Lecture 14
Principal Components Analysis
Generating data from mixture-of-Gaussians

series of stochastic selections

1. choose which mixture component (multinomial variable, distribution $\pi$)

2. sample from that Gaussian (parameters $m^k, \sigma^2_k$)

known as spherical (circularly symmetric) Gaussian: same variance along each dimension $i \in 1,..,N$

will also consider more general case: diagonal Gaussian with variance $\sigma^k_i$ along dimension $i$
Fitting a mixture-of-Gaussians

Assume that the parameters are not known, want to infer them from the data

Maximize $L$, the log-likelihood of the data [log because likelihood is product of probabilities of many points, can get quite small]

\[ L = \log P(\{x^t\}|\theta) = \sum_t \log P(x^t|\theta) \]

single Gaussian – solve for parameter $\theta$ by setting $\frac{\partial L}{\partial \theta} = 0$

for example, if $\sigma$ known, $\hat{m}_i = \sum_t x_i^t / T$

but in mixture-of-Gaussians, we don’t know which Gaussian generated a given point!
Hidden variables & mixtures

can think of the assignment of input to a cluster as the label/class $z$ of the input, which is hidden or missing

$$P(x|\theta) = \sum_k P(z = k|\theta)P(x|z = k, \theta_k) = \sum_k \pi_k P_k(x|\theta_k)$$

under this model, the responsibility of a mixture component for a data point is its posterior probability:

$$r_k(x) = P(z = k|x, \theta) = \frac{\pi_k P_k(x|\theta_k)}{\sum_{k'} \pi_{k'} P_{k'}(x|\theta_{k'})}$$

Now if do gradient descent on $L$, will find that:

$$\frac{\partial L}{\partial \theta_k} = \sum_t r_k^t \frac{\partial \log P_k(x^t|\theta_k)}{\partial \theta_k}$$

For example, for Gaussian $P_k(x|\theta_k)$:

$$\frac{\partial L}{\partial m_i^k} = -\sum_t r_k^t (x_i^t - m_i^k)/(\sigma_i^{(k)})^2$$
Expectation-Maximization algorithm

Rather than doing gradient descent to optimize the parameters, use a different approach — iterate:

- **E-step**: fill in value of $z_k^t$ (expected value of $z_k^t$ is $r_k^t$, the responsibility of $k$ for $t$)

- **M-step**: update parameters $\theta_k$ by maximizing $L$, assuming these are true values for $z_k^t$

It is useful to optimize objective (likelihood) when there is missing data

these two steps directly correspond to assignment and update steps in algorithm
**Adaptive mixture-of-Gaussians algorithm**

**Initialization:** Set $K$ cluster parameters $\{m^k, \sigma^2_k, \pi_k\}$ randomly

**Assignment:** Each data point $t$ (dimensionality $N$) given soft 'degree of assignment' to each cluster $k$, known as responsibilities:

$$r^t_k = \frac{\pi_k (2\pi \sigma^2_k)^{-N/2} \exp[-d(m^k, \mathbf{x}^t)/\sigma^2_k]}{\sum_{k'} \pi_{k'} (2\pi \sigma^2_{k'})^{-N/2} \exp[-d(m^{k'}, \mathbf{x}^t)/\sigma^2_{k'}]}$$

**Update:** Model parameters (means, widths, proportions) adjusted based on data points they are responsible for:

$$m^k = \frac{\sum_t r^t_k \mathbf{x}^t}{R^k}$$

$$\sigma^2_k = \frac{\sum_t r^t_k (\mathbf{x}^t - m^k)^2}{NR^k}$$

$$\pi_k = \frac{R_k}{\sum_k R^k}$$

where $R^k = \sum_t r^t_k$ is total responsibility of mean $k$

Repeat Assignment, Update until assignments stable
Principal Components Analysis

PCA is the most popular instance of second main class of unsupervised learning methods, *projection* methods

aim: find small number of “directions” in input space that explain correlations in input data; re-represent data by projecting along those directions

data is assumed to be continuous, linear relationship between data and learned representation
**PCA: Details**

assume data $x = (x_1, ..., x_N)$ is zero-mean, and the covariance matrix is

$$
\Sigma = \langle xx' \rangle = \frac{1}{C} \sum_{c=1}^{C} (x^c - \bar{x})(x^c - \bar{x})^T
$$

