}\}$ (where "max" is taken elementwise).
3. Set $\mathbf{u}^{[k+1]} = \mathbf{u}^{[k]} + \mathbf{x}^{[k+1]} - \mathbf{z}^{[k+1]}$.

Result is $\mathbf{z}^{[k+1]}, \mathbf{u}^{[k+1]}$ for next pass.

Sample ADMM Convergence Trace on an LP



Complementarity Property

Lemma 1. *After every pass, the vectors* $\mathbf{z}^{[k+1]}, \mathbf{u}^{[k+1]}$ *satisfy*

a. $\mathbf{z}^{[k+1]} \geq 0$,

b. $\mathbf{u}^{[k+1]} \leq 0$,

c. $z_i^{[k+1]} \cdot u_i^{[k+1]} = 0$, $\forall i$ *(a complementarity condition).*

d. $\mathbf{x}^{[k+1]}$ *satisfies the equality constraints* $A\mathbf{x}^{[k+1]} = \mathbf{b}$.

- Combine into a single vector $\mathbf{w} = \mathbf{z} - \mathbf{u}$.
- Use auxiliary flag vector $\mathbf{d}$ to indicate whether $w_i = z_i$ or $w_u = -u_i$.
- Previous iteration is linear in $\mathbf{w}$ as long as $\mathbf{d}$ is fixed.

ADMM as a Matrix Recurrence

Combine formulas

$$\begin{aligned}
\mathbf{x}^{[k+1]} &= N\mathbf{w}^{[k]} + \overbrace{RA^T S\mathbf{b} - N\mathbf{c}/\rho}^{\mathbf{h}} \\
\mathbf{w}_{\text{tmp}} &= \mathbf{x}^{[k+1]} - \tfrac{1}{2}(I - D^{[k]})\mathbf{w}^{[k]} \\
D^{[k+1]} &= \textsc{Diag}(\textsc{sign}(\mathbf{w}_{\text{tmp}})) \\
\mathbf{w}^{[k+1]} &= |\mathbf{w}_{\text{tmp}}| = D^{[k+1]}\mathbf{w}_{\text{tmp}}
\end{aligned}$$

with $R = (Q/\rho + I)^{-1}$ is the resolvent of Q, $S = (ARA^T)^{-1}$ is the inverse of the Schur complement, and $N = R - RA^T SAR$.
to get the following iteration

