Finding **low-dimensional** structure in **large-scale** neural datasets

Eva Dyer

Department of Biomedical Engineering
Georgia Institute of Technology //
Emory University

NeuroHackademy - Aug 3 2018
neural data deluge

fMRI

Voxel time courses

calcium imaging

DTI

eeg

extracellular arrays
why reduce dimensionality?

- compression
- denoising
- interpret complex data
low-dimensional models

\[ Y = [y_1 \ldots y_N] \]

\[ d = \# \text{ of neurons} \]
\[ N = \# \text{ of time points} \]
low-dimensional models

\[ Y = [y_1 \ldots y_N] \]

\( d = \# \text{ of neurons} \)

\( N = \# \text{ of time points} \)
low rank model

\[ \min_A \|Y - A\|_F \quad \text{s.t.} \quad \text{rank}(A) \leq k \]
low rank model

\[
\min_A \|Y - A\|_F \quad \text{s.t.} \quad \text{rank}(A) \leq k
\]

- linear subspace
- PCA

data matrix
low rank model

linear subspace

PCA

\[
\min_{\mathbf{A}} \| \mathbf{Y} - \mathbf{A} \|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq k
\]

"low rank" approximation
low rank model

\[ \min_{\mathbf{A}} \| \mathbf{Y} - \mathbf{A} \|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq k \]

subject to = constraints
low rank model

\[
\min_{\mathbf{A}} \| \mathbf{Y} - \mathbf{A} \|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq k
\]

\[\text{rank}(\mathbf{A}) = ?\]
low rank model

\[
\min_A \| \mathbf{Y} - \mathbf{A} \|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq k
\]

\[
\text{rank}(\mathbf{A}) = ?
\]
low rank model

\[
\min_A \| Y - A \|_F \quad \text{s.t.} \quad \text{rank}(A) \leq k
\]

\[
A = U_k S_k V_k^T \quad \text{(truncated SVD)}
\]
PCA:
1. Compute the covariance matrix (C)
2. Compute eigenvalue decomposition of C
3. Output > top k eigenvectors and their eigenvalues

Covariance matrix

\[ C = (Y - \bar{y})^T(Y - \bar{y})^T \]
1. Robust PCA - sparse LARGE errors
2. Factor analysis (FA) - noise of unequal variance
3. Non-negative matrix factorization (NMF)

\[ y = Ax + n \]
extensions of PCA

1. Robust PCA - sparse errors
2. **Factor analysis (FA) - noise of unequal variance**
3. Non-negative matrix factorization (NMF)

\[ y = Ax + n \]
extensions of PCA

1. Robust PCA - sparse errors
2. Factor analysis (FA) - noise of unequal variance
3. **Non-negative matrix factorization (NMF)**

Data and PCs are non-negative!
nonlinear models (manifolds)

\[
\min_{\mathcal{P}} |d(y_i, y_j) - d(\mathcal{P}y_i, \mathcal{P}y_j)|
\]

distance between original data points

nonlinear manifold
Isomap, LLE
nonlinear models (manifolds)

\[
\min_{\mathcal{P}} |d(y_i, y_j) - d(\mathcal{P}y_i, \mathcal{P}y_j)|
\]

distance between projected data points

nonlinear manifold
Isomap, LLE
nonlinear models (manifolds)

swiss roll

http://www.numerical-tours.com/matlab/shapes_7_isomap/
nonlinear models (manifolds)

Compute k-nearest neighbor graph

http://www.numerical-tours.com/matlab/shapes_7_isomap/
nonlinear models (manifolds)

Compute geodesic distances

http://www.numerical-tours.com/matlab/shapes_7_isomap/
nonlinear models (manifolds)

