Hard-Thresholding Algorithms for Sparsity-Constrained Optimization

Xiao-Tong Yuan

S-mart Lab, NUIST

VALSE'14, Qingdao, April 22, 2014

Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

Newton-type Greedy Pursuit

Summary

Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

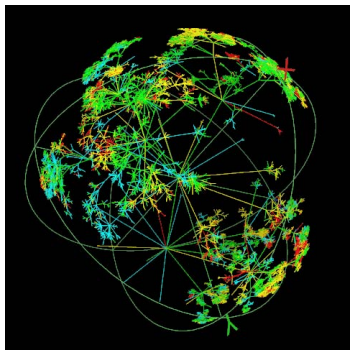
Newton-type Greedy Pursuit

Summary

Background: Massive Data

High-Dimensional: number of features is huge

High-Volume: number of collected samples is huge



High-Complexity: interactions among features and/or samples have complex structure

Challenges in Massive Data Analysis

Modeling challenges:

- Flexible statistical methods

Challenges in Massive Data Analysis

Modeling challenges:

- Flexible statistical methods

Computational challenges:

- Scalable optimization

Example I: Visual Analysis

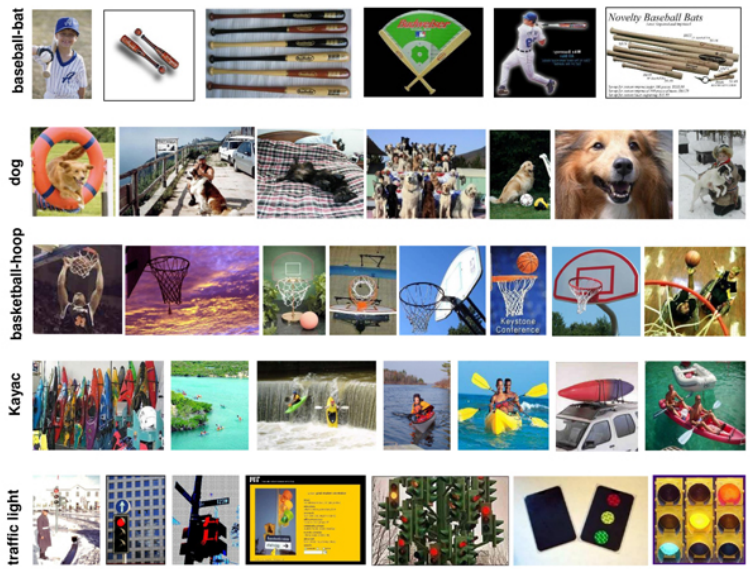
Real scene image data:

- Large-scale: $\sim 10^7$ images in ImageNet

Example I: Visual Analysis

Real scene image data:

- Large-scale: $\sim 10^7$ images in ImageNet
- Image clustering and classification (*TIP*'12, *TPAMI*'13, *TIT*'14)



Example II: Network Analysis

Web graph data:

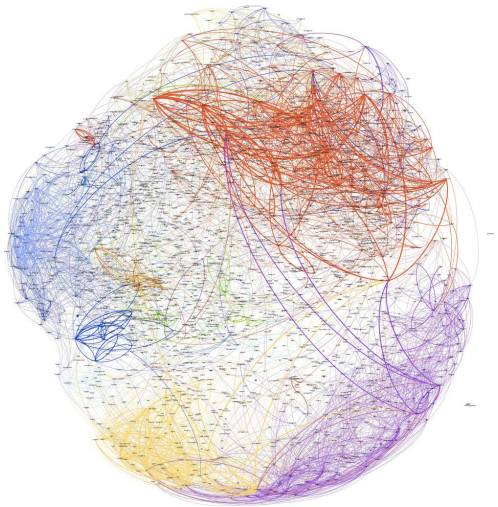
- High-dimensional: $> 10^7$ nodes (web sites) and $> 10^9$ edges (page links)

Example II: Network Analysis

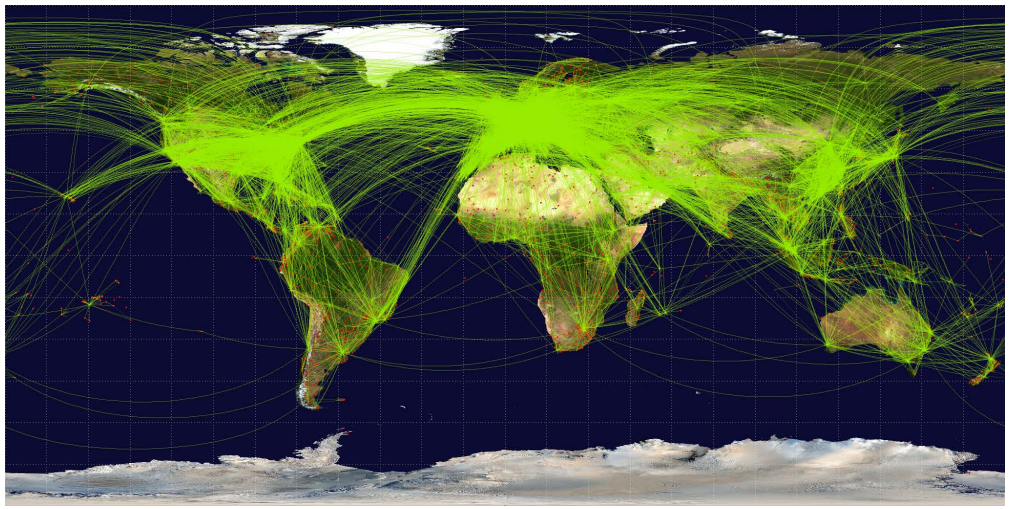
Web graph data:

- High-dimensional: $> 10^7$ nodes (web sites) and $> 10^9$ edges (page links)
- Dense subgraph finding on networks (*JMLR*'13)

Web graph



Commercial airline travel-route network



Example III: Computational Biology

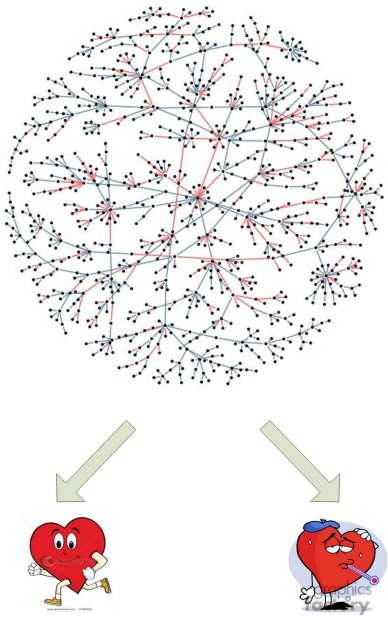
Biology Data:

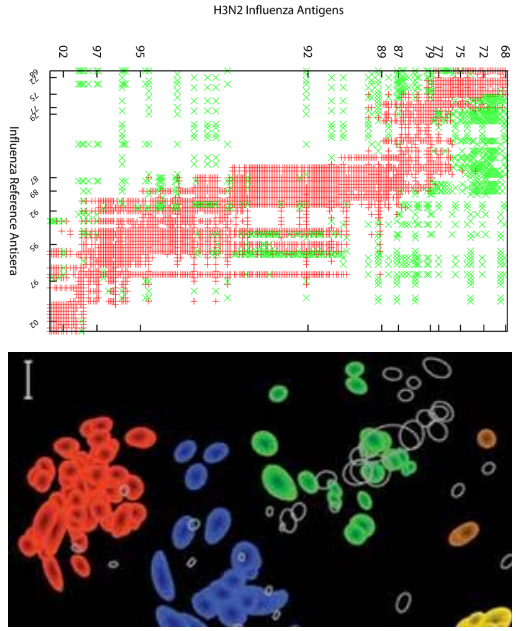
- High-dimensional and complex structures: tens of thousands of genes, missing data and noisy observations

Example III: Computational Biology

Biology Data:

- High-dimensional and complex structures: tens of thousands of genes, missing data and noisy observations
- Tumor classification and influenza serological data integration (*TPAMI*'13, *PLOS ONE*'13)





Sparsity-Constrained Learning

- Sparsity prior: to capture the low-dimensional structure of high-dimensional data

- Formulation: ℓ_0-constrained minimization

$$\min_{x \in \Omega} f(x), \quad \text{s.t. } \|x\|_0 \leq k.$$

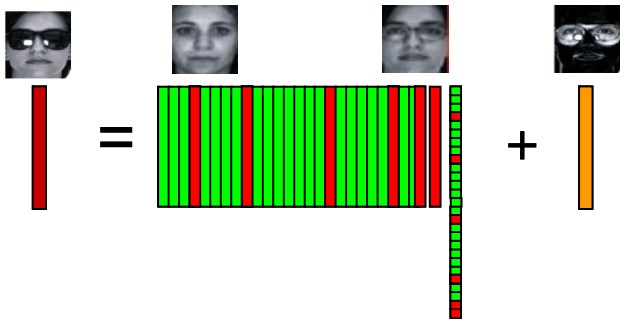
Sparsity-Constrained Learning

- Sparsity prior: to capture the low-dimensional structure of high-dimensional data
- Formulation: ℓ_0-constrained minimization

$$\min_{x \in \Omega} f(x), \quad \text{s.t. } \|x\|_0 \leq k.$$

- Advantages: Better interpretation and improved statistical behavior in high dimensional setup
- Challenges: NP-hard and Non-convex combinatorial problem

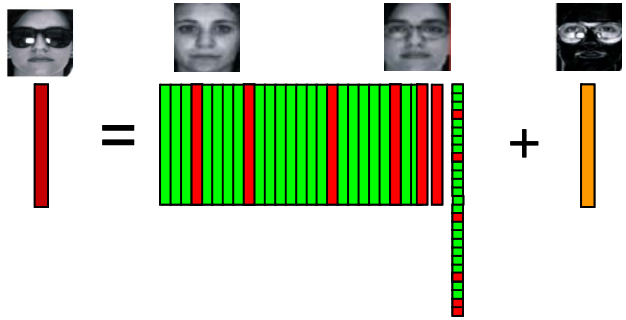
Examples



Compressive sensing:

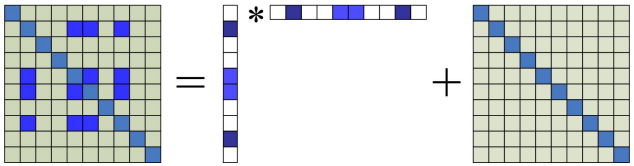
$$\min_x \|y - Ax\|^2, \quad \text{s.t. } \|x\|_0 \leq k.$$

Examples



Compressive sensing:

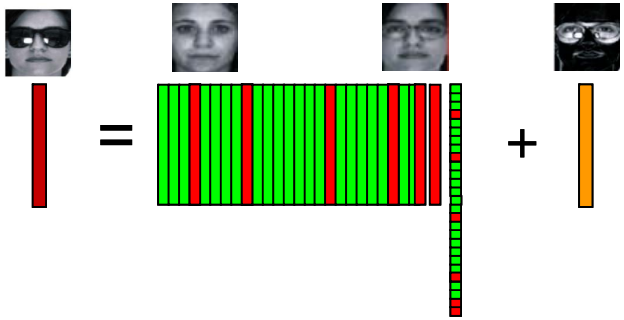
$$\min_x \|y - Ax\|^2, \quad \text{s.t. } \|x\|_0 \leq k.$$