ADMM as a Linear Recurrence

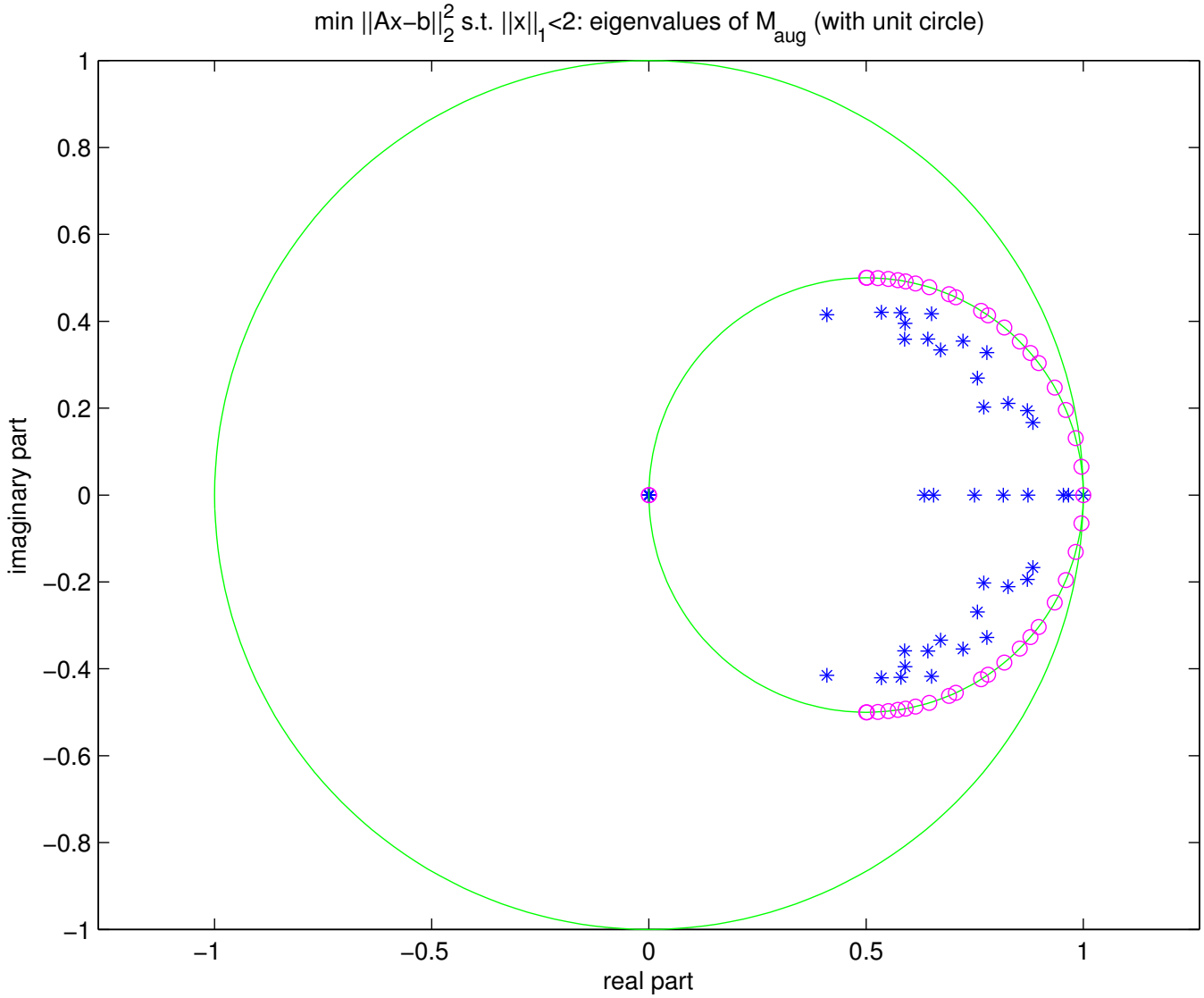
Algorithm 3: One Pass of Reduced ADMM

Start with $\mathbf{w}^{[k]}, D^{[k]}$.

0. $\mathbf{w}_{\text{tmp}} = (N - \frac{1}{2}(I - D^{[k]}))\mathbf{w}^{[k]} + \mathbf{h}$
1. $D^{[k+1]} = \textsc{Diag}(\textsc{sign}(\mathbf{w}_{\text{tmp}}))$
2. $\mathbf{w}^{[k+1]} = D^{[k+1]}\mathbf{w}_{\text{tmp}}$

Result is $\mathbf{w}^{[k+1]}, D^{[k+1]}$ for next pass.

Spectral Properties



- ○ = eigenvalues for LP near end of iteration.
- ∗ = eigenvalues for QP.

Matrix Recurrence.

Step 2 of Algorithm 3 is written as follows:

$$\begin{aligned}\begin{pmatrix}\mathbf{w}^{[k+1]}\\ 1\end{pmatrix} &= \mathbf{M}_{\text{aug}}^{[k]}\begin{pmatrix}\mathbf{w}^{[k]}\\ 1\end{pmatrix} = \begin{pmatrix}M^{[k]} & D^{[k+1]}\mathbf{h}\\ 0 & 1\end{pmatrix}\begin{pmatrix}\mathbf{w}^{[k]}\\ 1\end{pmatrix}\\ &= \begin{pmatrix}D^{[k+1]}(N-\frac{1}{2}(I-D^{[k]})) & D^{[k+1]}\mathbf{h}\\ 0 & 1\end{pmatrix}\begin{pmatrix}\mathbf{w}^{[k]}\\ 1\end{pmatrix},\end{aligned} \tag{6}$$

where $\mathbf{h} = RA^TS\mathbf{b} - N\mathbf{c}/\rho$

Converges to eigenvector: if eigenvector is all non-negative, get solution to original QP/LP. Otherwise, the flag matrix (D) will change to yield a new operator.

Regimes based on spectral properties.

If $D^{[k+1]} = D^{[k]}$:

[a] The spectral radius of $M^{[k]}$ is strictly less than 1. If close enough to the optimal solution (if it exists), the result is linear convergence to that solution.

[b] $M^{[k]}$ has an eigenvalue equal to 1 which results in a 2×2 Jordan block for $\mathbf{M}_{\text{aug}}^{[k]}$. The process tends to a constant step, either diverging, or driving some component negative, resulting in a change in the operator $M^{[k]}$.

[c] $M^{[k]}$ has an eigenvalue equal to 1, but $\mathbf{M}_{\text{aug}}^{[k]}$ still has no non-diagonal Jordan block for eigenvalue 1; If close enough to the optimal solution (if it exists), the result is linear convergence to that solution.

If $D^{[k+1]} \neq D^{[k]}$, then we transition to a new operator:

[d] $M^{[k]}$ has have an eigenvalue of absolute value 1, but not equal to 1. This can occur when the iteration transitions to a new set of active constraints.

Example: A Simple Basis Pursuit Problem

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } A\mathbf{x} = \mathbf{b}, \tag{7}$$

