Hybrid Deterministic-Stochastic Methods for Data Fitting

Michael Friedlander[1] Mark Schmidt[2]

[1]University of British Columbia

[2]INRIA/ENS

July 2011

Outline

Algorithm S vs. Algorithm D: Error vs. Iteration

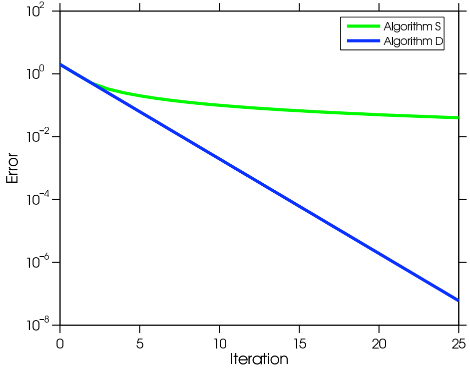
- Should we use Algorithm S or Algorithm D?

Algorithm S vs. Algorithm D: Error vs. Iteration

- Should we use Algorithm S or Algorithm D?
  - On iteration k, Algorithm S has an error of $1/k$.
  - On iteration k, Algorithm D has an error of $1/2^k$.

Algorithm S vs. Algorithm D: Error vs. Iteration

- Should we use Algorithm S or Algorithm D?
  - On iteration k, Algorithm S has an error of $1/k$.
  - On iteration k, Algorithm D has an error of $1/2^k$.



Algorithm S vs. Algorithm D: Error vs. Time

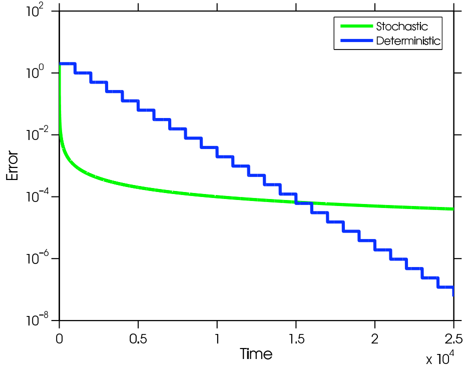
- But, the error is not the whole story:

Algorithm S vs. Algorithm D: Error vs. Time

- But, the error is not the whole story:
  - Iterations of Algorithm S are **cheap**.
  - Iterations of Algorithm D are **expensive**.

Algorithm S vs. Algorithm D: Error vs. Time

- But, the error is not the whole story:
  - Iterations of Algorithm S are **cheap**.
  - Iterations of Algorithm D are **expensive**.



Motivation for Hybrid Methods

Stochastic vs. Deterministic:

- Stochastic makes great progress initially, but slows down.
- Determinstic makes steady progress, but is expensive.

Motivation for Hybrid Methods

Stochastic vs. Deterministic:

- Stochastic makes great progress initially, but slows down.
- Determinstic makes steady progress, but is expensive.

Can a hybrid method get the best of both worlds?

Simple Hybrid Method

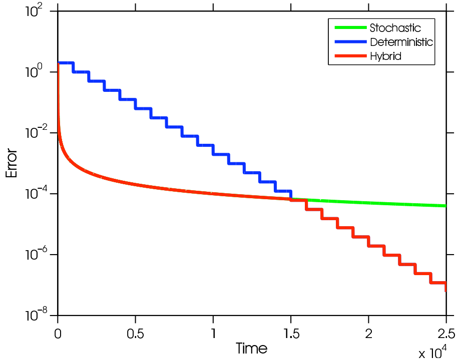
- Simple hybrid method [Cauchy, 1847]:

Simple Hybrid Method

- Simple hybrid method [Cauchy, 1847]:
  - Start out running the low cost method.
  - At some point switch to the low error method.

Simple Hybrid Method

- Simple hybrid method [Cauchy, 1847]:
  - Start out running the low cost method.
  - At some point switch to the low error method.



Better Hybrid Methods?

The question underlying our work:

- Can a hybrid method do **better** than both?

Better Hybrid Methods?

The question underlying our work:

- Can a hybrid method do **better** than both?

The basic idea:

- Start out running the low cost method.
- Gradually switch to the low error method,

Better Hybrid Methods?