Compute leading eigenvectors

http://www.numerical-tours.com/matlab/shapes_7_isomap/
cluster model

\[ \min_{c_1, \ldots, c_k} \sum_{j=1}^{k} \sum_{i \in \Omega_j} \|y_i - c_j\|_2 \]

ith data point

clusters

k-means
cluster model

clusters $k$-means

$$\min_{c_1, \ldots, c_k} \sum_{j=1}^{k} \sum_{i \in \Omega_j} \| y_i - c_j \|_2$$

jth cluster center
**kmeans:**

1. Randomly initialize cluster centers
2. Assign each data point to its closest cluster center
3. Update cluster centers (mean of all assigned points)
4. Iterate steps 2-3 until convergence

\[
\begin{align*}
\min_{c_1, \ldots, c_k} & \sum_{j=1}^{k} \sum_{i \in \Omega_j} \|y_i - c_j\|_2 \\
\end{align*}
\]
union of subspaces

\[ \min_{A_i, \Omega_i} \sum \| Y_{\Omega_i} - A_i \|_F \quad \text{s.t.} \quad \text{rank}(A_i) \leq k_i \]

subset of data in ith subspace
union of subspaces

\[
\min_{A_i, \Omega_i} \sum \| Y_{\Omega_i} - A_i \|_F \quad \text{s.t.} \quad \text{rank}(A_i) \leq k_i
\]

low rank approx of ith cluster
### Sparse Subspace Clustering (SSC):

1. Compute the subspace affinity matrix ($C$)
2. Cluster the affinity matrix $C$
3. For each subspace cluster, run SVD and get low rank approximation

---

**Dyer, JMLR 2013**
“tangled” manifolds
autoencoders

\[
\min_f f(Y_{in}, Y_{out})
\]
linear autoencoder $\rightarrow$ PCA

$$Y \approx USV^T$$

$$\min_f f(Y_{in}, Y_{out})$$
application to movement decoding
movement decoding

Record neural activity

movement distribution

velocity (x)

velocity (y)
neural responses are low-d
neural responses are low-d
neural responses are low-d

challenge: signal and noise structure vary!

bad low-d representation
neural responses are low-d

solution: leverage distribution of known movements

bad low-d representation
distribution alignment approach

\[ \hat{V} = HY_p \]

Record neural activity

Spikes

Prior distribution

Dimensionality reduction

\[ \tilde{V} \]

Predicted movement

Projected neural activity

Dyer, Nature BME, 2017  
github.com/nerdslab/DAD
KL-divergence minimization

**Goal:** align neural activities with prior movement distribution

Dyer, Nature BME, 2017
KL-divergence minimization

\[ H^* = \arg \min_{H \in \mathbb{R}^{d \times 3}} KL(p \| q) \]

Estimate \( p \) from \( \tilde{V} \)
Estimate \( q \) from \( \hat{V} = H Y_p \)

Dyer, Nature BME, 2017
model selection

Dyer, Nature BME, 2017
decoding results

optimal linear map on test set

Dyer, Nature BME, 2017
decoding results

L2-regularized estimate

Dyer, Nature BME, 2017
decoding results

supervised method which leverages dynamics

Dyer, Nature BME, 2017
decoding results

Dyer, Nature BME, 2017
Dyer, Nature BME, 2017
increasing the population size
application to visual coding
visual coding

Allen Institute Brain Observatory

Photo Credit: Allen Institute Brain Observatory  
De Vries, bioRxiv, 2018
naturalistic stimuli are low-d
visualizing population dynamics

all 212 neurons

start

stop
cluster analysis - natural movies
visualizing population dynamics

subset of 71 neurons
summary

overview of low-dimensional models
- Linear subspace models (PCA, FA, NMF)
- Manifold models (Isomap, LLE)
- Clustering models (kmeans)
- Unions of subspaces (SSC)

example 1: **movement decoding**
- use movement priors to guide factorizations
- decode without supervised data

example 2: **visual coding**
- cluster neurons then factorize
- not all neurons are created equal
collaborators

**movement decoding**
Mohammad Gheshlaghi Azar (DeepMind)
Konrad Kording (UPenn)
Lee Miller (Northwestern)

**visual coding**
Saskia de Vries (Allen Institute)
code/data refs

Matlab Toolbox for dimensionality reduction
• https://lvdmaaten.github.io/drtoolbox/

Python Tutorials on PCA
• https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

MATLAB Tutorial on Isomap
• http://www.numerical-tours.com/matlab/shapes_7_isomap/

Distribution Alignment Decoding (DAD)
• https://github.com/KordingLab/DAD/tree/master/data/demo
Dimensionality reduction for neural data (Review)

Distribution Alignment Decoding (DAD)
• http://rdcu.be/Bafy

Brain Observatory Pipeline Paper
• https://www.biorxiv.org/content/early/2018/06/29/359513
thank you!

(web)
dyerlab.gatech.edu