Numerical Mathematics in Machine Learning

Organizers:
John Cunningham, Stanford University
Matthias Seeger, Saarland University / MPI Informatics
Suvrit Sra, MPI Biological Cybernetics

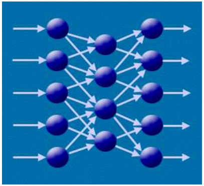
18 June 2009

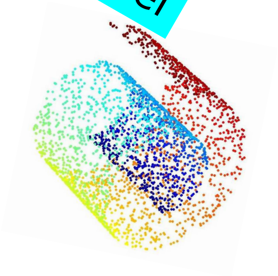
PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

The Old Days

Old Days

Toy Model





Fun!

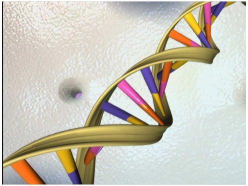
Toy Data

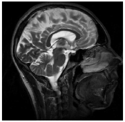
Today is Different

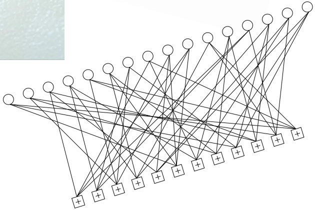
Today

Still Fun!

yet often...







anything but toy!

Layered Architectures

- Building big, complicated systems: Layered architecture
- Whatever your base layer: Make sure
  - it is robust (not “$\times$ fingers”)
  - to understand its limitations, how they affect you on top

Layered Architectures

- Building big, complicated systems: Layered architecture
- Whatever your base layer: Make sure
  - it is robust (not "$\times$ fingers")
  - to understand its limitations, how they affect you on top
- Not our business?
  - Nobody else will do this for us (but we can be helped)
  - Limits our scope up front

Layered Architectures

- Building big, complicated systems: Layered architecture
- Whatever your base layer: Make sure
  - it is robust (not "$\times$ fingers")
  - to understand its limitations, how they affect you on top
- Not our business?
  - Nobody else will do this for us (but we can be helped)
  - Limits our scope up front

(Almost) anything continuous-variable in ML:
Base layer is
Numerical Mathematics

Even Fancier Method

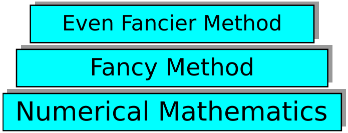
Fancy Method

Numerical Mathematics

Base Layer: Numerical Mathematics

Workshop Motivation

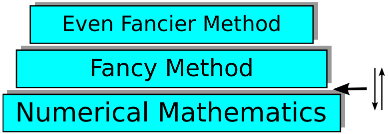
- Numerical Mathematics 101 for MLers (basic do's, don't's)



Base Layer: Numerical Mathematics

Workshop Motivation

- Numerical Mathematics 101 for MLers (basic do's, don't's)

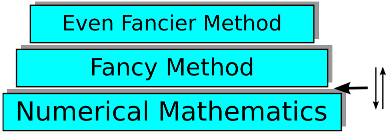


- NM offers some black boxes (dense matrix algebra). Black boxes are for solved problems, not for (most) ML: Understand interface to NM layer (ML→NM, NM→ML)

Base Layer: Numerical Mathematics

Workshop Motivation

- Numerical Mathematics 101 for MLers (basic do's, don't's)

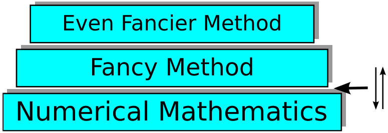


- NM offers some black boxes (dense matrix algebra). Black boxes are for solved problems, not for (most) ML: Understand interface to NM layer (ML→NM, NM→ML)
- Awareness: To
  - Faster, convex, smaller test error than X, Y on dataset Z,

Base Layer: Numerical Mathematics

Workshop Motivation

- Numerical Mathematics 101 for MLers (basic do's, don't's)



- NM offers some black boxes (dense matrix algebra). Black boxes are for solved problems, not for (most) ML: Understand interface to NM layer (ML→NM, NM→ML)
- Awareness: To
  - Faster, convex, smaller test error than X, Y on dataset Z,
  - Add: Numerically stable, reliable, reducible to well-solved problems
- What high-quality NM code is out there? Which ML-specific primitives does it serve?