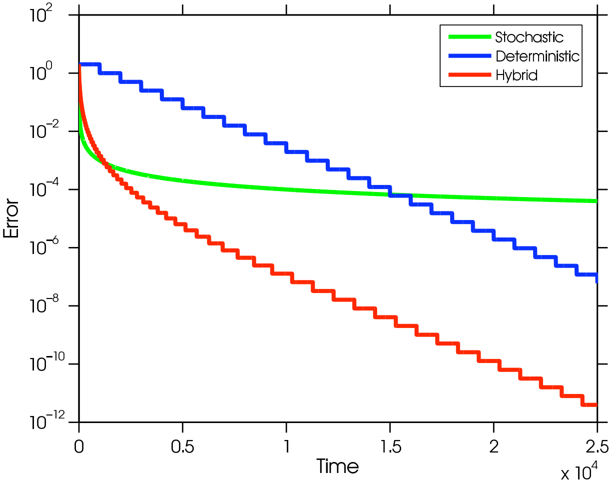
The question underlying our work:

- Can a hybrid method do **better** than both?

The basic idea:

- Start out running the low cost method.
- Gradually switch to the low error method,
  keeping the global convergence rate.

Better Hybrid Methods?



Outline

1. Deterministic vs. Stochastic Optimization
2. Convergence Rates of Gradient Methods
   - Deterministic and Stochastic Convergence Rates
   - Hybrid Algorithm Convergence Rates
3. Practical Issues and Application
4. Other Projects and Summary

Problem Formulation

- We want to minimize a once-differentiable function $f(x)$,

$$\min_{x \in \mathbb{R}^p} f(x).$$

- We assume that $f(x)$ is strongly convex.
- We assume that $\nabla f(x)$ is Lipschitz-continuous.
- For twice-differentiable functions, these are equivalent to

$$\mu I \preceq \nabla^2 f(x) \preceq LI,$$

for some $\mu > 0$ and some $L \geq \mu$.

Deterministic Algorithm Convergence Rate

- Consider the deterministic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

Deterministic Algorithm Convergence Rate

- Consider the deterministic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

- This algorithm has a **strong linear** convergence rate,

$$f(x_k) - f(x_*) \leq (1 - \mu/L)^k [f(x_0) - f(x_*)].$$

Deterministic Algorithm Convergence Rate

- Consider the deterministic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

- This algorithm has a **strong linear** convergence rate,

$$f(x_k) - f(x_*) \leq (1 - \mu/L)^k [f(x_0) - f(x_*)].$$

- But, it uses the *exact gradient* on each iteration.

Stochastic Algorithm Convergence Rate

- Now consider the stochastic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k).$$

Here, $g(x_k)$ is an *approximate gradient*,

$$g(x_k) = \nabla f(x_k) + e_k.$$

Stochastic Algorithm Convergence Rate

- Now consider the stochastic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k).$$

Here, $g(x_k)$ is an *approximate gradient*,

$$g(x_k) = \nabla f(x_k) + e_k.$$

- The (random) error e_k must be zero-mean, finite-variance.
- This might be **much** cheaper to compute.

Stochastic Algorithm Convergence Rate

- Now consider the stochastic gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k).$$

Here, $g(x_k)$ is an *approximate gradient*,

$$g(x_k) = \nabla f(x_k) + e_k.$$

- The (random) error e_k must be zero-mean, finite-variance.
- This might be **much** cheaper to compute.
- But, it leads to a **weak sublinear** convergence rate,

$$\mathbb{E}[f(x_k) - f(x_*)] = O(1/k).$$

Hybrid Algorithm: Bounded Error

- The hybrid gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k), \quad \text{where } g(x_k) = \nabla f(x_k) + e_k.$$

Hybrid Algorithm: Bounded Error

- The hybrid gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k), \quad \text{where } g(x_k) = \nabla f(x_k) + e_k.$$

- We do not assume the error e_k is zero-mean (or random).
- Instead we assume we can bound the error size,

$$||e_k||^2 \leq B^k.$$

Hybrid Algorithm: Bounded Error

- The hybrid gradient descent algorithm:

$$x_{k+1} = x_k - \alpha_k g(x_k), \quad \text{where } g(x_k) = \nabla f(x_k) + e_k.$$

- We do not assume the error e_k is zero-mean (or random).
- Instead we assume we can bound the error size,

$$||e_k||^2 \leq B^k.$$

- Can we achieve a **strong linear** convergence rate?
  (without requiring $B^k = 0$?)