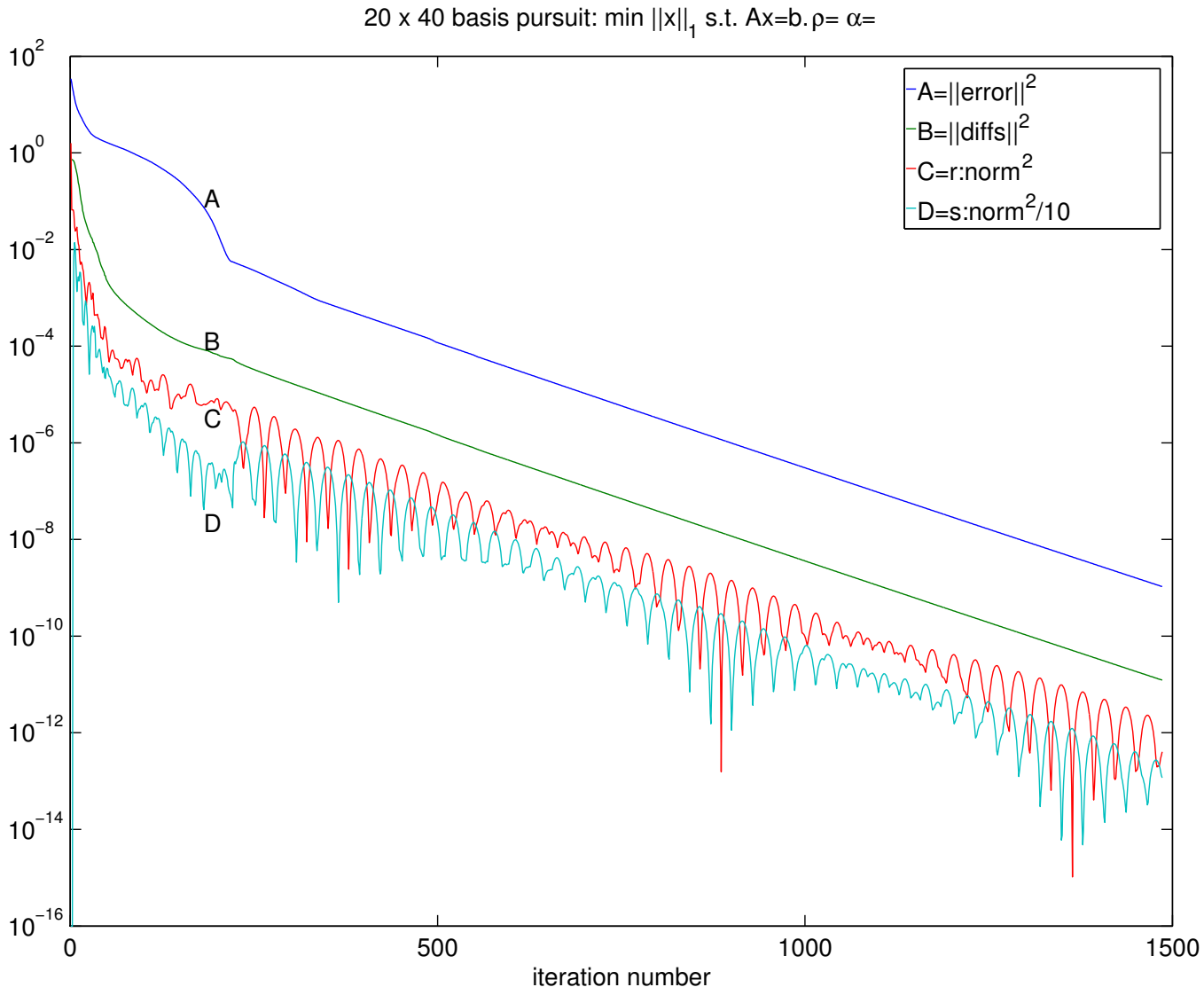
or a soft variation allowing for noise (similar to LASSO)

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2 \text{ subject to } \|\mathbf{x}\|_1 \leq \text{tol}, \tag{8}$$

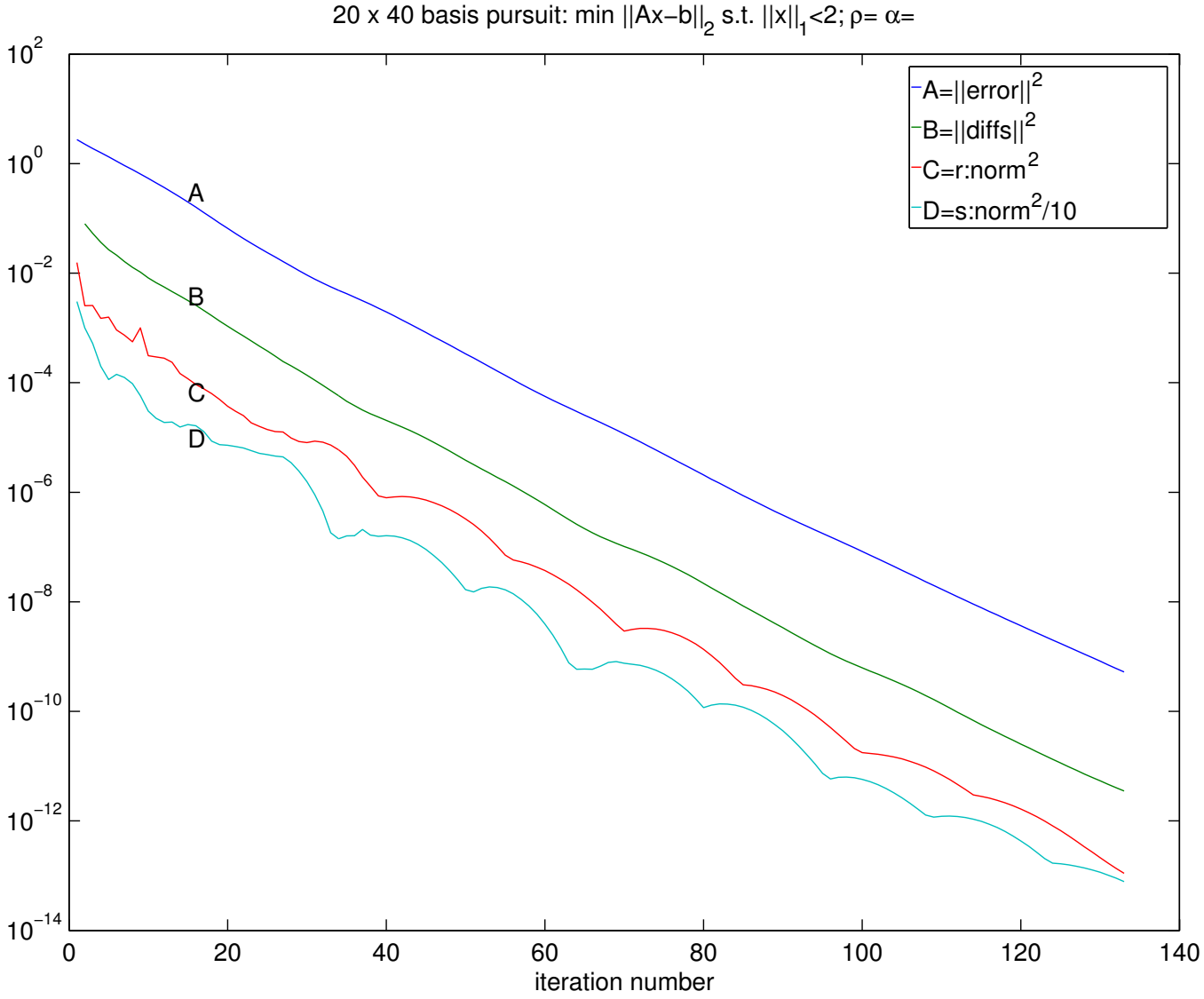
where the elements of $A, \mathbf{b}$ are generated independently by a uniform distribution over $[-1, +1]$. A is 20×40.

