Semi-definite Programming and Nuclear Magnetic Resonance

Christopher Kumar Anand
Anuroop Sharma

http://sqrl.mcmaster.ca/~anand/papers/preprints.html
SDP + NMR

- Applications
- Imaging
- NMR
- Problem: Maximize Signal
- Trust-Region Method + CSDP
- Results
Applications: Imaging

- Fourier Transform
- MRI
- Radon Transform
- CT, PET, SPECT, EPR
s(t) = ∫_{R^3} e^{i⟨x, k(t)⟩} ρ(x) dx.
Different Under Sampling

Different Errors
Line-of-Sight

• CT, PET, SPECT, EPR sample projections

• FT(1d Projection) = FT(image) | line

• Constraint on Sampling
Problem Sneak Peak

- Inverse Problem with Noisy Data
- Minimize Expected Reconstructed Noise
Application: NMR

- Know DNA Sequences
- Defines Strings of Amino Acids
- Missing Info:
  - Protein Structure
    - only works if folded
  - Protein Function
    - interaction = wiggling

http://en.wikipedia.org/wiki/Protein
Protein NMR

• Protein Structure

• 2 methods:
  • heteronuclear, multi-dimensional NMR
  • x-ray Crystallography (faster?)

• Protein Dynamics

• 1 method
  • repeated n-d NMR
Basic NMR

- radio-frequency magnetic field excites spins
- high-energy state decays producing free-induction decay (direct dimension)
2-d NMR

- pulse @ 2 frequencies
- transfer spin state H-N-H
- phase variation proportional to delay (indirect)
• regular sampling + FFT

• contour with x-section

• clustered peaks (lorentzian or gaussian)
n-d NMR

• induction on number of colours in diagram

• since delays are positive, can only sample positive values in indirect dimensions

• sampling in indirect dimensions is expensive

• days, weeks, months, years for full
Protein Dynamics

• in n-d 1 peak per residue
• no overlap for n=4,5,...,10?
• measure with additional delay
  • signal of each peak decays exponentially
• rate linked to mobility
Questions

• How many samples do we need to estimate 200 peak areas?

• Is regular sampling for FFT optimal?
Slow FT Inverse Problem

\[ \hat{f}(k_i) = \sum_{j=1}^{m} f(x_j) e^{\sqrt{-1} \langle k_i, x_j \rangle} \]

- x - known peaks
- k - samples
- general form still a linear system (S)
- Moore-Penrose

\[ S_{i,j} = e^{\sqrt{-1} \langle k_i, x_j \rangle} \]

\[
\begin{pmatrix}
\hat{f}(k_1) \\
\vdots \\
\hat{f}(k_n)
\end{pmatrix} = S
\begin{pmatrix}
f(x_1) \\
\vdots \\
f(x_m)
\end{pmatrix} + \begin{pmatrix}
\epsilon_1 \\
\vdots \\
\epsilon_n
\end{pmatrix}
\]

\[
\begin{pmatrix}
f(x_1) \\
\vdots \\
f(x_m)
\end{pmatrix} = (S^* S)^{-1} S^* \begin{pmatrix}
\hat{f}(k_1) \\
\vdots \\
\hat{f}(k_n)
\end{pmatrix}
\]
Noise ~ Conditioning

\[(S^* S)_{i,j} = \sum_{l=1}^{n} e^{-1}(k_l, x_j - x_i)\]

• expected maximum error
  \(\sim 1/\) minimal eigenvalue

• leads to semi-definite constraint

see
Real Nonlinear Problem

\[
\min_{\{k_i\}} \quad -\lambda
\]

subject to \( A - \lambda I \succeq 0 \)

\[
A_{2i-1,2j-1} = \sum_{l=1}^{n} \cos\langle k_l, x_j - x_i \rangle
\]

\[
A_{2i,2j} = \sum_{l=1}^{n} \cos\langle k_l, x_j - x_i \rangle
\]

\[
A_{2i,2j-1} = \sum_{l=1}^{n} \sin\langle k_l, x_j - x_i \rangle
\]

\[
A_{2i-1,2j} = -\sum_{l=1}^{n} \sin\langle k_l, x_j - x_i \rangle
\]
Trust Region