Hybrid Algorithm Strong Linear Convergence Rate

We get the **strong linear** *convergence rate,*

$$f(x_k) - f(x_*) \leq (1-\rho)^k [f(x_0) - f(x_*)].$$

if the errors satisfy

$$||e_k||^2 \leq 2L(\mu/L - \rho)[f(x^k) - f(x^*)],$$

Hybrid Algorithm Strong Linear Convergence Rate

We get the **strong linear** *convergence rate,*

$$f(x_k) - f(x_*) \leq (1-\rho)^k [f(x_0) - f(x_*)].$$

if the errors satisfy

$$||e_k||^2 \leq 2L(\mu/L - \rho)[f(x^k) - f(x^*)],$$

- Error can be large if you are far from the solution.
- Classic deterministic rate is the special case that $\rho = \mu/L$.
- For $\rho < \mu/L$, this **never requires the exact gradient**.

Hybrid Algorithm Weak Linear Convergence Rate

- What if we don't know μ, L, $f(x^*)$, or $f(x^k)$?

Hybrid Algorithm Weak Linear Convergence Rate

- What if we don't know μ, L, $f(x^*)$, or $f(x^k)$?

If the errors satisfy

$$||e_k||^2 \leq O(\gamma^k),$$

then the algorithm has a **weak linear** *convergence rate,*

$$f(x_k) - f(x_*) = O(\sigma^k),$$

for all $\sigma > \max\{1 - \mu/L, \gamma\}$.

Hybrid Algorithm Expected Weak Linear Rate

- What if we can only bound e_k in expectation?

Hybrid Algorithm Expected Weak Linear Rate

- What if we can only bound e_k in expectation?

If the errors satisfy

$$\mathbb{E}[||e_k||^2] \leq O(\gamma^k),$$

then the algorithm has an **expected weak linear** *convergence rate,*

$$\mathbb{E}[f(x_k) - f(x_*)] = O(\sigma^k),$$

for all $\sigma > \max\{\gamma, 1 - \mu/L\}$.

Hybrid Algorithm Expected Weak Sublinear Rate

- What if we can't geometrically decrease the error?

Hybrid Algorithm Expected Weak Sublinear Rate

- What if we can't geometrically decrease the error?

If the errors satisfy

$$\mathbb{E}[||e_k||^2] \leq O(1/k^\gamma),$$

then the algorithm has an **expected weak sublinear** *rate,*

$$\mathbb{E}[f(x_k) - f(x_*)] = O(1/k^\gamma).$$

Hybrid Algorithm Expected Weak Sublinear Rate

- What if we can't geometrically decrease the error?

If the errors satisfy

$$\mathbb{E}[||e_k||^2] \leq O(1/k^\gamma),$$

then the algorithm has an **expected weak sublinear** *rate,*

$$\mathbb{E}[f(x_k) - f(x_*)] = O(1/k^\gamma).$$

Rough summary:

- *the algorithm converges at the same rate as the errors* (up to the speed of the deterministic algorithm).

Extensions and Future Work

We have generalized our analysis to a variety of scenarios:

- Newton-like scaling of the gradient (next section)
- Convex (but not necessarily strongly convex) objectives.
- Accelerated-gradient methods (faster rates of convergence).
- Projected-gradient methods (constrained optimization).
- Proximal-gradient methods (non-smooth optimization).

Extensions and Future Work

We have generalized our analysis to a variety of scenarios:

- Newton-like scaling of the gradient (next section)
- Convex (but not necessarily strongly convex) objectives.
- Accelerated-gradient methods (faster rates of convergence).
- Projected-gradient methods (constrained optimization).
- Proximal-gradient methods (non-smooth optimization).

There remain several other directions to explore:

- Mirror descent methods.
- Concentration bounds, quasi-random sampling.
- *Other applications where the gradient is measured with error.*

Outline

Application: Incremental Gradient Methods

- Many data fitting applications lead to problems of the form

$$\min_x \frac{1}{M} \sum_{i=1}^{M} f_i(x).$$

(i.e. maximum likelihood estimation from i.i.d. data)

Application: Incremental Gradient Methods

- Many data fitting applications lead to problems of the form

$$\min_x \frac{1}{M} \sum_{i=1}^{M} f_i(x).$$

  (i.e. maximum likelihood estimation from i.i.d. data)
- If M is very large, exact gradient calculation may be expensive.
- It is common to use a mini-batch gradient approximation

$$g(x_k) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x^k).$$

Application: Incremental Gradient Methods

- Many data fitting applications lead to problems of the form

$$\min_{x} \frac{1}{M} \sum_{i=1}^{M} f_i(x).$$

(i.e. maximum likelihood estimation from i.i.d. data)

- If M is very large, exact gradient calculation may be expensive.
- It is common to use a mini-batch gradient approximation

$$g(x_k) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x^k).$$

- With a fixed batch size, the convergence rate is *sublinear*.

