Introduction to (Statistical) Machine Learning

Brown University CSCI1420 & ENGN2520
Prof. Erik Sudderth

Lecture for Nov. 21, 2013:
HMMs: Forward-Backward & EM Algorithms,
Principal Components Analysis (PCA)

Many figures courtesy Kevin Murphy’s textbook,
Machine Learning: A Probabilistic Perspective
Inference for HMMs

• Assume parameters defining the HMM are fixed and known: distributions of initial state, state transitions, observations
• Given observation sequence, want to estimate hidden states

Minimize sequence (word) error rate: \[ L(z, a) = \mathbb{I}(z \neq a) \]
\[
\hat{z} = \arg \max_z p(z | x) = \arg \max_z \left[ p(z_1) \prod_{t=2}^T p(z_t | z_{t-1}) \right] \cdot \left[ \prod_{t=1}^T p(x_t | z_t) \right]
\]

Minimize state (symbol) error rate: \[ L(z, a) = \sum_{t=1}^T \mathbb{I}(z_t \neq a_t) \]
\[
\hat{z}_t = \arg \max_{z_t} p(z_t | x) = \arg \max_{z_t} \sum_{z_1} \cdots \sum_{z_{t-1}} \sum_{z_{t+1}} \cdots \sum_{z_T} p(z, x)
\]

Problem: Naïve computation of either estimate requires \( O(K^T) \)
Forward Filtering for HMMs

\[
p(z, x) = p(z)p(x | z) = \left[ p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1}) \right] \cdot \left[ \prod_{t=1}^{T} p(x_t | z_t) \right]
\]

**Filtered state estimates:** \( \alpha_t(z_t) = p(z_t | x_t, x_{t-1}, \ldots, x_1) \)
- Directly useful for online inference or tracking with HMMs
- Building block towards finding posterior given all observations

**Initialization:** Easy from known HMM parameters

\[
\alpha_1(z_1) = p(z_1 | x_1) \propto p(z_1)p(x_1 | z_1)
\]

**Recursion:** Derivation will follow from Markov properties

\[
\alpha_t(z_t) \propto p(x_t | z_t) \sum_{z_{t-1}=1}^{K} p(z_t | z_{t-1}) \alpha_{t-1}(z_{t-1}) \quad \mathcal{O}(K^2)
\]

multiply be proportionality constant so sums to one
Forward Filtering for HMMs

\[ \alpha_t(z_t) = p(z_t \mid x_t, x_{t-1}, \ldots, x_1) \]

\[ \alpha_{t+1}(z_{t+1}) \propto p(x_{t+1} \mid z_{t+1}) \sum_{z_t=1}^{K} p(z_{t+1} \mid z_t)\alpha_t(z_t) \]

**Prediction Step:** Given current knowledge, what is next state?

\[ p(z_{t+1} \mid x_t, \ldots, x_1) = \sum_{z_t=1}^{K} p(z_{t+1} \mid z_t)\alpha_t(z_t) \]

**Update Step:** What does latest observation tell us about state?

\[ \alpha_{t+1}(z_{t+1}) = p(z_{t+1} \mid x_{t+1}, x_t, \ldots, x_1) \propto p(x_{t+1} \mid z_{t+1})p(z_{t+1} \mid x_t, \ldots, x_1) \]

**Key Markov Identities:** From generative structure of HMM,

\[ p(z_{t+1} \mid z_t, x_t, \ldots, x_1) = p(z_{t+1} \mid z_t) \quad p(x_{t+1} \mid z_{t+1}, x_t, \ldots, x_1) = p(x_{t+1} \mid z_{t+1}) \]
Forward-Backward for HMMs

**Forward Recursion:** Distribution of State Given Past Data

\[
\alpha_1(z_1) \propto p(z_1)p(x_1 | z_1) \quad \alpha_t(z_t) = p(z_t | x_t, x_{t-1}, \ldots, x_1)
\]

\[
\alpha_{t+1}(z_{t+1}) \propto p(x_{t+1} | z_{t+1}) \sum_{z_t=1}^K p(z_{t+1} | z_t) \alpha_t(z_t)
\]

**Backward Recursion:** Likelihood of Future Data Given State

\[
\beta_T(z_T) = 1 \quad \beta_t(z_t) \propto p(x_{t+1}, \ldots, x_T | z_t)
\]

\[
\beta_t(z_t) \propto \sum_{z_{t+1}=1}^K p(x_{t+1} | z_{t+1})p(z_{t+1} | z_t)\beta_{t+1}(z_{t+1})
\]