• general non-linear solvers do not use semi-definite cone structure

• use trust region method with linear problem

• shape trust region relative to sensitivity
Linear Subproblem

\[
\min_k \quad -\lambda
\]

subject to \( A|_{\tilde{k}} + \sum_{\alpha = 1 \ldots n}^{\beta = 1 \ldots r} (k_{\alpha,\beta} - \tilde{k}_{\alpha,\beta}) \frac{\partial A}{\partial k_{\alpha,\beta}} \bigg|_{\tilde{k}} - \lambda I \geq 0. \)

\[
\frac{\partial A_{2i-1,2j-1}}{\partial k_{\alpha,\beta}} = - (\sin(k_{\alpha,\beta} - x_i))(x_{j,\beta} - x_{i,\beta})
\]

\[
\frac{\partial A_{2i,2j}}{\partial k_{\alpha,\beta}} = - (\sin(k_{\alpha,\beta} - x_i))(x_{j,\beta} - x_{i,\beta})
\]

\[
\frac{\partial A_{2i,2j-1}}{\partial k_{\alpha,\beta}} = (\cos(k_{\alpha,\beta} - x_i))(x_{j,\beta} - x_{i,\beta})
\]

\[
\frac{\partial A_{2i-1,2j}}{\partial k_{\alpha,\beta}} = - (\cos(k_{\alpha,\beta} - x_i))(x_{j,\beta} - x_{i,\beta})
\]

\[
|k_{\alpha,\beta} - \tilde{k}_{\alpha,\beta}| \leq \frac{\pi/2}{\max|x_{j,\beta} - x_{i,\beta}|}
\]
Implementation

- C program calling CSDP for subproblem
- use random, and greedy-random seeding in incremental and one-step solvers
Hyperplane Decomposition

- CSDP cannot solve full problem
- use dense H-freq sampling
- reduce dimension of parameters and variables
- only optimize hyperplanes with peaks
Hyperplane Decomposition - II

\[
(S^* S)_{i,j} = \begin{cases} 
\sum_{l=0}^{n} e^{1/(k_l, x_j - x_i)} & \text{if } x_j, x_i \text{ belongs to plane } l \\
0 & \text{otherwise}
\end{cases}
\]

- block diagonal structure
- still to large
- separate blocks (hyperplanes) into independent problems
Numerical Tests

• used peak positions for protein R1a (119-244)

• omitted residues with missing frequencies

• clustered peaks into fat 2D hyperplanes
Robust in 3D

- greedy random optimization limited to 80%
- continuous optimization consistently better
- 2x more samples required with greedy approach
**Better in Higher Dimensions**

- 17 peaks with full frequency information
- Efficiency increases with dimension (34 samples)
- Fewer samples required in higher dimensions (conventionally grows exponentially)
Full Problem

<table>
<thead>
<tr>
<th>plane</th>
<th>number of peaks</th>
<th>number of samples</th>
<th>efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7+2</td>
<td>26</td>
<td>0.99</td>
</tr>
<tr>
<td>2</td>
<td>38+2</td>
<td>119</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>26+2</td>
<td>83</td>
<td>0.82</td>
</tr>
<tr>
<td>4</td>
<td>15+2</td>
<td>50</td>
<td>0.87</td>
</tr>
<tr>
<td>5</td>
<td>22+2</td>
<td>71</td>
<td>0.80</td>
</tr>
<tr>
<td>6</td>
<td>2+2</td>
<td>11</td>
<td>1.00</td>
</tr>
</tbody>
</table>

- overall 88.2% efficient
- 100-fold reduction in sampling for equal noise
Conclusion

• NMR dynamics
  • significant potential cost savings
• NMR structure
  • more complicated prior information
• imaging
  • dimension limited 3d
  • practical problems not yet solvable
• tough dense SDP problems available
Thanks

- Chemistry
- Alex Bain
- Giuseppe Melacini
- Rahul Das
- Advanced Optimization Lab
- Imre Polik
- CSDP
- Brian Borchers