Application: Incremental Gradient Methods

- Many data fitting applications lead to problems of the form

$$\min_x \frac{1}{M} \sum_{i=1}^{M} f_i(x).$$

  (i.e. maximum likelihood estimation from i.i.d. data)
- If M is very large, exact gradient calculation may be expensive.
- It is common to use a mini-batch gradient approximation

$$g(x_k) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x^k).$$

- With a fixed batch size, the convergence rate is *sublinear*.
- We can pick the batch sizes $|\mathcal{B}_k|$ to achieve a linear rate.

Incremental Gradient Method Error Bounds

Under standard assumptions on the $\nabla f_i(x)$, *we obtain*

$$f(x_k) - f(x_*) = O(\sigma^k),$$

for all $\sigma > \max\{1 - \mu/L, \gamma\}$ *by choosing* $|\mathcal{B}_k|$ *to satisfy*

$$|\mathcal{B}_k| = M - O(\gamma^{k/2}).$$

Incremental Gradient Method Error Bounds

Under standard assumptions on the $\nabla f_i(x)$*, we obtain*

$$f(x_k) - f(x_*) = O(\sigma^k),$$

for all $\sigma > \max\{1 - \mu/L, \gamma\}$ *by choosing* $|\mathcal{B}_k|$ *to satisfy*

$$|\mathcal{B}_k| = M - O(\gamma^{k/2}).$$

- The error decreases at *twice* the rate the batch size increases.

Incremental Gradient Method Error Bounds

Under standard assumptions on the $\nabla f_i(x)$*, we obtain*

$$f(x_k) - f(x_*) = O(\sigma^k),$$

for all $\sigma > \max\{1 - \mu/L, \gamma\}$ *by choosing* $|\mathcal{B}_k|$ *to satisfy*

$$|\mathcal{B}_k| = M - O(\gamma^{k/2}).$$

- The error decreases at *twice* the rate the batch size increases.
- This holds for *any sampling without* replacement scheme (but bound is better in expectation for uniform sampling).

Improved Rates with Newton-like Scaling

- The algorithm may converge slowly if μ/L is small.
- We can also analyze a Newton-like algorithm

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k is the solution of

$$H_k d_k = -g(x_k).$$

Improved Rates with Newton-like Scaling

- The algorithm may converge slowly if μ/L is small.
- We can also analyze a Newton-like algorithm

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k is the solution of

$$H_k d_k = -g(x_k).$$

- We can then show rates using a modified μ and L based on the Hessian approximation H_k.

Quasi-Newton Scaling and Heuristic Line Search

- In our implementation, we use the *L-BFGS* quasi-Newton Hessian approximation.

Quasi-Newton Scaling and Heuristic Line Search

- In our implementation, we use the *L-BFGS* quasi-Newton Hessian approximation.
- To choose the step size, we use the *Armijo* condition

$$\bar{f}(x_k + \alpha_k d_k) < \bar{f}(x_k) + \eta \alpha g(x_k)^T d_k,$$

on the sampled objective

$$\bar{f}(x_k) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} f_i(x_k).$$

Quasi-Newton Scaling and Heuristic Line Search

- In our implementation, we use the *L-BFGS* quasi-Newton Hessian approximation.
- To choose the step size, we use the *Armijo* condition

$$\bar{f}(x_k + \alpha_k d_k) < \bar{f}(x_k) + \eta \alpha g(x_k)^T d_k,$$

on the sampled objective

$$\bar{f}(x_k) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} f_i(x_k).$$

- By increasing the batch size this eventually reduces to a conventional line-search quasi-Newton method, inheriting the global and local convergence guarantees of this method.