**Marginal:** Posterior distribution of state given all data

\[
p(z_t | x_1, \ldots, x_T) \propto \alpha_t(z_t)\beta_t(z_t)
\]
EM for Hidden Markov Models

- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden states
  - Dynamic programming to efficiently infer state marginals
- **M-Step:** Given posterior distributions, find likely parameters
  - Like training of mixture models and Markov chains
- **Iteration:** Alternate E-step & M-step until convergence

\[ z_1, \ldots, z_N \rightarrow \text{hidden discrete state sequence} \]
\[ \pi, \theta \rightarrow \text{parameters (state transition & emission dist.)} \]
E-Step: HMMs

\[ q^{(t)}(z) = p(z \mid x, \pi^{(t-1)}, \theta^{(t-1)}) \propto p(z \mid \pi^{(t-1)})p(x \mid z, \theta^{(t-1)}) \]

Mixture Models

\[ q^{(t)}(z) \propto \prod_{i=1}^{N} p(z_i \mid \pi^{(t-1)})p(x_i \mid z_i, \theta^{(t-1)}) \]

- Hidden states are \emph{conditionally independent} given parameters
- Naïve representation of full posterior has size \( O(KN) \)

HMMs

\[ q^{(t)}(z) \propto \prod_{i=1}^{N} p(z_i \mid \pi^{(t-1)}_{z_{i-1}})p(x_i \mid z_i, \theta^{(t-1)}) \]

- Hidden states have \emph{Markov dependence} given parameters
- Naïve representation of full posterior has size \( O(K^N) \)
- But, our \emph{forward-backward dynamic programming} can quickly find the marginals (at each time) of the posterior distribution
M-Step: HMMs

\[ \theta^{(t)} = \arg \max_{\theta} \mathcal{L}(q^{(t)}, \theta) = \arg \max_{\theta} \sum_z q(z) \ln p(x, z \mid \theta) \]

Initial state dist.

\[ \sum_{k=1}^{K} \mathbb{E} \left[ N_{k}^{1} \right] \log \pi_k + \sum_{j=1}^{K} \sum_{k=1}^{K} \mathbb{E} \left[ N_{jk} \right] \log A_{jk} \]

State transition dist.

\[ \hat{\pi}_k = \frac{\mathbb{E} \left[ N_{k}^{1} \right]}{N} \]
\[ \hat{A}_{jk} = \frac{\mathbb{E} \left[ N_{jk} \right]}{\sum_{k'} \mathbb{E} \left[ N_{jk'} \right]} \]

emissions via weighted moment matching

State emission dist. (observation likelihoods)

\[ p(z_t = k \mid x_i, \theta^{\text{old}}) \log p(x_i, t \mid \phi_k) \]

\[ \mathbb{E} \left[ N_{k}^{1} \right] = \sum_{i=1}^{N} p(z_{i1} = k \mid x_i, \theta^{\text{old}}) \]
\[ \mathbb{E} \left[ N_{jk} \right] = \sum_{i=1}^{N} \sum_{t=2}^{T_i} p(z_{i,t-1} = j, z_{i,t} = k \mid x_i, \theta^{\text{old}}) \]
\[ \mathbb{E} \left[ N_{j} \right] = \sum_{i=1}^{N} \sum_{t=1}^{T_i} p(z_{i,t} = j \mid x_i, \theta^{\text{old}}) \]

Need posterior marginal distributions of single states, and pairs of sequential states

\[ p(z_t \mid x) \]
\[ p(z_t, z_{t+1} \mid x) \]
<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Clustering</td>
</tr>
<tr>
<td>Classification or Categorization</td>
<td>Regression</td>
</tr>
<tr>
<td>Continuous</td>
<td>Dimensionality reduction</td>
</tr>
</tbody>
</table>

- **Goal:** Infer label/response $y$ given only features $x$
- **Classical:** Find latent variables $y$ good for *compression* of $x$
- **Probabilistic learning:** Estimate parameters of joint distribution $p(x,y)$ which *maximize marginal probability* $p(x)$
Dimensionality Reduction

