Large Scale Matrix Analysis and Inference

Wouter M. Koolen - Manfred Warmuth
Reza Bosagh Zadeh - Gunnar Carlsson - Michael Mahoney

Dec 9, NIPS 2013
Introductory musing — What is a matrix?

1. A vector of $n^2$ parameters
2. A covariance
3. A generalized probability distribution
4. ...
1. A vector of $n^2$ parameters

When you regularize with the squared Frobenius norm

$$\min_W \|W\|_F^2 + \sum_n \text{loss} (\text{tr}(WX_n))$$
1. A vector of $n^2$ parameters

When you regularize with the squared Frobenius norm

$$\min_W ||W||_F^2 + \sum_n \text{loss}(\text{tr}(WX_n))$$

Equivalent to

$$\min_{\text{vec}(W)} ||\text{vec}(W)||_2^2 + \sum_n \text{loss}(\text{vec}(W) \cdot \text{vec}(X_n))$$

**No structure: $n^2$ independent variables**
View the symmetric positive definite matrix $C$ as a covariance matrix of some random feature vector $c \in \mathbb{R}^n$, i.e.

$$C = \mathbb{E} \left( (c - \mathbb{E}(c))(c - \mathbb{E}(c))^{\top} \right)$$

$n$ features plus their pairwise interactions
Symmetric matrices as ellipses

- **Ellipse** = \( \{ Cu : \|u\|_2 = 1 \} \)
- Dotted lines connect point \( u \) on unit ball with point \( Cu \) on ellipse
Symmetric matrices as ellipses

- Eigenvectors form **axes**
- Eigenvalues are lengths
Dyads

$uu^T$, where $u$ unit vector

- One eigenvalue one
- All others zero
- Rank one projection matrix
Directional variance along direction $\mathbf{u}$

$$\nabla(\mathbf{c}^\top \mathbf{u}) = \mathbf{u}^\top \mathbf{C} \mathbf{u} = \text{tr}(\mathbf{C} \mathbf{u} \mathbf{u}^\top) \geq 0$$

The outer figure eight is direction $\mathbf{u}$ times the variance $\mathbf{u}^\top \mathbf{C} \mathbf{u}$

PCA: find direction of largest variance
$\text{tr}(\mathbf{C uu}^\top)$ is generalized probability when $\text{tr}(\mathbf{C}) = 1$
3. Generalized probability distributions

Probability vector

\[ \omega = (\cdot.2, \cdot.1, \cdot.6, \cdot.1)^\top \]
\[ = \sum_i \omega_i \]

Density matrix

\[ W = \sum_i \omega_i \]
\[ e_i \]

Mixture coefficients

Pure events

Pure density matrices
3. Generalized probability distributions

Probability vector
\[ \omega = (0.2, 0.1, 0.6, 0.1)^\top \]
\[ = \sum_i \omega_i \]

\( \omega_i \) mixture coefficients
\( e_i \) pure events

Density matrix
\[ W = \sum_i \omega_i w_i w_i^\top \]

\( \omega_i \) mixture coefficients
\( w_i w_i^\top \) pure density matrices

Matrices as generalized distributions
3. Generalized probability distributions

**Probability vector**

\[ \omega = (0.2, 0.1, 0.6, 1)^\top \]

\[ = \sum_i \omega_i \]

- mixture coefficients
- pure events

**Density matrix**

\[ W = \sum_i \omega_i w_i w_i^\top \]

- mixture coefficients
- pure density matrices

**Matrices as generalized distributions**

- Many mixtures lead to same density matrix

\[ 0.2 + 0.3 \triangleleft + 0.5 = \begin{pmatrix} 0.35 & 0.15 \\ 0.15 & 0.65 \end{pmatrix} = \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \]

- There always exists a decomposition into \textit{n eigendyads}

- Density matrix: Symmetric positive matrix of trace one
It’s like a probability!

Total variance along orthogonal set of directions is 1

\[ u_1^T W u_1 + u_2^T W u_2 = 1 \]

\[ a + b + c = 1 \]
Uniform density?

- All dyads have generalized probability $\frac{1}{n}$
  
  \[
  \text{tr} \left( \frac{1}{n} uu^\top \right) = \frac{1}{n} \text{tr}(uu^\top) = \frac{1}{n}
  \]

- Generalized probabilities of $n$ orthogonal dyads sum to 1
Conventional Bayes Rule

\[ P(M_i|y) = \frac{P(M_i)P(y|M_i)}{P(y)} \]

- **4 updates** with the same data likelihood
- Update maintains uncertainty information about maximum likelihood
- **Soft max**
Conventional Bayes Rule

\[ P(M_i | y) = \frac{P(M_i)P(y | M_i)}{P(y)} \]

- **4 updates** with the same data likelihood
- Update maintains uncertainty information about maximum likelihood
- **Soft max**
Conventional Bayes Rule

\[ P(M_i|y) = \frac{P(M_i)P(y|M_i)}{P(y)} \]

- 4 updates with the same data likelihood
- Update maintains uncertainty information about maximum likelihood
- Soft max
Conventional Bayes Rule

\[ P(M_i|y) = \frac{P(M_i)P(y|M_i)}{P(y)} \]