Numerical Evaluation

We performed experiments comparing three algorithms:

- Deterministic: Conventional L-BFGS quasi-Newton method.
- Stochastic: Constant step-size stochastic gradient descent.
- Hybrid: An L-BFGS quasi-Newton method with batch size

$$|\mathcal{B}_{k+1}| = \lceil \min\{1.1 \cdot |\mathcal{B}_k| + 1, M\} \rceil.$$

Numerical Evaluation

We performed experiments comparing three algorithms:

- Deterministic: Conventional L-BFGS quasi-Newton method.
- Stochastic: Constant step-size stochastic gradient descent.
- Hybrid: An L-BFGS quasi-Newton method with batch size

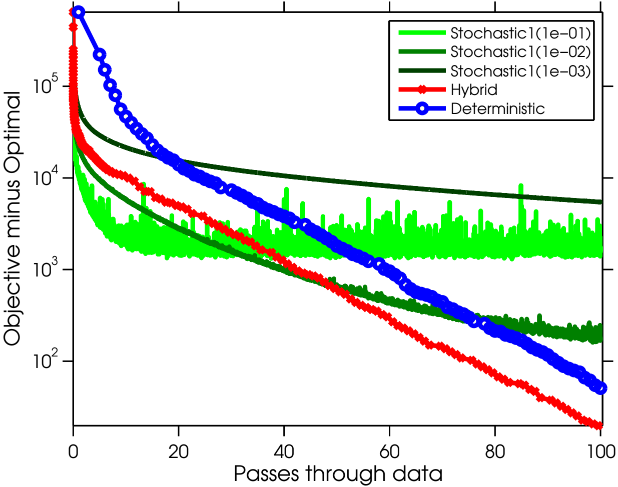
$$|\mathcal{B}_{k+1}| = \lceil \min\{1.1 \cdot |\mathcal{B}_k| + 1, M\} \rceil.$$

We trained conditional random fields (CRFs) on:

- The CoNLL-2000 noun-phrase chunking shared task (chain-structure).
- A binary image-denoising problem (lattice-structure).

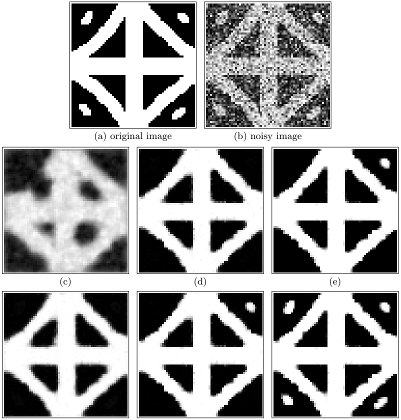
Evaluation on Chain-Structured CRFs

Results on chain-structured conditional random field:



Evaluation on Lattice-Structured CRF

Results on lattice-structured conditional random field:



(a) original image (b) noisy image

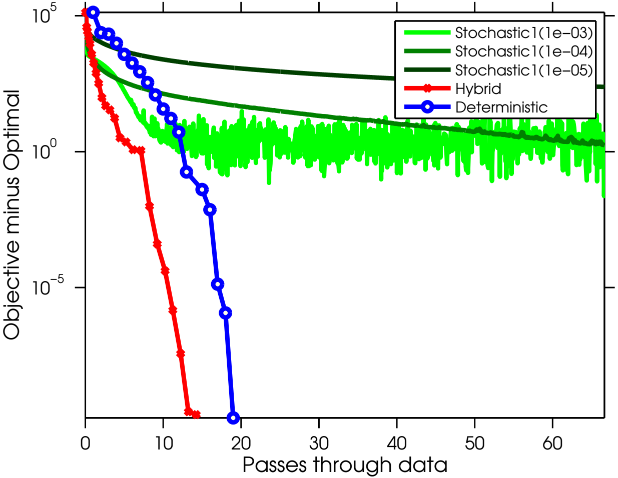
(c) (d) (e)

(f) (g) (h)

FIG. 5.5. *Top row: original (a) and noisy (b) image. Second row: marginals after 2 passes through the data for deterministic (c), stochastic (d), and hybrid (e). Third row: marginals after 5 passes through the data for deterministic (f), stochastic (g), and hybrid (h).*

Evaluation on Lattice-Structured CRFs

Results on lattice-structured conditional random field:



Outline

Optimization Costly Functions with Simple Constrains

We often have optimization problems with 3 complicating factors:

1. the number of parameters is large.
2. evaluating the objective is expensive.
3. the parameters have constraints.

Optimization Costly Functions with Simple Constrains

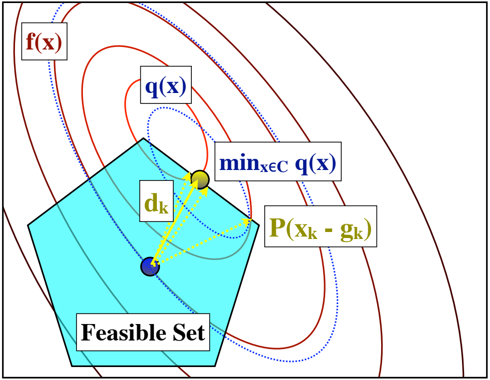
We often have optimization problems with 3 complicating factors:

1. the number of parameters is large.
2. evaluating the objective is expensive.
3. the parameters have constraints.