Example I: Gaussian Means, (Co)Variances

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{\Psi}^{-1})$$

$$\mathrm{E}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\boldsymbol{\Psi})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\mathrm{Var}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2 \operatorname{diag}[(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\boldsymbol{\Psi})^{-1}]$$

- Means in Gaussian models? Least squares estimation

Example I: Gaussian Means, (Co)Variances

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{\Psi}^{-1})$$

$$\mathrm{E}[\boldsymbol{u}|\boldsymbol{y}] \approx \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\boldsymbol{\Psi})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\mathrm{Var}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2 \,\mathrm{diag}[(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\boldsymbol{\Psi})^{-1}]$$

- Means in Gaussian models? Least squares estimation
  - Structured models ($\boldsymbol{X}$) $\rightarrow$ Iterative solvers (CG)
  - Needs preconditioning, for $\boldsymbol{X}$ MLers care about [Malioutov]
  - Randomized approaches for huge systems [Mahoney]

Example I: Gaussian Means, (Co)Variances

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \Psi^{-1})$$

$$\mathrm{E}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2 (\boldsymbol{X}^T \boldsymbol{X} + \sigma^2 \Psi)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

$$\mathrm{Var}[\boldsymbol{u}|\boldsymbol{y}] \approx \sigma^2 \,\mathrm{diag}[(\boldsymbol{X}^T \boldsymbol{X} + \sigma^2 \Psi)^{-1} \boldsymbol{G}\boldsymbol{G}^T], \ \boldsymbol{G} \in \mathbb{R}^{n \times k}, \ k \ll n$$

- Means in Gaussian models? Least squares estimation
  - Structured models ($\boldsymbol{X}$) $\rightarrow$ Iterative solvers (CG)
  - Needs preconditioning, for $\boldsymbol{X}$ MLers care about [Malioutov]
  - Randomized approaches for huge systems [Mahoney]
- Variances in Gaussian models? Low-rank approximations
  - Projections based on model properties [Malioutov]
  - Projections based on PCA (Lanczos)

Example I: Gaussian Means, (Co)Variances

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{u} + \varepsilon, \quad \varepsilon \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \Psi^{-1})$$

$$\mathrm{E}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2 (\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\Psi)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\mathrm{Var}[\boldsymbol{u}|\boldsymbol{y}] \approx \sigma^2 \,\mathrm{diag}[(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\Psi)^{-1}\boldsymbol{G}\boldsymbol{G}^T], \; \boldsymbol{G} \in \mathbb{R}^{n\times k}, \; k \ll n$$

- Means in Gaussian models? Least squares estimation
  - Structured models ($\boldsymbol{X}$) $\rightarrow$ Iterative solvers (CG)
  - Needs preconditioning, for $\boldsymbol{X}$ MLers care about [Malioutov]
  - Randomized approaches for huge systems [Mahoney]
- Variances in Gaussian models? Low-rank approximations
  - Projections based on model properties [Malioutov]
  - Projections based on PCA (Lanczos)
- How about non-Gaussian models?
  Approximate inference methods reduce to
  Gaussian mean / variance computations

Example I: Gaussian Means, (Co)Variances

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{u} + \varepsilon, \quad \varepsilon \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u} \sim N(\boldsymbol{0}, \Psi^{-1})$$

$$\mathrm{E}[\boldsymbol{u}|\boldsymbol{y}] = \sigma^2 (\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\Psi)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\mathrm{Var}[\boldsymbol{u}|\boldsymbol{y}] \approx \sigma^2 \,\mathrm{diag}[(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\Psi)^{-1}\boldsymbol{G}\boldsymbol{G}^T], \ \boldsymbol{G} \in \mathbb{R}^{n\times k}, \ k \ll n$$

- Means in Gaussian models? Least squares estimation
    - Structured models ($\boldsymbol{X}$) $\rightarrow$ Iterative solvers (CG)
    - Needs preconditioning, for $\boldsymbol{X}$ MLers care about [Malioutov]
    - Randomized approaches for huge systems [Mahoney]
- Variances in Gaussian models? Low-rank approximations
    - Projections based on model properties [Malioutov]
    - Projections based on PCA (Lanczos)
- How about dynamical systems?
  Kalman filtering/smoothing reduces to
  Gaussian mean / covariance computations [Malioutov]

Example II: Eigendecomposition

$$\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q} = \Lambda \quad \Rightarrow \quad \boldsymbol{A} \approx \boldsymbol{Q} \Lambda \boldsymbol{Q}^T, \; \boldsymbol{A}^{-1} \approx \boldsymbol{Q} \Lambda^{-1} \boldsymbol{Q}^T, \; \ldots$$