- **4 updates** with the same data likelihood
- **Update maintains uncertainty information about maximum likelihood**
- **Soft max**
Bayes Rule for density matrices

\[ D(M|y) = \frac{\exp (\log D(M) + \log D(y|M))}{\text{tr (above matrix)}} \]

- **1 update with data likelyhood matrix** \( D(y|M) \)
- **Update maintains uncertainty information about maximum eigenvalue**
- **Soft max eigenvalue calculation**
Bayes Rule for density matrices

\[ D(M|y) = \frac{\exp \left( \log D(M) + \log D(y|M) \right)}{\text{tr} \ (\text{above matrix})} \]

- 2 updates with same data likelihood matrix \( D(y|M) \)
- Update maintains uncertainty information about maximum eigenvalue
- **Soft max eigenvalue calculation**
Bayes Rule for density matrices

\[
D(M|y) = \frac{\exp \left( \log D(M) + \log D(y|M) \right)}{\text{tr} \ (\text{above matrix})}
\]

- 3 updates with same data likelihood matrix \( D(y|M) \)
- Update maintains uncertainty information about maximum eigenvalue
- Soft max eigenvalue calculation
Bayes Rule for density matrices

\[ D(M|y) = \exp \left( \log D(M) + \log D(y|M) \right) / \text{tr (above matrix)} \]

- 4 updates with same data likelihood matrix \( D(y|M) \)
- Update maintains uncertainty information about maximum eigenvalue
- **Soft max eigenvalue calculation**
Bayes Rule for density matrices

\[ D(M|y) = \exp\left(\log D(M) + \log D(y|M)\right) \]

\[ \text{tr (above matrix)} \]

- 10 updates with same data likelihood matrix \( D(y|M) \)
- Update maintains uncertainty information about maximum eigenvalue
- Soft max eigenvalue calculation
Bayes Rule for density matrices

\[ D(M|y) = \frac{\exp \left( \log D(M) + \log D(y|M) \right)}{\text{tr (above matrix)}} \]

- 20 updates with same data likelihood matrix \( D(y|M) \)
- Update maintains uncertainty information about maximum eigenvalue
- Soft max eigenvalue calculation
Bayes’ rules

<table>
<thead>
<tr>
<th>Bayes rule</th>
<th>Vector equation</th>
<th>Matrix equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(M_i</td>
<td>y) = \frac{P(M_i) \cdot P(y</td>
<td>M_i)}{\sum_j P(M_j) \cdot P(y</td>
</tr>
</tbody>
</table>

$\bullet \circ \mathbf{B} := \exp(\log \mathbf{A} + \log \mathbf{B})$
### Bayes’ rules

<table>
<thead>
<tr>
<th></th>
<th>Vector</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bayes rule</strong></td>
<td>( P(M_i</td>
<td>y) = \frac{P(M_i) \cdot P(y</td>
</tr>
<tr>
<td><strong>Regularizer</strong></td>
<td>Entropy</td>
<td>Quantum Entropy</td>
</tr>
</tbody>
</table>

\( \odot \) denotes the matrix product and \( \oplus \) denotes the quantum product.

**Entropy** and **Quantum Entropy** are defined as:

\[ H(X) = -\sum_x p(x) \log p(x) \]

\[ S(X) = -\sum_x p(x) S(x) \]

where \( S(x) \) is the entropy of the state \( x \).
Vectors as diagonal matrices

All matrices same eigensystem

Fancy $\odot$ becomes $\cdot$

Often the hardest problem
ie bounds for the vector case “lift” to the matrix case
Vector case as special case of matrix case

- Vectors as diagonal matrices
- All matrices same eigensystem
- Fancy ⊙ becomes ·

- Often the hardest problem
  ie bounds for the vector case “lift” to the matrix case
- This phenomenon has been dubbed the “free matrix lunch”

Size of matrix = size of vector = $n$
PCA setup

Data vectors $\mathbf{c} = \sum_n \mathbf{x}_n \mathbf{x}_n^\top$

\[
\max_{\text{unit } \mathbf{u}} \quad \mathbf{u}^\top \mathbf{c} \mathbf{u} \quad \text{not convex in } \mathbf{u}
\]

\[
\max_{\text{dyad } \mathbf{u} \mathbf{u}^\top} \quad \text{tr}(\mathbf{c} \mathbf{u} \mathbf{u}^\top)
\]

Corresponding vector problem

\[
\max_{\mathbf{e}_i} \quad \mathbf{c}^\top \mathbf{e}_i \quad \text{linear in } \mathbf{e}_i
\]

Vector problem is matrix problem when everything happens in the same eigensystem

Uncertainty over unit: probability vector
Uncertainty over dyads: density matrix
Uncertainty over $k$-sets of units: capped probability vector
Uncertainty over rank $k$ projection matrices: capped density matrix
For PCA

- Solve the vector problem first
- Do all bounds
- Lift to matrix case: essentially replace $\cdot$ by $\odot$
- Regret bounds stay the same
- Free Matrix Lunch
Questions

- When can you “lift” vector case to matrix case?
- When is there a free matrix lunch?
- Lifting matrices to tensors?
- Efficient algorithms for large matrices?
  - Approximations of ⊙
  - Avoid eigenvalue decomposition by sampling
  - ...