But, the constraints are simple.

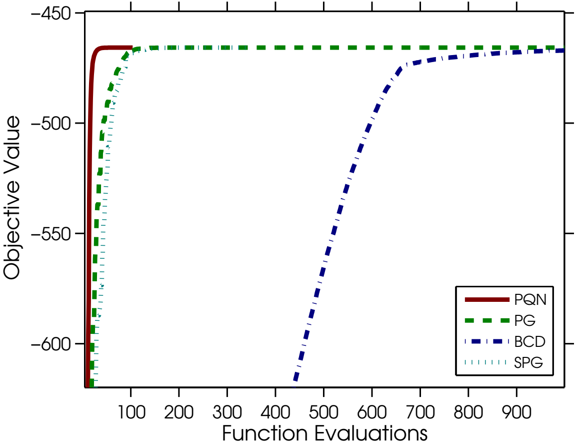
Optimization Costly Functions with Simple Constrains

We give a limited-memory inexact projected quasi-Newton algorithm for optimizing costly functions with simple constraints. [Schmidt, van den Berg, Friedlander, Murphy, 2009].



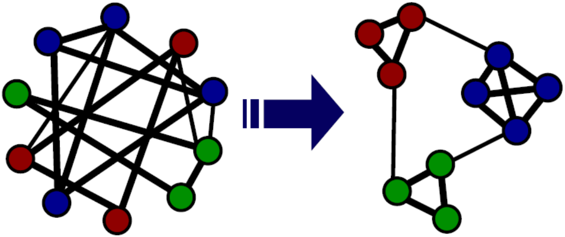
Optimization Costly Functions with Simple Constrains

Comparison of optimizers for fitting Gaussian graphical models with ℓ_1-regularization:



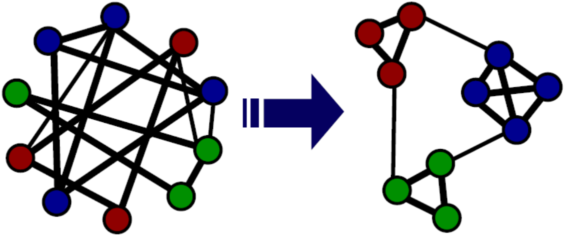
Group Sparse Priors for Covariance Estimation

- There has been work on group ℓ_1-regularization for structure learning in Gaussian graphical models with variable types:



Group Sparse Priors for Covariance Estimation

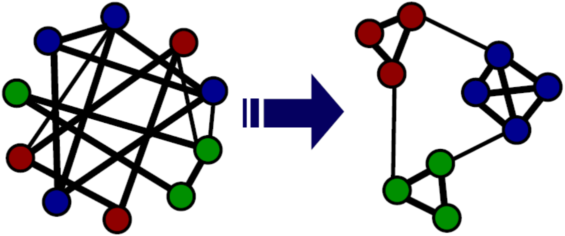
- There has been work on group ℓ_1-regularization for structure learning in Gaussian graphical models with variable types:



- What if we don't know the variable types?

Group Sparse Priors for Covariance Estimation

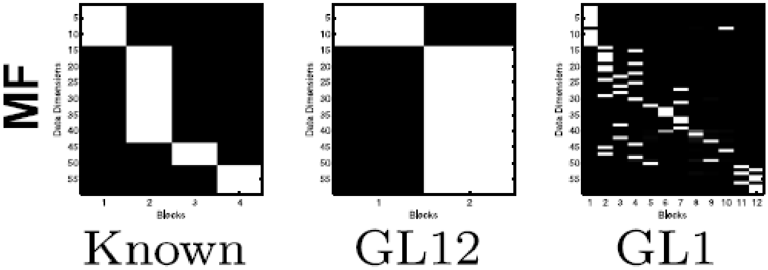
- There has been work on group ℓ_1-regularization for structure learning in Gaussian graphical models with variable types:



- What if we don't know the variable types?
- We give bounds on integrals of priors over positive-definite matrices, and a variational method that learns the types. [Marlin, Schmidt, Murphy, 2009]