Sparse principal component analysis:

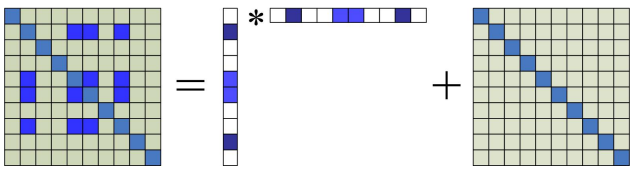
$$\max_x x^\top Ax, \quad \text{s.t. } \|x\| = 1, \ \|x\|_0 \leq k.$$

Examples



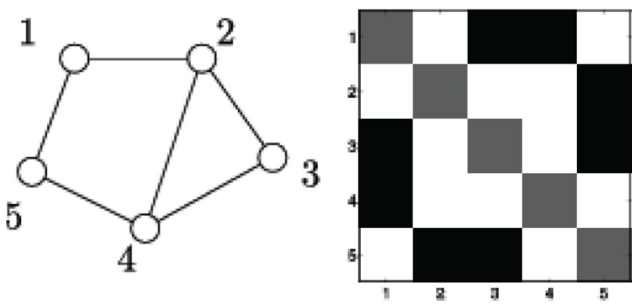
Compressive sensing:

$$\min_x \|y - Ax\|^2, \quad \text{s.t. } \|x\|_0 \leq k.$$



Sparse principal component analysis:

$$\max_x x^\top Ax, \quad \text{s.t. } \|x\| = 1, \ \|x\|_0 \leq k.$$



Gaussian graphical models learning:

$$\min_{\Omega \succ 0} -\log\det(\Omega) + \langle \Sigma_n, \Omega \rangle, \quad \text{s.t. } \|\Omega\|_0 \leq k.$$

Greedy Selection Algorithms

- Compressive sensing literature
  - Orthogonal Matching Pursuit (OMP) (TG07)
  - Compressive Sampling Matching Pursuit (CoSaMP) (NT09)
  - Iterative Hard Thresholding (IHT) (BD09)
  - ...
- For the generic objective
  - Forward Greedy Selection (FGC) (SSZ10)
  - Forward-Backward algorithm (FoBa) (Zhang08)
  - Gradient Support Pursuit Method (GraSP) (BRB13)
  - ...

Greedy Selection Algorithms

- Compressive sensing literature
  - Orthogonal Matching Pursuit (OMP) (TG07)
  - Compressive Sampling Matching Pursuit (CoSaMP) (NT09)
  - Iterative Hard Thresholding (IHT) (BD09)
  - ...
- For the generic objective
  - Forward Greedy Selection (FGC) (SSZ10)
  - Forward-Backward algorithm (FoBa) (Zhang08)
  - Gradient Support Pursuit Method (GraSP) (BRB13)
  - ...

This talk focuses on truncation-type algorithms for

- Sparse eigenvalue problems
- ℓ_0-constrained generic minimization

Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

Newton-type Greedy Pursuit

Summary

Largest k-Sparse Eigenvalue Problem

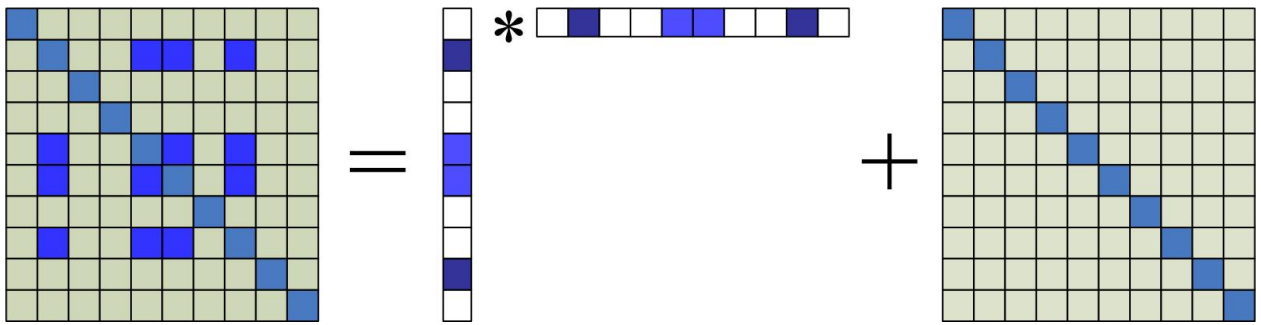
Given a $p \times p$ positive semi-definite matrix A.

$$\max x^\top A x, \quad \text{s.t. } \|x\| = 1, \|x\|_0 \leq k.$$

Largest k-Sparse Eigenvalue Problem

Given a $p \times p$ positive semi-definite matrix A.