Problem (8) is a model to find a sparse best fit, with a trade-off between goodness of fit and sparsity.

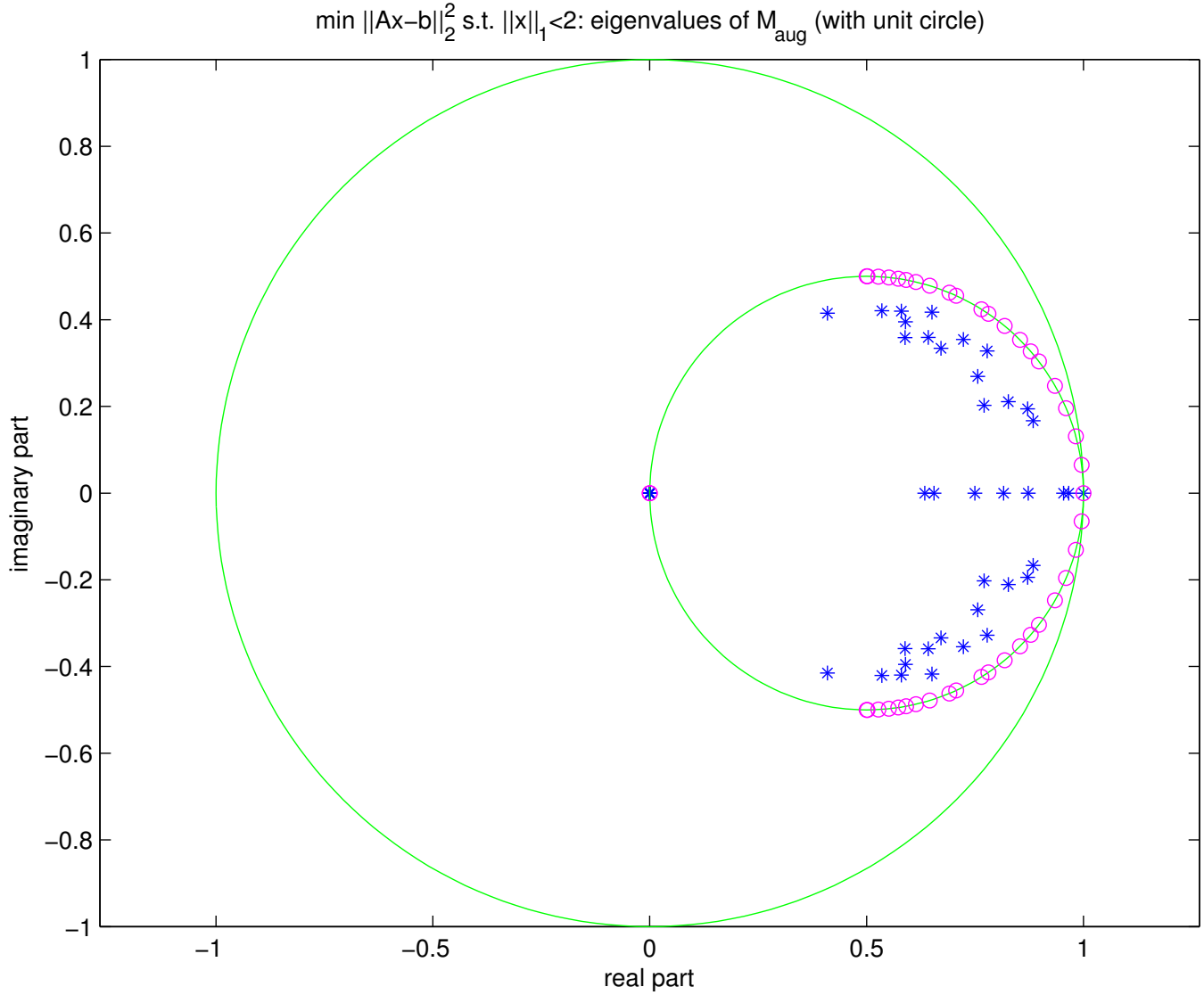
ADMM applied to the Basis Pursuit LP



ADMM applied to the LASSO QP



ADMM Iteration Operator: Spectrum – LASSO