Group Sparse Priors for Covariance Estimation

Learned variable types on mutual fund data:
[Scott & Carvalho, 2008]



The methods discover the ‘stocks’ and ‘bonds’ groups.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the ‘hungry at work’ problem:

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the 'hungry at work' problem:
  - If I see that my watch says 11:55, then it's almost lunch time

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the ‘hungry at work’ problem:
  - If I see that my watch says 11:55, then it’s almost lunch time
  - If I set my watch so it says 11:55, it doesn’t help

Causality: Modeling Interventions

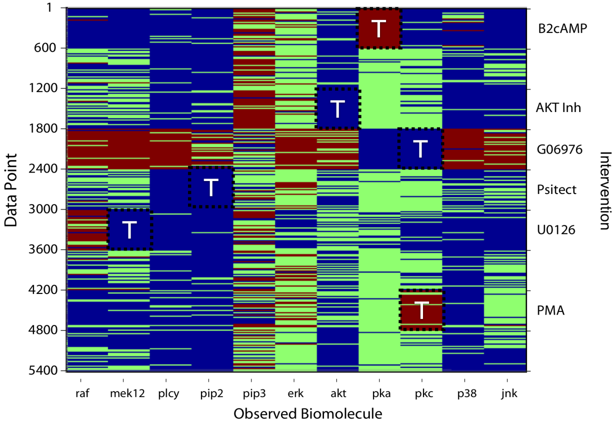
- The difference between conditioning by observation and conditioning by intervention in the ‘hungry at work’ problem:
  - If I see that my watch says 11:55, then it’s almost lunch time
  - If I set my watch so it says 11:55, it doesn’t help
- Without knowing the difference, predictions may be useless.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the ‘hungry at work’ problem:
  - If I see that my watch says 11:55, then it’s almost lunch time
  - If I set my watch so it says 11:55, it doesn’t help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

Interventional Cell Signaling Data [Sachs et al., 2005]



Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.

Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a ‘true’ underlying DAG.
- For real data, the structure may not be known, or even a DAG.

Causality: Modeling Interventions

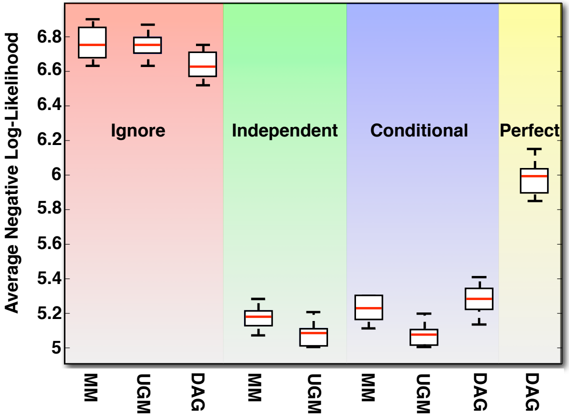
- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?

Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?
- Given this task, there are a variety of approaches to causality.
  [Eaton & Murphy, 2007]
  [Schmidt & Murphy, 2009]
  [Duvenaud, Eaton, Murphy, Schmidt, 2010]

Causality: Modeling Interventions

Interventional Cell Signaling Data [Sachs et al., 2005]:



Convex Structure Learning with Higher-Order Potentials

- Several authors have recently examined structure learning in graphical models with ℓ_1-regularization.
- Almost all of this work focuses on pairwise models.

Convex Structure Learning with Higher-Order Potentials

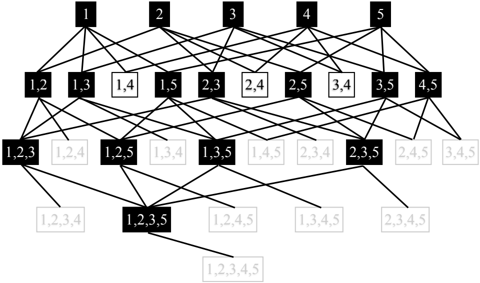
- Several authors have recently examined structure learning in graphical models with ℓ_1-regularization.
- Almost all of this work focuses on pairwise models.
- This is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.

