Principal Component Analysis (PCA) for Sparse High-Dimensional Data

Tapani Raiko, Alexander Ilin, and Juha Karhunen
Helsinki University of Technology, Finland
Adaptive Informatics Research Center
Principal Component Analysis

• Data X consists of n d-dimensional vectors

• Matrix X is decomposed into a product of smaller matrices such that the square reconstruction error is minimized

\[ X \approx AS, \]

\[ C = \|X - AS\|_F^2 = \sum_{i=1}^{d} \sum_{j=1}^{n} (x_{ij} - \sum_{k=1}^{c} a_{ik}s_{kj})^2 \]
Algorithms for PCA

• Eigenvalue decomposition (standard approach)
  • Compute the covariance matrix and its eigenvectors
Algorithms for PCA

- Eigenvalue decomposition (standard approach)
  - Compute the covariance matrix and its eigenvectors

- EM algorithm
  - Iterates between:

\[ A \leftarrow X S^T (SS^T)^{-1}, \quad S \leftarrow (A^T A)^{-1} A^T X. \]
Algorithms for PCA

- Eigenvalue decomposition (standard approach)
  - Compute the covariance matrix and its eigenvectors

- EM algorithm
  - Iterates between:

\[
A \leftarrow XS^T(SS^T)^{-1}, \quad S \leftarrow (A^T A)^{-1} A^T X.
\]

- Minimization of cost C (Oja’s subspace rule)

\[
A \leftarrow A + \gamma(X - AS)S^T, \quad S \leftarrow S + \gamma A^T(X - AS).
\]
PCA with Missing Values

- Red and blue data points are reconstructed based on only one of the two dimensions
Adapting the Algorithms for Missing Values

- Iterative imputation

- Alternately 1) fill in missing values and 2) solve normal PCA with the standard approach

- EM algorithm becomes computationally heavier

\[
\begin{align*}
\mathbf{s}_{:j} &= (\mathbf{A}_j^T \mathbf{A}_j)^{-1} \mathbf{A}_j^T \hat{\mathbf{X}}_{:j} \\
&= 1, \ldots, n \\
\mathbf{A}_{i:}^T &= \hat{\mathbf{X}}_{i:}^T \mathbf{S}_i^T (\mathbf{S}_i \mathbf{S}_i^T)^{-1} \\
&= 1, \ldots, d
\end{align*}
\]

- Minimization of cost \( C \)

- Easy to adapt: Take error over observed values only
Speeding up Gradient Descent

• Newton’s method is known to converge fast, but
  • It requires computing the Hessian matrix which is computationally too demanding in high-dimensional problems

• We propose using only the diagonal part of the Hessian

• We also include a control parameter to interpolate between standard gradient descent (0) and the diagonal Newton’s method (1)
The cost function:
\[ C = \sum_{(i,j) \in O} e_{ij}^2, \text{ with } e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik}s_{kj}. \]
The cost function:

\[ C = \sum_{(i,j) \in O} e_{ij}^2 \]

with \[ e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik} s_{kj} \].

Its partial derivatives:

\[ \frac{\partial C}{\partial a_{il}} = -2 \sum_{j \mid (i,j) \in O} e_{ij} s_{lj} \]

\[ \frac{\partial C}{\partial s_{lj}} = -2 \sum_{i \mid (i,j) \in O} e_{ij} a_{il} \].
The cost function:

\[ C = \sum_{(i,j) \in O} e_{ij}^2, \quad \text{with} \quad e_{ij} = x_{ij} - \sum_{k=1}^{c} a_{ik}s_{kj}. \]

Its partial derivatives:

\[ \frac{\partial C}{\partial a_{il}} = -2 \sum_{j \mid (i,j) \in O} e_{ij}s_{lj}, \quad \frac{\partial C}{\partial s_{lj}} = -2 \sum_{i \mid (i,j) \in O} e_{ij}a_{il}. \]

Update rules:

\[ a_{il} \leftarrow a_{il} - \gamma' \left( \frac{\partial^2 C}{\partial a_{il}^2} \right)^{-\alpha} \frac{\partial C}{\partial a_{il}} = a_{il} + \gamma \frac{\sum_{j \mid (i,j) \in O} e_{ij}s_{lj}}{\left( \sum_{j \mid (i,j) \in O} s_{lj}^2 \right)^{\alpha}}, \]

\[ s_{lj} \leftarrow s_{lj} - \gamma' \left( \frac{\partial^2 C}{\partial s_{lj}^2} \right)^{-\alpha} \frac{\partial C}{\partial s_{lj}} = s_{lj} + \gamma \frac{\sum_{i \mid (i,j) \in O} e_{ij}a_{il}}{\left( \sum_{i \mid (i,j) \in O} a_{il}^2 \right)^{\alpha}}. \]
Overfitting in Case of Sparse Data

Overfitted solution

Regularized solution
Regularization against Overfitting

- Penalizing the use of large parameter values
- Estimating the distribution of unknown parameters (Variational Bayesian learning)
Experiments with Netflix Data

www.netflixprize.com

• Collaborative filtering task: predict people’s preferences based on other people’s preferences

• $d = 18\,000$ movies, $n = 500\,000$ customers, $N = 100\,000\,000$ movie ratings from 1 to 5

• 98.8% of the values are missing

• Find $c=15$ principal components
## Computational Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
<th>Seconds/Iter</th>
<th>Hours to $E_O = 0.85$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>$O(Nc + nc)$</td>
<td>58</td>
<td>1.9</td>
</tr>
<tr>
<td>Speed-up</td>
<td>$O(Nc + nc)$</td>
<td>110</td>
<td>0.22</td>
</tr>
<tr>
<td>Natural Grad.</td>
<td>$O(Nc + nc^2)$</td>
<td>75</td>
<td>3.5</td>
</tr>
<tr>
<td>Imputation</td>
<td>$O(nd^2)$</td>
<td>110000</td>
<td>$\gg 64$</td>
</tr>
<tr>
<td>EM</td>
<td>$O(Nc^2 + nc^3)$</td>
<td>45000</td>
<td>58</td>
</tr>
</tbody>
</table>

- $N = 100\,000\,000$, # of ratings
- $n = 500\,000$, # of people
- $c = 15$, # of components
- $d = 18\,000$, # of movies
Error on Training Data against computation time in hours
Error on Validation Data against computation time in hours
Summary

• PCA with sparse data and high dimensionality has two problems that require attention:
  • Standard algorithms are computationally inefficient
  • Oversized model does not generalize well to new data

• We proposed solutions to both problems:
  • Using gradient descent to minimize the cost, and speeding it up by an approximated Newton’s method
  • Regularization of the model by Variational Bayesian learning