- ○ = eigenvalues for LP in final regime.
- ∗ = eigenvalues for QP.

Toy Example

Simple resource allocation model:

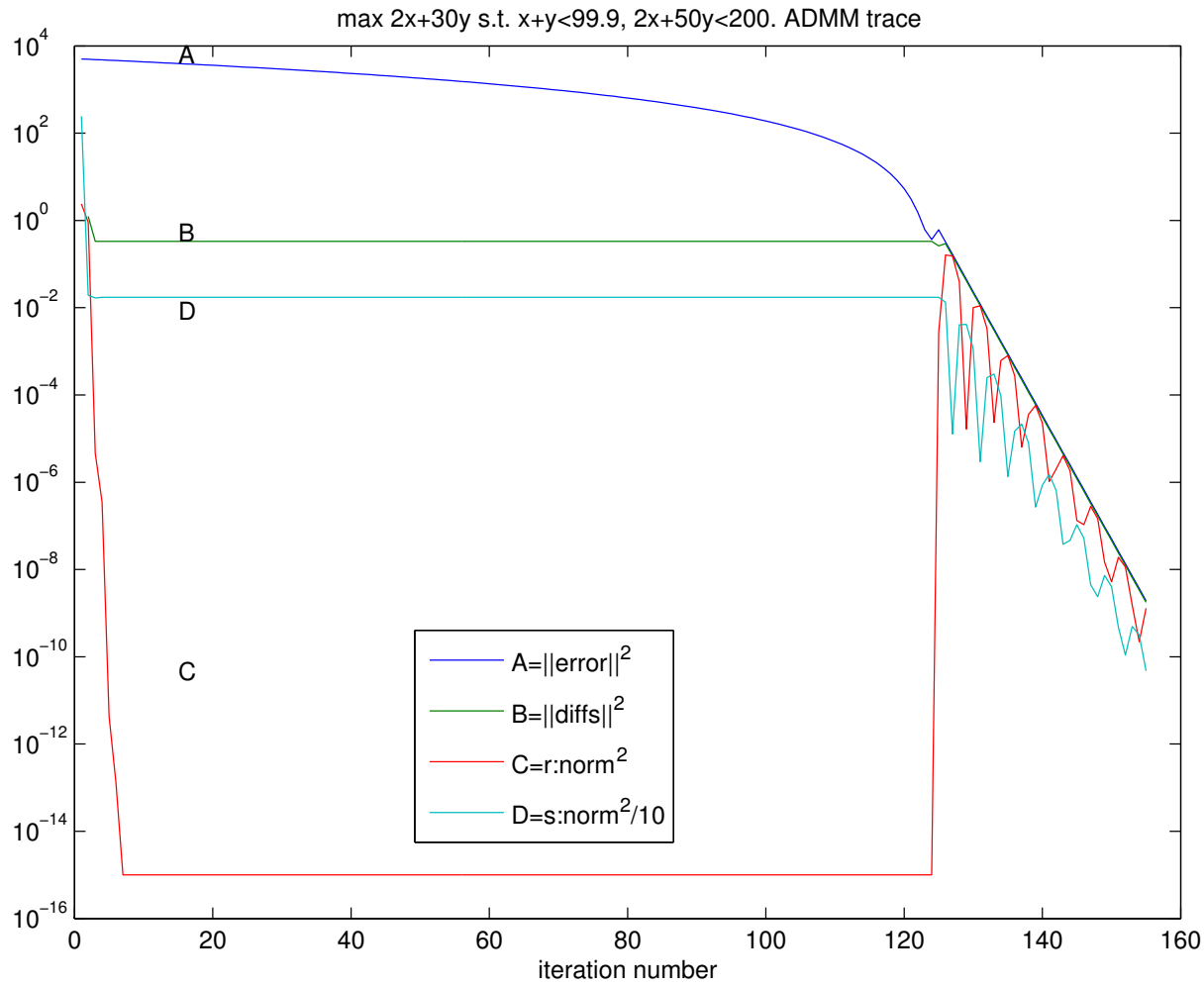
- x_1 = rate of cheap process (e.g. fermentation),
- x_2 = rate of costly process (e.g. respiration).