$$\max x^\top A x, \quad \text{s.t. } \|x\| = 1, \|x\|_0 \leq k.$$



Motivation

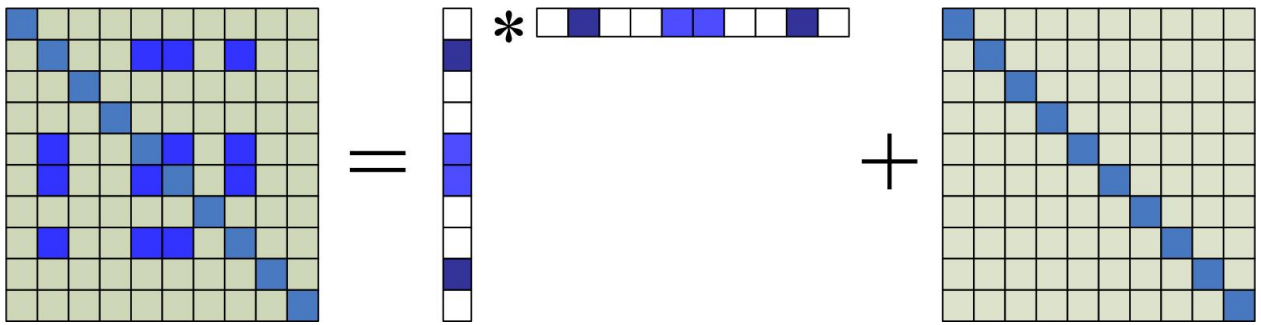
A admits a perturbation formulation $A = \bar{A} + E$

- $\bar{A}$: true matrix with sparse dominant eigenvector $\bar{x}$.
- E: random perturbation due to, e.g., finite sampling.

Largest k-Sparse Eigenvalue Problem

Given a $p \times p$ positive semi-definite matrix A.

$$\max x^\top A x, \quad \text{s.t. } \|x\| = 1, \|x\|_0 \leq k.$$



Motivation

A admits a perturbation formulation $A = \bar{A} + E$

- $\bar{A}$: true matrix with sparse dominant eigenvector $\bar{x}$.
- E: random perturbation due to, e.g., finite sampling.
- Can we approximately estimate $\bar{x}$ from the noisy observation A with large p but small $\bar{k} = \|\bar{x}\|_0$?

Our Contribution

- Algorithm: an efficient power-truncation procedure to estimate the sparse dominant eigenvector.
- Theory: a perturbation theory based error analysis for the proposed algorithm.
- Applications: sparse PCA and densest k-subgraph finding problems.

Yuan & Zhang, *JMLR*, 2013

Truncated Power Method

Power Method

Choose a starting point x_0.

For $t = 1, 2, ...$

$$x_t = Ax_{t-1}/\|Ax_{t-1}\|.$$

Truncated Power Method

Power Method

Choose a starting point x_0.

For $t = 1, 2, ...$

$$x_t = Ax_{t-1}/\|Ax_{t-1}\|.$$

Truncated Power Method

Choose a starting point x_0.

For $t = 1, 2, ...$

(a) Compute $x'_t = Ax_{t-1}/\|Ax_{t-1}\|$;

(b) Let $F_t = \text{supp}(x'_t, k)$ be the indices of x'_t with the largest k absolute values;

(c) Compute $\hat{x}_t = \text{Truncate}(x'_t, F_t)$;

(d) Normalize $x_t = \hat{x}_t/\|\hat{x}_t\|$.

Key idea: maintain k-sparsity at each power iteration.

Sparse Recovery Analysis

Data model:

$$A = \bar{A} + E.$$

Key techniques:

- Perturbation theory of symmetric eigenvalue problem.
- Convergence analysis of untruncated power method.
- Error analysis of hard-thresholding operation.

Sparse Recovery Analysis

Data model:

$$A = \bar{A} + E.$$

Key techniques:

- Perturbation theory of symmetric eigenvalue problem.
- Convergence analysis of untruncated power method.
- Error analysis of hard-thresholding operation.

Main result on the estimation error

Under proper conditions, if we start iteration from an appropriate x_0, then x_t converges geometrically towards $\bar{x}$ until $\|x_t - \bar{x}\| = O(\rho(E, s))$ where $s = 2k + \bar{k}$ and $\rho(E, s) = \max_{\|x\|_0 \leq s} x^\top E x$.

Applications

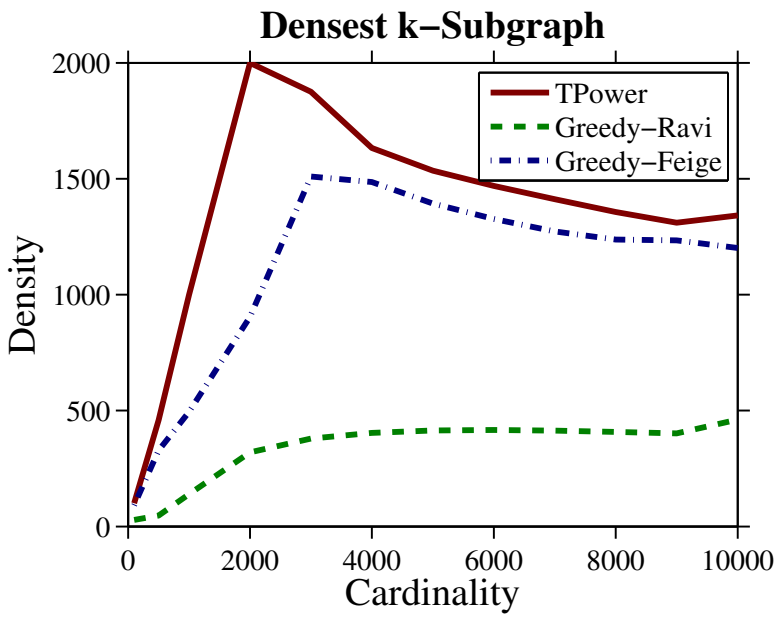
- Sparse PCA: iterative deflation for calculating multiple loading vectors:

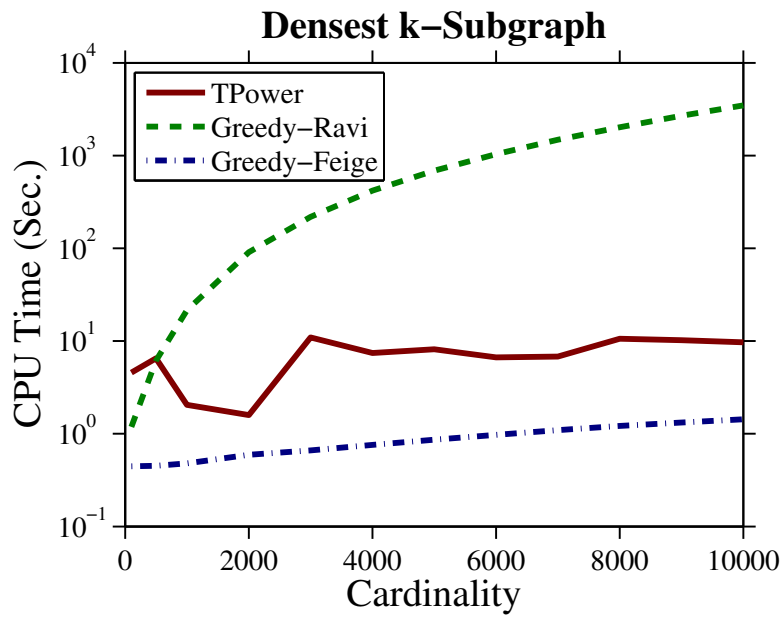
- Densest k-Subgraph Finding: find the a set of k vertices with maximum average degree in the subgraph induced by this set.

$$\max_{\pi \in \mathbb{R}^n} \pi^\top W \pi, \qquad \text{s.t. } \pi \in \{1, 0\}^n, \|\pi\|_0 = k.$$

Results on Web Graph Data

Graph	Nodes ($\|V\|$)	Total Arcs ($\|E\|$)	Average Degree
hollywood-2009	1,139,905	113,891,327	99.91



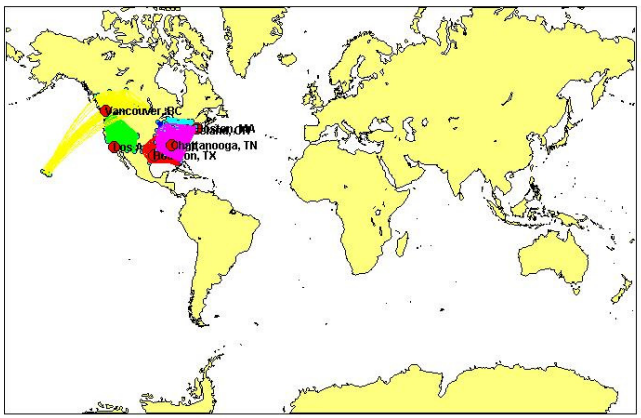


Results on Air-Travel Route Data

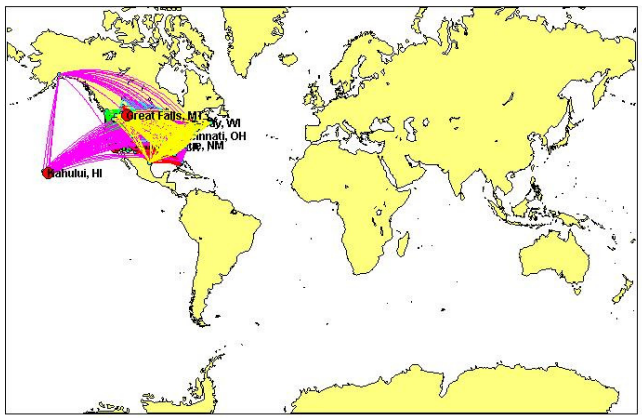
A graph of size $|V| = 456$ and $|E| = 71,959$:

- Vertices: 456 busiest commercial airports in USA and Canada
- Edge weights: inverse of the mean flight time

TPower



Greedy-Feige



Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

Newton-type Greedy Pursuit

Summary

Sparsity-Constrained Minimization

$$\min_{x \in \mathbb{R}^p} f(x), \quad \text{s.t. } \|x\|_0 \leq k,$$

where f is a convex loss function, e.g.,

- logistic loss for logistic regression
- hinge loss for support vector machines
- exponential loss for boosting classification

Gradient Hard Thresholding Pursuit (GraHTP)

Set $x^{(0)} = 0$.

For $t = 1, 2, ...,$

(a) Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta \nabla f(x^{(t-1)})$;

(b) Compute $x^{(t)} = \tilde{x}_k^{(t)}$ as the truncation of $\tilde{x}^{(t)}$ with top k (in magnitude) entries preserved;

(c) Debiasing: $x^{(t)} = \arg\min\{f(x), \mathrm{supp}(x) \subseteq \mathrm{supp}(x^{(t)})\}$ (optional);

Gradient Hard Thresholding Pursuit (GraHTP)

Set $x^{(0)} = 0$.
For $t = 1, 2, ...,$

(a) Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta \nabla f(x^{(t-1)})$;

(b) Compute $x^{(t)} = \tilde{x}_k^{(t)}$ as the truncation of $\tilde{x}^{(t)}$ with top k (in magnitude) entries preserved;

(c) Debiasing: $x^{(t)} = \arg\min\{f(x), \text{supp}(x) \subseteq \text{supp}(x^{(t)})\}$ (optional);

- (a): Traditional gradient descent, η is the step-size

Gradient Hard Thresholding Pursuit (GraHTP)