PCA Objective: Compression

- Observed feature vectors: \( x_n \in \mathbb{R}^D, \quad n = 1, 2, \ldots, N \)
- Hidden manifold coordinates: \( z_n \in \mathbb{R}^M, \quad n = 1, 2, \ldots, N \)
- Hidden linear mapping: \( \tilde{x}_n = Wz_n + b \)

\[
J(z, W, b \mid x, M) = \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2 = \sum_{n=1}^{N} \| x_n - Wz_n - b \|^2
\]

- Unlike clustering objectives like K-means, we can find the **global optimum** of this objective efficiently:

\[
b = \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n
\]

**Construct \( W \) from the top eigenvectors of the sample covariance matrix** (the directions of largest variance)
Principal Components Analysis Example

**PCA Analysis of MNIST Images of the Digit 3**

- PCA models all translations of data equally well (by shifting \( b \))
- PCA models all rotations of data equally well (by rotating \( W \))
- Appropriate when modeling quantities over time, space, etc.

\[
J(z, W, b \mid x, M) = \sum_{n=1}^{N} ||x_n - \tilde{x}_n||^2 = \sum_{n=1}^{N} ||x_n - Wz_n - b||^2
\]
PCA Derivation: One-Dimension

- Observed feature vectors: \( x_n \in \mathbb{R}^D, \quad n = 1, 2, \ldots, N \)

- Hidden manifold coordinates: \( z_n \in \mathbb{R}, \quad n = 1, 2, \ldots, N \)

- Hidden linear mapping: \( \tilde{x}_n = wz_n \quad w \in \mathbb{R}^{D \times 1} \quad w^T w = 1 \)

Assume mean already subtracted from data (centered)

\[
J(z, w \mid x) = \frac{1}{N} \sum_{n=1}^{N} ||x_n - \tilde{x}_n||^2 = \frac{1}{N} \sum_{n=1}^{N} ||x_n - wz_n||^2 
\]

- Step 1: Optimal manifold coordinate is always projection
  \( \hat{z}_n = w^T x_n \)

- Step 2: Optimal mapping maximizes variance of projection

\[
J(\hat{z}, w \mid x) = C - \frac{1}{N} \sum_{n=1}^{N} (w^T x_n)(x_n^T w) = C - w^T \Sigma w \quad \Sigma = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^T 
\]
Gaussian Geometry

• Eigenvalues and eigenvectors:
  \[ \sum u_i = \lambda_i u_i, \quad i = 1, \ldots, d \]
  \[ \sum U = U \Lambda \quad \sum \in \mathbb{R}^{d \times d} \]
  • For a *symmetric* matrix:
    \[ \lambda_i \in \mathbb{R} \quad u_i^T u_i = 1 \quad u_i^T u_j = 0 \]
    \[ \Sigma = U \Lambda U^T = \sum_{i=1}^{d} \lambda_i u_i u_i^T \]
  • For a *positive semidefinite* matrix:
    \[ \lambda_i \geq 0 \]
  • For a *positive definite* matrix:
    \[ \lambda_i > 0 \]
    \[ \Sigma^{-1} = U \Lambda^{-1} U^T = \sum_{i=1}^{d} \frac{1}{\lambda_i} u_i u_i^T \]

\[
\begin{align*}
U &= [u_1, \ldots, u_d] \\
\Lambda &= \text{diag}(\lambda_1, \ldots, \lambda_d)
\end{align*}
\]

\[ \Delta^2 = (x - \mu)^T \Sigma^{-1} (x - \mu) \]

• Quadratic forms:
  \[ \Delta^2 = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i} \]
  \[ y_i = u_i^T (x - \mu) \]

*Projection of difference from mean onto eigenvector*
Maximizes Variance & Minimizes Error

C. Bishop, Pattern Recognition & Machine Learning
Principal Components Analysis (PCA)
PCA Optimal Solution

\[ J(z, W, b \mid x, M) = \sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2 = \sum_{n=1}^{N} \|x_n - W z_n - b\|^2 \]

\[ b = \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \]
\[ X = [x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_N - \bar{x}] \]

- Option A: Eigendecomposition of sample covariance matrix

\[ \Sigma = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{N} XX^T = U \Lambda U^T \]

Construct \( W \) from eigenvectors with \( M \) largest eigenvalues

- Option B: Singular value decomposition (SVD) of centered data

\[ X = U S V^T \]

Construct \( W \) from singular vectors with \( M \) largest singular values