normalized eigenvectors $e_1, ..., e_N$ have corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$

form output (re-representation of input) $y$ by projecting input along eigenvectors – multiply by matrix $U = [e_1; e_2; ...; e_N]$

$$
y = Ux
$$

$y_1 = e_1 x$ is first **principal component** of the input; $(y_1, .., y_M)$ are the first $M$ principal components

components of $y$ are decorrelated from each other, in order of decreasing variance $\sigma^2_{y_i} = \langle y_i^2 \rangle = \lambda_i$
Review: Eigenvectors

an eigenvector e can be thought of as a special vector for a matrix \( W \): for that particular \( W \)

\[
We = \lambda e
\]  

(1)

where \( \lambda \) is a scalar (a number) called the eigenvalue that corresponds to \( e \)

a large eigenvalue means that the matrix \( W \) stretches that eigenvector, while eigenvectors with small eigenvalues are not so lucky (magnified)

an \( N \times N \) matrix \( W \) can have at most \( N \) eigenvectors (for now assume there are exactly \( N \))

these eigenvectors form a basis for the space, which means we can write any vector \( v \) in that space as a linear combination of the eigenvectors:

\[
v = \sum_{i=1}^{N} c_i \hat{e}_i
\]  

(2)

where we have normalized each eigenvector
**Optimality properties**

dimensionality reduction – find some linear projection of input $x$ that preserves information

PCA examines dataset, finds appropriate directions: *projects data along coordinate directions with largest variation* – assumed to be the most important directions

PCA is optimal linear dimensionality reduction method

- minimizes MSE of reconstructed data

- linear transform which preserves maximal information under uncorrelated equal-variance additive Gaussian noise model

aim: map vectors $x^c$ in $N$-dim space onto vectors $y^c$ in $M$-dim space, $M \ll N$, where $y^c = Wx^c$
Deriving optimality property

\[
x = \sum_{i=1}^{N} y_i w_i
\]

assume weights \( W = [w_1...w_N] \) are orthonormal \( (W^T W = I) \)

\[
y_i = w_i^T x
\]

dimensionality reduction – keep only \( M \ll N \) basis vectors

\[
\hat{x} = \sum_{i=1}^{M} y_i w_i + \sum_{i=M+1}^{N} b_i w_i
\]

\[
x^c - \hat{x}^c = \sum_{i=M+1}^{N} (y_i^c - b_i) w_i
\]
\[
\min \text{ MSE:} \\
E = \frac{1}{2} \sum_{c=1}^{C} ||x^c - \tilde{x}^c||^2 = \frac{1}{2} \sum_{c=1}^{C} \sum_{i=M+1}^{N} (y_i^c - b_i)^2 \\

b_i = \frac{1}{C} \sum_{c=1}^{C} y_i^c = w_i^T (\frac{1}{C} \sum_{c=1}^{C} x^c) = w_i^T \bar{x} \\

E = \frac{1}{2} \sum_{i=M+1}^{N} \sum_{c=1}^{C} [w_i^T (x^c - \bar{x})]^2 = \frac{1}{2} \sum_{i=M+1}^{N} w_i^T \Sigma w_i \\

\text{where } \Sigma \text{ is the covariance matrix} \\
\Sigma = \sum_{c=1}^{C} (x^c - \bar{x})(x^c - \bar{x})^T \\
\text{turns out that the vectors } w_i \text{ that minimize } E \text{ are the eigenvectors } e_i \text{ of the covariance matrix} \\
\Sigma e_i = \lambda_i e_i \]
choosing these weights \((w_i = e_i)\) makes

\[
E = \frac{1}{2} \sum_{i=M+1}^{N} \lambda_i
\]

which can be minimized by choosing the \(N - M\) smallest eigenvalues

For second property:

- information proportional to the variance of the output \(y\), under a Gaussian model

- yields similar expression—\(\sum_{i=1}^{M} w_i^T \Sigma w_i\)—which can be maximized by selecting the \(M\) largest eigenvalues
**PCA: Algorithm**

1. compute and subtract off the mean of the vectors $x^c$

2. calculate the covariance matrix $\Sigma$

3. find its eigenvectors and eigenvalues

4. retain the $M$ largest eigenvectors