Set $x^{(0)} = 0$.
For $t = 1, 2, ...,$

(a) Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta \nabla f(x^{(t-1)})$;

(b) Compute $x^{(t)} = \tilde{x}_k^{(t)}$ as the truncation of $\tilde{x}^{(t)}$ with top k (in magnitude) entries preserved;

(c) Debiasing: $x^{(t)} = \arg\min\{f(x), \text{supp}(x) \subseteq \text{supp}(x^{(t)})\}$ (optional);

- (a): Traditional gradient descent, η is the step-size
- (b): Truncation operation to keep the iterate k-sparse

Gradient Hard Thresholding Pursuit (GraHTP)

Set $x^{(0)} = 0$.
For $t = 1, 2, ...,$

(a) Compute $\tilde{x}^{(t)} = x^{(t-1)} - \eta \nabla f(x^{(t-1)})$;

(b) Compute $x^{(t)} = \tilde{x}_k^{(t)}$ as the truncation of $\tilde{x}^{(t)}$ with top k (in magnitude) entries preserved;

(c) Debiasing: $x^{(t)} = \arg\min\{f(x), \text{supp}(x) \subseteq \text{supp}(x^{(t)})\}$ (optional);

- (a): Traditional gradient descent, η is the step-size
- (b): Truncation operation to keep the iterate k-sparse
- (c): An optional debiasing step

Yuan *et al.*, *ICML*, 2014

Convergence

Under proper conditions,

- the sequence $\{x^{(t)}\}$ defined by GraHTP converges in a finite number of iterations.
- the sequence $\{f(x^{(t)})\}$ defined by GraHTP without debiasing step (or Fast GraHTP) converges.

Sparse Recovery

Key techniques:

- Convergence analysis of unconstrained gradient descent procedure.
- Error analysis of truncation operation.

Sparse Recovery

Key techniques:

- Convergence analysis of unconstrained gradient descent procedure.
- Error analysis of truncation operation.

Main results

Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector and $k \geq \bar{k}$. Under proper conditions,

- the sequence $\{x^{(t)}\}$ defined by GraHTP converges geometrically towards $\bar{x}$ until $\|x^{(t)} - \bar{x}\| = O(\|\nabla_k f(\bar{x})\|)$
- the sequence $\{x^{(t)}\}$ defined by FGraHTP converges geometrically towards $\bar{x}$ until $\|x^{(t)} - \bar{x}\| = O(\|\nabla_s f(\bar{x})\|)$ where $s = 2k + \bar{k}$.

Remarks

- In the ideal case where $\nabla f(\bar{x}) = 0$, under proper conditions, GraHTP is able to recover $\bar{x}$ in finite iterations.
- In the setup of CS, GraHTP reduces to HTP (Foucart12) which requires weaker RIP condition than prior CS algorithms.
- Although have similar theoretical guarantees, GraHTP is cheaper than GraSP (BRB13) in iteration.

Applications

Sparsity-constrained M-estimation:

$$\min_{w} f(w) = \frac{1}{n} \sum_{i=1}^{n} \phi(x^{(i)} \mid w), \quad \text{subject to } \|w\|_0 \leq k.$$

- Sparsity-constrained logistic regression
- Sparsity-constrained support vector machines
- Sparsity-constrained Gaussian graphical models learning

Results on News20 Data

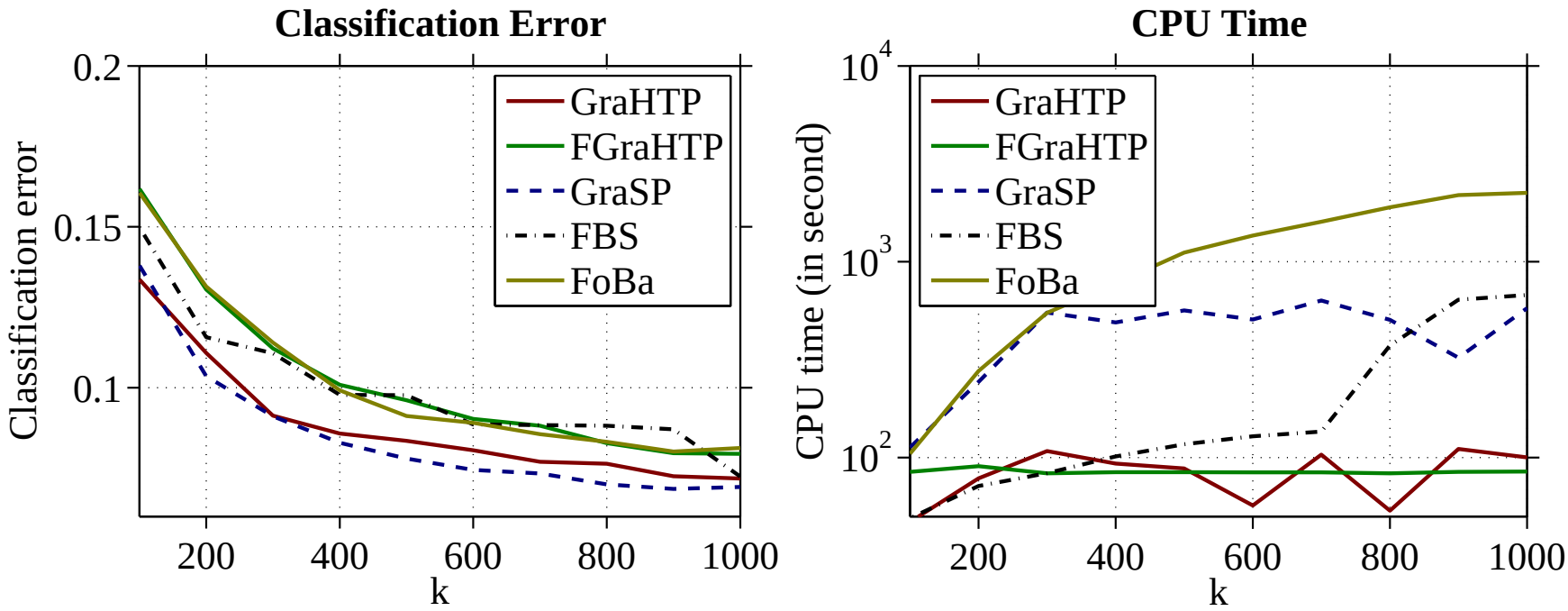


Figure: Logistic regression: Classification error and CPU running time curves of the considered methods.

Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

Newton-type Greedy Pursuit

Summary

Motivation

- First-order methods: iterate fast but converge slow.
- Newton-type methods gain significant interests in ℓ_1-regularized/-constrained convex minimization.
- Natural question: can we adopt Newton-type methods for ℓ_0-constrained minimization?

Constrained Newton-type Methods

To minimizes a smooth convex objective f over a convex set Ω, i.e.,

$$\min_x f(x), \qquad \text{s.t. } x \in \Omega,$$

constrained Newton-type methods

- form a quadratic approximation to the function around $x^{(t)}$:

$$Q_f(y; x^{(t)}) := f(x^{(t)}) + \nabla f(x^{(t)})^\top (y - x^{(t)}) + \frac{1}{2}(y - x^{(t)})^\top H^{(t)} (y - x^{(t)}),$$

- minimizes the quadratic model over the original feasible set Ω:

$$\tilde{x}^{(t)} = \arg\min_{y \in \Omega} Q_f(y; x^{(t)})$$

- perform line search:

$$x^{(t+1)} = x^{(t)} + \beta(\tilde{x}^{(t)} - x^{(t)}),$$

- The method has a super-linear rate of convergence at a local minimizer (Bertsekas99).

Proximal Newton-type Methods

Constrained Newton-type methods can be extended to proximal Newton-type methods for composite optimization:

$$\min_x f(x) + h(x), \qquad \text{s.t. } x \in \Omega,$$

where f is smooth and h is non-smooth.

- construct a scaled proximal mapping around $x^{(t)}$:

$$\tilde{x}^{(t)} = \arg\min_{y \in \Omega} \left\{ Q_f(y; x^{(t)}) + h(y) \right\}$$

- perform line search:

$$x^{(t+1)} = x^{(t)} + \beta(\tilde{x}^{(t)} - x^{(t)}),$$

- The method has a super-linear rate of convergence at a local minimizer (LSA14).

Newton-type Greedy Pursuit

Idea: to adapt the constrained Newton-type methods to ℓ_0-constrained minimization.

Newton-type Greedy Pursuit

Idea: to adapt the constrained Newton-type methods to ℓ_0-constrained minimization.

Initialization: set $x^{(0)} = 0$.
For $t = 1, 2, ...,$
Find any $x^{(t)}$ with $\|x^{(t)}\|_0 \leq k$ such that for all $\bar{y}$ with $\|\bar{y}\|_0 \leq k$,

$$Q_f(x^{(t)}; x^{(t-1)}) \leq Q_f(\bar{y}; x^{(t-1)}) + \epsilon,$$

where $\epsilon \geq 0$ controls the solution precision.

Yuan & Liu, *CVPR*, 2014

Choosing $H^{(t)}$

- When Hessian is readily available: set $H^{(t)} = \nabla^2 f(x^{(t)})$.
- When exact estimation of Hessian is expensive: use quasi-Newton strategy to build $H^{(t)} \approx \nabla^2 f(x^{(t)})$.
- Most strategies for choosing Hessian approximations in Newton-type methods can be adapted to choosing $H^{(t)}$.

Constrained Quadratic Model

We use IHT to solve the ℓ_0-constrained quadratic model.

Iterative Hard-Thresholding

Initialization: set $y^{(0)} = x^{(t-1)}$.
For $t = 1, 2, ...,$
(S1) Compute gradient descent:

$$\tilde{y}^{(\tau)} = y^{(\tau-1)} - \eta \nabla Q_f(y^{(\tau-1)}; x^{(t-1)}).$$

(S2) Identify support: $T^{(\tau)} = \text{supp}(\tilde{y}^{(\tau)}, k)$;
(S3) Minimizer over support:

$$y^{(\tau)} = \underset{\text{supp}(y) \subseteq T^{(\tau)}}{\arg\min} \; Q_f(y; x^{(t-1)}).$$

Computational Cost

- Step S1: $O(kp)$ for sparse matrix-vector product.
- Step S3: solve a linear system of size k.
- Observed to be as efficient as first-order greedy methods in practice

Sparse Recovery

Key techniques:

- Convergence analysis of unconstrained Newton-type procedure.
- Error analysis of truncation operation.

Sparse Recovery

Key techniques:

- Convergence analysis of unconstrained Newton-type procedure.
- Error analysis of truncation operation.

Main result

Let $\bar{x}$ be an arbitrary $\bar{k}$-sparse vector and $k \geq \bar{k}$. Under proper conditions, the sequence $\{x^{(t)}\}$ defined by Newton-type greedy pursuit converges superlinearly towards $\bar{x}$ until $\|x^{(t)} - \bar{x}\| = O(\|\nabla_s f(\bar{x})\|)$ where $s = 2k + \bar{k}$.

