Unsupervised Learning

What is it?
- no labels
- discover patterns in data
- clustering

Just a bunch of vectors, what to do with them?

What are our goals?
- understand data
- sanity check
- remove uninformative part of data (dimensionality reduction) (compression)
- learning features (and feature selection)
- visualization

Clustering

K-means: adds points to clusters

Representation: 3 centers, assignment (lossy compression)

Magnitude of displacement vector

New points are likely in the cluster

Only displacement, which is small
**Principle Component Analysis (PCA)**

- Dimensionality reduction method
  approximate data set in some dimension \( m \), \( m < d \)

PCA is a linear model

\[
V_n = W x_n
\]

\( W \in \mathbb{R}^{m \times d} \)

\( x \in \mathbb{R}^d \)

\( X \) : whole data set \( \mathbb{R}^{D \times N} \)

- one dimensional subspace (project)
  approximate representation

"maximize variance"
  of projected values
  equivalent to small l2 error / fitting

\( m = 1 \) (for now)

\( u \in \mathbb{R}^d \)

\( u^T u = 1 \)

Variance of projected data:

\[
= \frac{1}{N} \sum_{n=1}^{N} (u^T x_n)^2 \\
= u^T S u, \text{ where } S \text{ is covariance matrix } XX^T
\]

\[
\max_u u^T S u + \lambda (1 - u^T u)
\]

Lagrange multipliers: \( \lambda = u^T u \)

\[
S u = \lambda u = 0
\]

\[
S u = \lambda u
\]

Eigenvalue of covariance matrix of the data

- but which eigenvector?

Variance is now:

\[
u^T S u = u^T \lambda u = \lambda u^T u = \lambda
\]

- pick biggest eigenvalue

\( m > 1 \) ? repeat with remaining variance
Singular Value Decomposition (SVD)

\[
X = U \Sigma V^T
\]
eigenvalues in \( U \) get biggest
another way of doing eigenvalue decomposition

How to pick \( m \)?

\[
\text{example "compromise" point}
\]

"how much of the variance am I throwing away?"

Probabilistic PCA

\[
X_n = W z_n + \epsilon \sim N(0, \Sigma) \quad \text{latent space}
\]
\[
z_n \sim N(0, I) \quad \text{i.i.d.}
\]
\[
\epsilon \sim N(0, \sigma^2 I) \quad \text{i.i.d. (isotropic noise)}
\]

\[
p(x|z) \sim N(Wz, \sigma^2 I)
\]
\[
p(x) \sim N(\mu, \Sigma)
\]

\[
\mu = E[x] = E[Wz + \epsilon] = 0
\]
\[
\Sigma = \text{cov}[x] = E[(Wz + \epsilon)(Wz + \epsilon)^T]
\]
\[
= E[Wzz^TW^T] + E[\epsilon\epsilon^T]
\]
\[
= WW^T + \sigma^2 I
\]

MLE is PCA

\[
W \text{ only appears as } WW^T \quad \text{correction}
\]
\[
W = WR
\]
\[
\hat{W} \hat{W} = WRR^TW^T = WW^T
\]
could be considered undesirable, any orthogonal transformation on \( W \) gives the same model
Cocktail Party Problem
microphones listening to sum of S voices
decompose signal to S sources

**Blind Source Separation**
true as 1980s data

\[
Z = \begin{bmatrix}
\text{Person 1} \\
\text{Person 2} \\
\vdots
\end{bmatrix}
\]

inter distances, disambiguate people

\[
\begin{aligned}
X &= WZ \\
Y &= X \\
t &= T \times \text{time samples} \\
N_S &= T \times \text{of speakers} \\
N_M &= T \times \text{of microphones} \\
N &= T \times (N_M + T N_S)
\end{aligned}
\]

\[
\text{equations: } T N_M \quad \text{if } \quad T N_M > N
\]

if \( T N_M > N \) we can hopefully solve it

PCA: disaster! \( Z \) means S people
PCA could be multiplied by some orthogonal matrix
the problem has a "correct answer"

**Independent Component Analysis (ICA)**
- linear model (like PCA)
- not Gaussian prior on \( Z \)
- \( p(Z) = \prod_i p(Z_i) \)

**Factor Analysis**
- like probabilistic PCA, but instead of variance or II variance,
each element can have its own