- Eigendecomposition all over ML (PCA, CCA, spectral clustering, manifold regularization, posterior covariance approximation, . . .) [Dhillon]

Example II: Eigendecomposition

$$\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q} = \Lambda \quad \Rightarrow \quad \boldsymbol{A} \approx \boldsymbol{Q} \Lambda \boldsymbol{Q}^T,\ \boldsymbol{A}^{-1} \approx \boldsymbol{Q} \Lambda^{-1} \boldsymbol{Q}^T,\ \ldots$$

- Eigendecomposition all over ML (PCA, CCA, spectral clustering, manifold regularization, posterior covariance approximation, ...) [Dhillon]
- A lot of effort put into model ($\boldsymbol{A}$).
  Can its structure help to find $\boldsymbol{Q}$, Λ better than black box?
  - Preconditioning [Malioutov]
  - Make use of parallel hardware? [Gondzio]

Example III: Low-Rank Kernel Approximations

$$\boldsymbol{K} \approx \boldsymbol{K}_{\cdot,I} \boldsymbol{K}_I^{-1} \boldsymbol{K}_{I,\cdot}$$

- Kernel methods (SVM, GP, ...) use dense unstructured matrices: They just don't scale up
- Nyström method, incomplete Cholesky, ...
  But how do I select those columns in the best way?

Example III: Low-Rank Kernel Approximations

$$\boldsymbol{K} \approx \boldsymbol{K}_{\cdot,I} \boldsymbol{K}_I^{-1} \boldsymbol{K}_{I,\cdot}$$

- Kernel methods (SVM, GP, . . .) use dense unstructured matrices: They just don't scale up
- Nyström method, incomplete Cholesky, . . .
  But how do I select those columns in the best way?
- This problem has just another name in NM! [Mahoney]

Example IV: Interior Point Methods

- IPM: Reduce convex optimization to solving many linear systems (Newton on objective $+$ barrier)
- For ML: Black box packages not an option (recall rise of SVM?) $\Rightarrow$ Use first-order methods, right?

Example IV: Interior Point Methods

- IPM: Reduce convex optimization to solving many linear systems (Newton on objective + barrier)
- For ML: Black box packages not an option (recall rise of SVM?) $\Rightarrow$ Use first-order methods, right?
- Abandon black box IPMs, don't abandon IPMs [Gondzio]

Example IV: Interior Point Methods

- IPM: Reduce convex optimization to solving many linear systems (Newton on objective $+$ barrier)
- For ML: Black box packages not an option (recall rise of SVM?) $\Rightarrow$ Use first-order methods, right?
- Abandon black box IPMs, don't abandon IPMs [Gondzio]
  - How can model structure be exploited in IPMs?
    Harder than in first-order methods, but worth it
  - Blocking in IPMs: What should I decompose, what not?
  - Difference between algorithms in terms of stability?
  - Impact of approximate Newton directions? Preconditioning?

Have A Great Workshop!

Workshop Motivation

- Numerical Mathematics 101 for MLers
- Understand interface to NM layer
- Awareness/relevance of numerical properties
- NM code MLers should know about

Have A Great Workshop!

Workshop Motivation

- Numerical Mathematics 101 for MLers $\Leftarrow$ [NM$\leftrightarrow$ML]
- Understand interface to NM layer
- Awareness/relevance of numerical properties
- NM code MLers should know about

Have A Great Workshop!

Workshop Motivation

- Numerical Mathematics 101 for MLers $\Leftarrow$ [NM$\leftrightarrow$ML]
- Understand interface to NM layer $\Leftarrow$ [NM$\leftrightarrow$ML]
- Awareness/relevance of numerical properties
- NM code MLers should know about

Have A Great Workshop!

Workshop Motivation

- Numerical Mathematics 101 for MLers $\Leftarrow$ [NM$\leftrightarrow$ML]
- Understand interface to NM layer $\Leftarrow$ [NM$\leftrightarrow$ML]
- Awareness/relevance of numerical properties $\Leftarrow$ [NM$\leftrightarrow$ML]
- NM code MLers should know about

Have A Great Workshop!

Workshop Motivation

- Numerical Mathematics 101 for MLers $\Leftarrow$ [NM$\leftrightarrow$ML]
- Understand interface to NM layer $\Leftarrow$ [NM$\leftrightarrow$ML]
- Awareness/relevance of numerical properties $\Leftarrow$ [NM$\leftrightarrow$ML]
- NM code MLers should know about $\Leftarrow$ [NM$\leftrightarrow$ML]