Simulation

$p = 2000$, $\bar{k} = 200$,

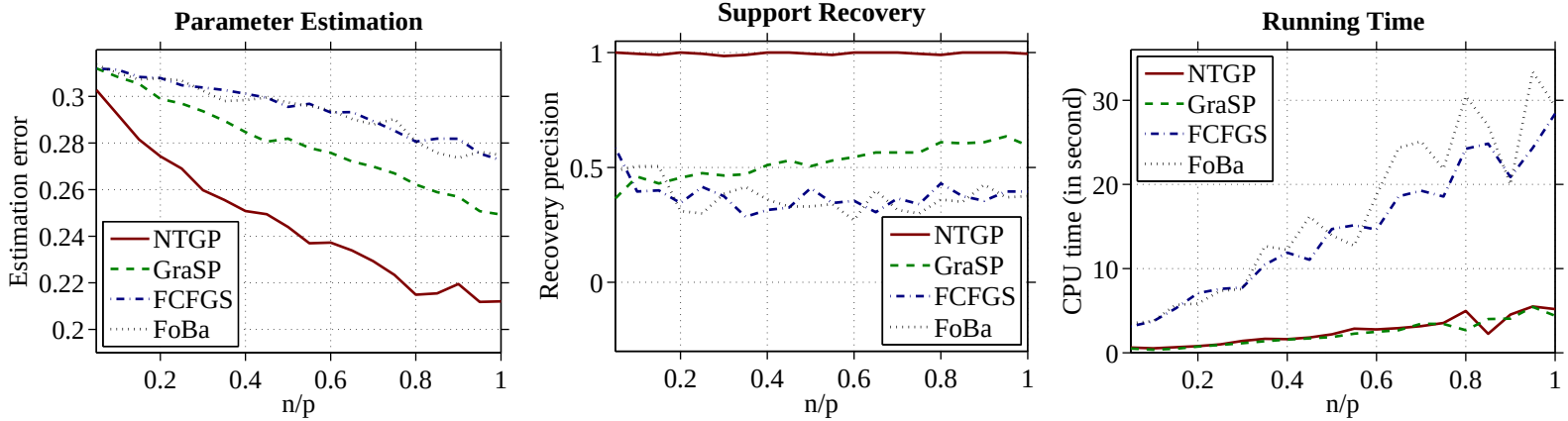


Figure: Logistic regression: parameter estimation error, support recovery precision and running time.

Results on RCV1 Data

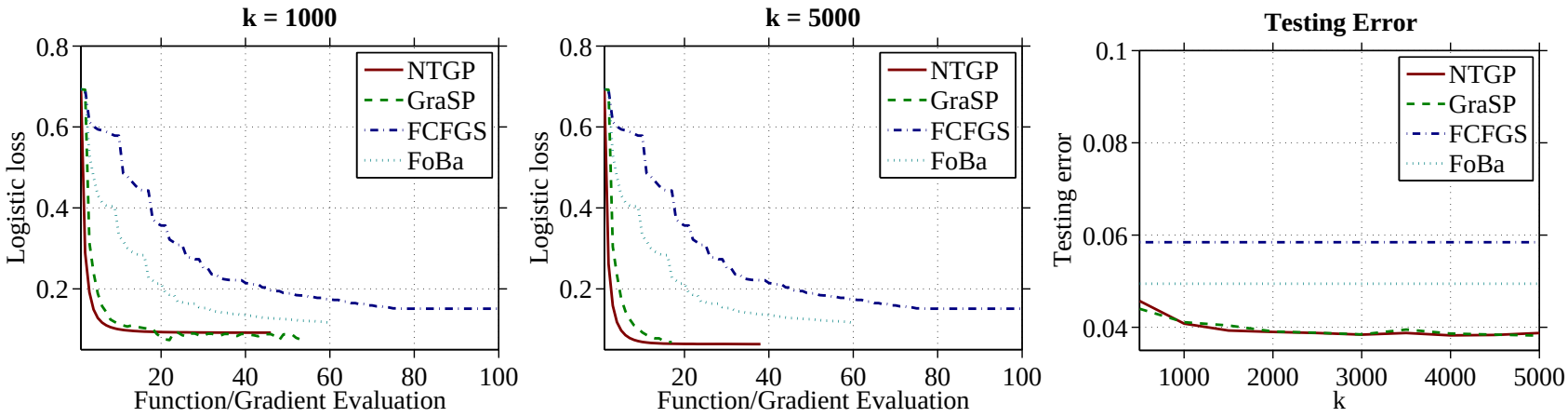


Figure: Logistic regression: objective value convergence curves and testing error curves.

Outline

Sparsity Models

Truncated Power Method

Gradient Hard-Thresholding Pursuit

Newton-type Greedy Pursuit

Summary

Take-Home Messages

Truncation-type methods work favorably for sparsity models:

- TPower for sparse eigenvalue problems: power method + hard-truncation.
- GraHTP for ℓ_0-constrained minimization: gradient descent method + hard-truncation.
- NTGP for ℓ_0-constrained minimization: Newton-type method + hard-truncation.

References

- Xiao-Tong Yuan, Tong Zhang, “Truncated Power Method for Sparse Eigenvalue Problems”, *JMLR*, 2013.
- Xiao-Tong Yuan, Qingshan Liu, “Newton Greedy Pursuit: a Quadratic Approximation Method for Sparsity-Constrained Optimization”, *CVPR*, 2014, accepted.
- Xiao-Tong Yuan, Ping Li, Tong Zhang, “Gradient Hard Thresholding Pursuit for Sparsity-Constrained Optimization”, *ICML*, 2014, accepted.

Thank You!

Questions?