Convex Structure Learning with Higher-Order Potentials

- Several authors have recently examined structure learning in graphical models with ℓ_1-regularization.
- Almost all of this work focuses on pairwise models.
- This is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.
- We give one way to go beyond pairwise potentials. [Schmidt & Murphy, 2010]

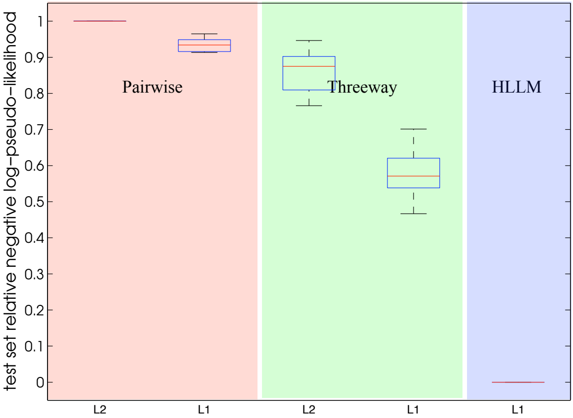
Convex Structure Learning with Higher-Order Potentials

- We focus on the special case of hierarchical models.
- We give a convex formulation that uses overlapping group ℓ_1-regularization to enforce the hierarchy.
- A heuristic hierarchical search allows us to tractably search the exponential number of possible higher-order potentials.



Convex Structure Learning with Higher-Order Potentials

Results on traffic flow data.
[Krause & Guestrin, 2005, Shahaf et al., 2009]

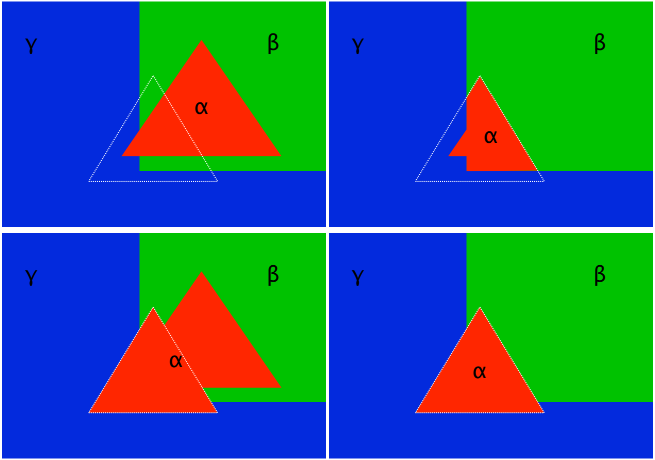


Generalized α-Expansions for Energy Minimization

- $\alpha\beta$-swaps and α-expansions are two minimum-cut methods for approximate MAP estimation in 'metric' graphical models.
- These both 'dominate' the classic ICM algorithm.
- But, neither dominates the other.
- We present a generalization of both moves that:
  - Dominates them both
  - Is still solvable in polynomial time.

Generalized α-Expansions for Energy Minimization

Example of α-expansion β-shrink move [Schmidt & Alahari, 2011]:



Generalized α-Expansions for Energy Minimization

Relative energy of local minima with respect to different moves.

Name	$\alpha\beta$-Swap	α-Expansion	New Moves
Family	1.0203	1	0.9998
Pano	1.3182	1	1
Tsukuba	1.0315	1	1.0000
Venus	1.8561	1	0.9968
Teddy	1.0037	1	0.9999
Penguin	1.1283	1	0.9758
House	0.7065	1	0.7032

Summary

- There are many instances of optimization problems where we can not compute the gradient exactly.

Summary

- There are many instances of optimization problems where we can not compute the gradient exactly.
- Most previous work on rate of convergence considers unbiased gradient error or a fixed error level.

Summary

- There are many instances of optimization problems where we can not compute the gradient exactly.
- Most previous work on rate of convergence considers unbiased gradient error or a fixed error level.
- We considered the case of a decreasing sequence of errors:
  - We analyze the rate of convergence under different sequences.
  - A practical quasi-Newton batching algorithm for maximum likelihood and related problems.

Summary

- There are many instances of optimization problems where we can not compute the gradient exactly.
- Most previous work on rate of convergence considers unbiased gradient error or a fixed error level.
- We considered the case of a decreasing sequence of errors:
  - We analyze the rate of convergence under different sequences.
  - A practical quasi-Newton batching algorithm for maximum likelihood and related problems.
- Code is on-line.

Summary

- There are many instances of optimization problems where we can not compute the gradient exactly.
- Most previous work on rate of convergence considers unbiased gradient error or a fixed error level.
- We considered the case of a decreasing sequence of errors:
  - We analyze the rate of convergence under different sequences.
  - A practical quasi-Newton batching algorithm for maximum likelihood and related problems.
- Code is on-line.
- Thank you for inviting me!