$$
\begin{array}{lllll}
\text{maximize}_{\mathbf{x}} & +2x_1 + 30x_2 & & & (\textit{desired end product production}) \\
\text{subject to} & x_1 \quad + x_2 & & \leq\ x_{0,max} & (\textit{limit on raw material}) \\
& 2x_1 \quad + 50x_2 & & \leq\ 200 & (\textit{internal capacity limit}) \\
& x_1 \geq 0 & x_2 \geq 0 & & (\textit{irreversibility of reactions})
\end{array}
$$

Put into standard form:

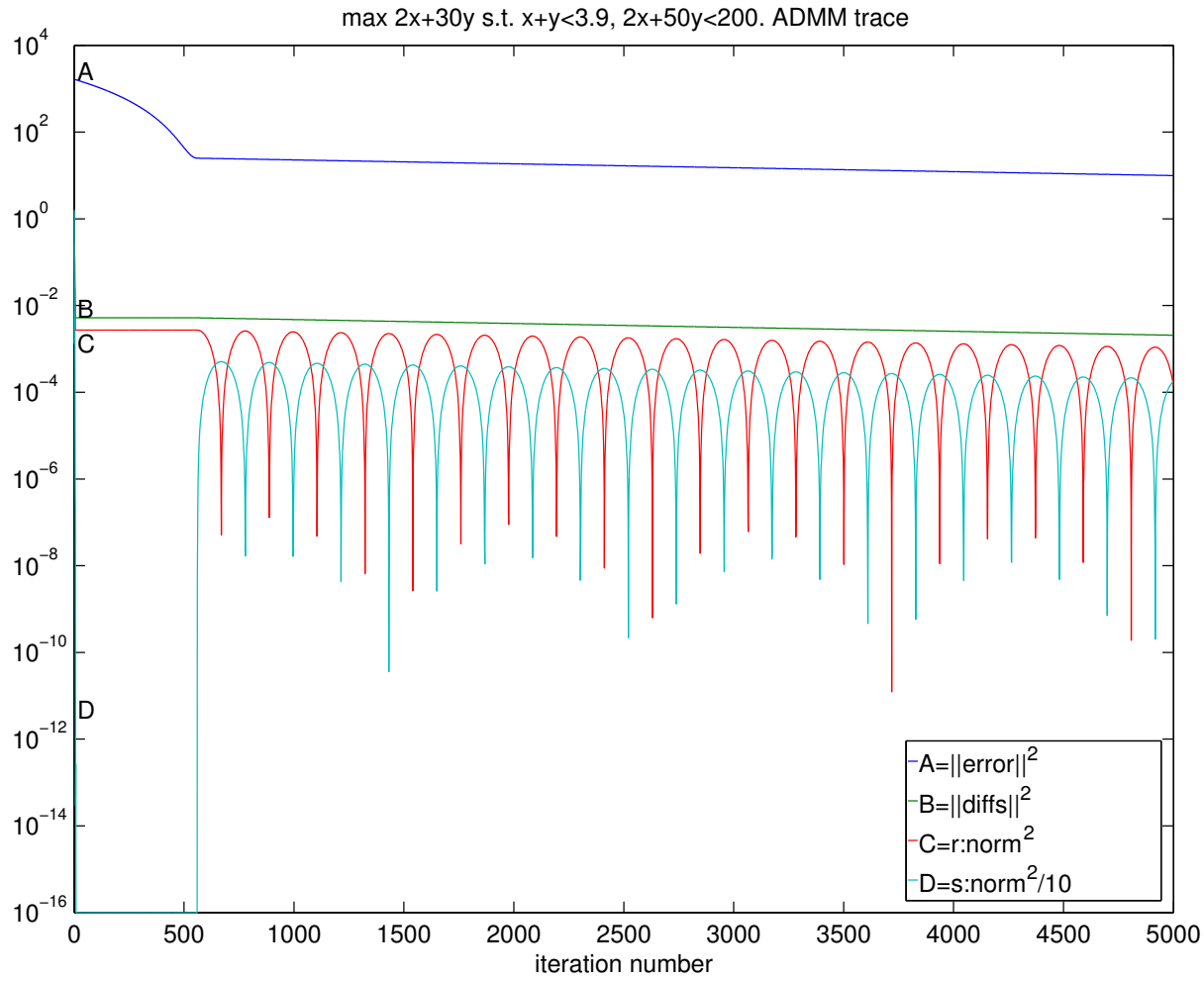
$$
\begin{array}{lllll}
\text{minimize}_{\mathbf{x}} & -2x_1 - 30x_2 & & & (\textit{desired end product production}) \\
\text{subject to} & x_1 \quad + x_2 \quad + x_3 & = & x_{0,max} & (\textit{limit on raw material}) \\
& 2x_1 \quad + 50x_2 \quad + x_4 & = & 200 & (\textit{internal capacity limit}) \\
& x_1 \geq 0 & x_2 \geq 0 & & (\textit{irreversibility of reactions}) \\
& x_3 \geq 0 & x_4 \geq 0 & & (\textit{slack variables})
\end{array}
\tag{9}
$$

Typical Convergence Behavior $v_{0,max} = 99.9$



ADMM on Example 1: typical behavior. Curves: A: error $\|(\mathbf{z}^{[k]}-\mathbf{u}^{[k]})-(\mathbf{z}^{*}-\mathbf{u}^{*})\|^2$. B: $\|(\mathbf{z}^{[k]}-\mathbf{u}^{[k]})-(\mathbf{z}^{[k-1]}-\mathbf{u}^{[k-1]})\|^2$. C: $\|(\mathbf{x}^{[k]}-\mathbf{z}^{[k]})\|^2$. D: $\|(\mathbf{z}^{[k]}-\mathbf{z}^{[k-1]})\|^2/10$ (D is scaled by 1/10 just to separate it from the rest).

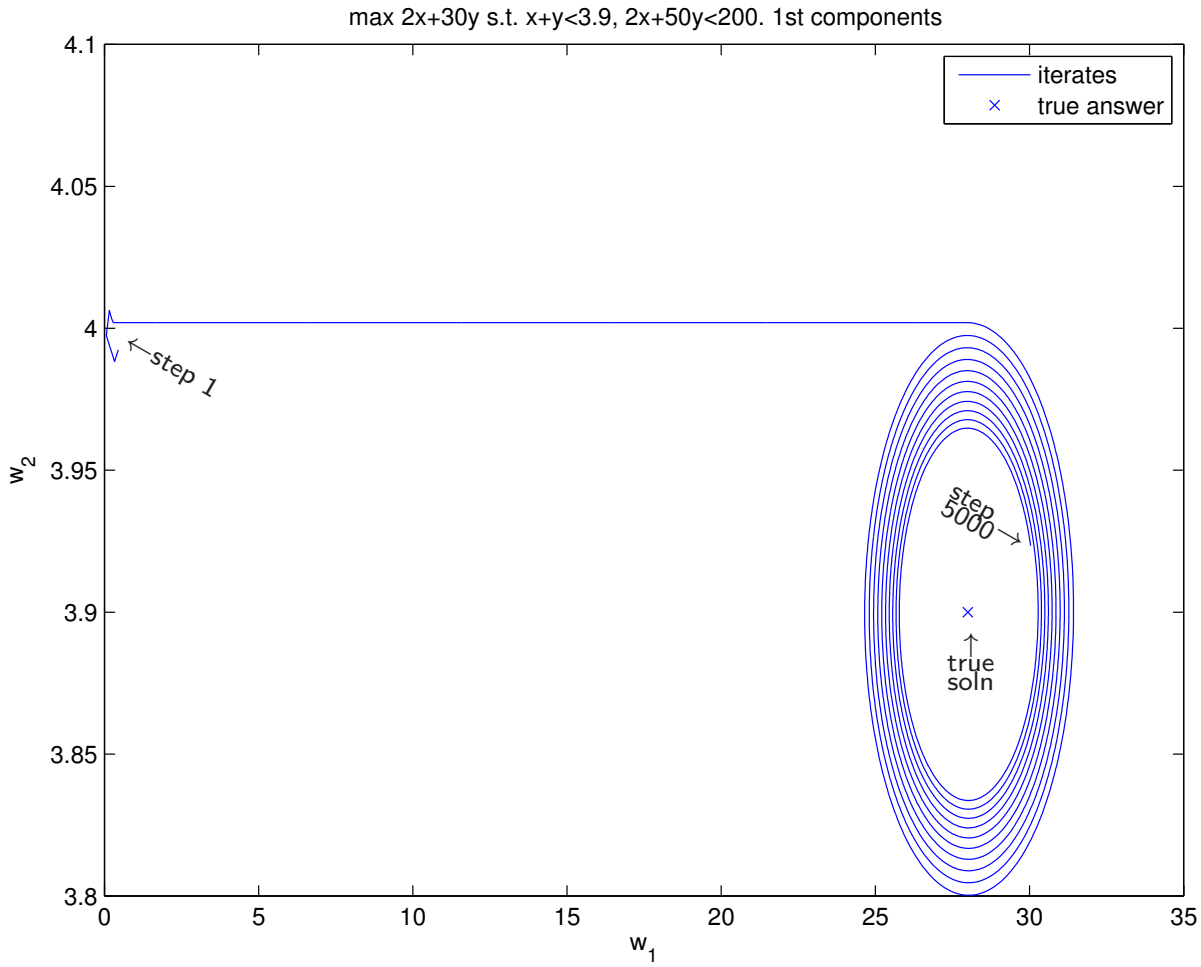
Second Toy Example $v_{0,max} = 3.99$



ADMM on Example 2: slow linear convergence.

Second largest eigenvalue $= \sigma(M) = 0.999896$. convergence is very slow: $-1/\log_{10}(\sigma(M)) = 22135$ iterations needed per decimal digit of accuracy.

Convergence Of Second Toy Example



Convergence behavior of first two components of $\mathbf{w}^{[k]}$ for Example 2, showing the initial straight line behavior (initial regime [b]) leading to the spiral (final regime [a]).

Alternative Iterations for LASSO

- Model Problem $\min_{\mathbf{x}} f(x) + \lambda\|\mathbf{x}\|_1$.
- ISTA: $\mathbf{x}_{k+1}^{\text{ISTA}} = \text{Shr}_{\lambda/c}\left(\mathbf{x}_k - {}^1\!/_c \nabla f(\mathbf{x})\right)$, where c satisfies $cI - f''(\mathbf{x})$ is pos.def.
- FISTA: $\mathbf{x}_{k+1}^{\text{FISTA}} = \mathbf{x}_k^{\text{ISTA}} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_k^{\text{ISTA}} - \mathbf{x}_{k-1})$, where $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$.
- Here $\text{Shr}_\sigma(x) = x - \sigma\text{sign}(x)$ if $|x| > \sigma$, else 0.
- Coordinate descent: like Gauss-Seidel on $\min f(\mathbf{x}) + \lambda \mathbf{e}^T\mathbf{x}$, where $\mathbf{e} = \{0, \pm 1\}_1^n$.

Conclusions

- Many different types of data, many highly unstructured.
- Extracting patterns or connections in data involves somehow reducing the volume of data one must look at.
- Data Reduction is an old paradigm that has been updated for the modern digital age.
- Methods discussed here started with classical PCA - SVD based approaches (e.g., assuming independent gaussian noise).
- Connections and pair-wise correlations modeled by graphs.
- Graphs modeled by random walks, counting subgraphs, min-cut/max-flow, models, ...
- Sparse representations: wide variety of sparse approximations: low fill, short basis, non-negative basis, non-squared loss function, count violations of some constraints, low rank (nuclear norm = $L1$-norm on the singular values), ...
- Leads to need for scalable solvers for very large convex programs.

FUTURE WORK

ADMM $\Longleftrightarrow$ power method with different operators, changing with regime. Replace power method with faster eigensolver.

Conduct similar analysis on other patterns (e.g. LASSO).

Discover relation between eigenvalues controlling convergence rate and original QP/LP.

THANK YOU!

References

Bennett, K. P., & Campbell, C. (2000). Support vector machines: Hype or hallelujah? In *Sigkdd explorations*, vol. 2 #2. ACM.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*, 1–122. http://www.stanford.edu/~boyd/papers/admm/.

Candés, E. J., & Recht, B. (2008). Exact matrix completion via convex optimization. *Found. of Comput. Math.*, *9*, 717–772.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Rev.*, *43*, 129–159.

Davis, G., Mallat, S., & Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive Approximation*, *13*, 57–98. 10.1007/BF02678430.

Donoho, D., & Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. Saul and B. Schölkopf (Eds.), *Advances in neural information processing systems 16*. Cambridge, MA: MIT Press.

Elad, M., Figueiredo, M., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, *98*, 972 –982.

Grant, M., & Boyd, S. (2010). CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx.

Han, Y., Sun, Z., Tan, T., & Hao, Y. (2009). Palmprint recognition based on regional rank correlation of directional features. In M. Tistarelli and M. Nixon (Eds.), *Advances in biometrics*, vol. 5558 of *Lecture Notes in Computer Science*, 587–596. Springer Berlin / Heidelberg.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich and V. Tresp (Eds.), *Advances in neural information processing systems 16*, vol. 13, 556–562. Cambridge, MA: MIT Press.

Li, Y., Zhang, Z.-L., & Boley, D. (2011). the routing continuum from shortest-path to all-path: A unifying theory. *The 31st Int'l Conference on Distributed Computing Systems (ICDCS 2011)* (pp. 847–856). IEEE.

Liu, J., Ji, S., & Ye, J. (2009). Slep: Sparse learning with efficient projections. http://www.public.asu.edu/~jye02/Software/SLEP. Arizona State University.

Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. *Proc. CACSD Conf.*. Taipei, Taiwan. http://users.isy.liu.se/johanl/yalmip .

Mallat, S., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, *41*, 3397 –3415.

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In T. Pajdla and J. Matas (Eds.), *Computer vision - ECCV 2004*, vol. 3022 of *Lecture Notes in Computer Science*, 71–84. Springer Berlin / Heidelberg.

Palaio, H., Maduro, C., Batista, K., & Batista, J. (2009). Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on* (pp. 825 –830).

Pang, Y., Yuan, Y., & Li, X. (2008). Effective feature extraction in high-dimensional space. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *38*, 1652 –1656.

Porikli, F., & Kocak, T. (2006). Robust license plate detection using covariance descriptor in a neural network framework. *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on* (pp. 107 –107).

Shi, X., Fan, W., & Yu, P. S. (2010). Efficient semi-supervised spectral co-clustering with constraints. *ICDM* (pp. 1043–1048).

Sivalingam, R., Boley, D., Morellas, V., & Papanikolopoulos, N. (2010). Tensor sparse coding for region covariances. *European Conf. on Comp. Vision (ECCV 2010)* (pp. 722–735). Springer.

Sivalingam, R., Boley, D., Morellas, V., & Papanikolopoulos, N. (2011). Positive definite dictionary learning for region covariances. *IEEE Int'l Conf. on Comp. Vision (ICCV 2011)* (pp. 1013–1019).

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

Tuzel, O., Porikli, F., & Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. In A. Leonardis, H. Bischof and A. Pinz (Eds.), *Computer vision ECCV 2006*, vol. 3952 of *Lecture Notes in Computer Science*, 589–600. Springer Berlin / Heidelberg.

Tuzel, O., Porikli, F., & Meer, P. (2007). Human detection via classification on riemannian manifolds. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (pp. 1 –8).