

Lecture 14
Principal Components Analysis

Generating data from mixture-of-Gaussians

series of stochastic selections

1. choose which mixture component (multinomial variable, distribution π)
2. sample from that Gaussian (parameters \mathbf{m}^k, σ_k^2)

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

will also consider more general case: diagonal Gaussian with variance σ_i^k 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]

$$L = \log P(\{\mathbf{x}^t\}|\theta) = \sum_t \log P(\mathbf{x}^t|\theta)$$

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

for example, if σ 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

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

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

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

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

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

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

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

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 θ_k by maximizing L , assuming these are true values for z_k^t

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 $\{\mathbf{m}^k, \sigma_k^2, \pi_k\}$ randomly

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

$$r_k^t = \frac{\pi_k (2\pi\sigma_k^2)^{-N/2} \exp[-d(\mathbf{m}^k, \mathbf{x}^t)/\sigma_k^2]}{\sum_{k'} \pi_{k'} (2\pi\sigma_{k'}^2)^{-N/2} \exp[-d(\mathbf{m}^{k'}, \mathbf{x}^t)/\sigma_{k'}^2]}$$

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

$$\begin{aligned}\mathbf{m}^k &= \frac{\sum_t r_k^t \mathbf{x}^t}{R^k} \\ \sigma_k^2 &= \frac{\sum_t r_k^t (\mathbf{x}^t - \mathbf{m}^k)^2}{N R^k} \\ \pi_k &= \frac{R_k}{\sum_k R^k}\end{aligned}$$

where $R^k = \sum_t r_k^t$ 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 $\mathbf{x} = (x_1, \dots, x_N)$ is zero-mean, and the covariance matrix is

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

normalized eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_N$ have corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$

form output (re-representation of input) \mathbf{y} by projecting input along eigenvectors – multiply by matrix $U = [\mathbf{e}_1; \mathbf{e}_2; \dots; \mathbf{e}_N]$:

$$\mathbf{y} = U\mathbf{x}$$

$y_1 = \mathbf{e}_1\mathbf{x}$ is first **principal component** of the input; (y_1, \dots, y_M) are the first M principal components

components of \mathbf{y} are decorrelated from each other, in order of decreasing variance $\sigma_{y_i}^2 = \langle y_i^2 \rangle = \lambda_i$

Review: Eigenvectors

an *eigenvector* \mathbf{e} can be thought of as a special vector for a matrix W : for that particular W

$$W\mathbf{e} = \lambda\mathbf{e} \quad (1)$$

where λ is a scalar (a number) called the *eigenvalue* that corresponds to \mathbf{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 \mathbf{v} in that space as a linear combination of the eigenvectors:

$$\mathbf{v} = \sum_{i=1}^N c_i \hat{\mathbf{e}}_i \quad (2)$$

where we have normalized each eigenvector

Optimality properties

dimensionality reduction – find some linear projection of input \mathbf{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 \mathbf{x}^c in N -dim space onto vectors \mathbf{y}^c in M -dim space, $M \ll N$, where $\mathbf{y}^c = W\mathbf{x}^c$

Deriving optimality property

$$\mathbf{x} = \sum_{i=1}^N y_i \mathbf{w}_i$$

assume weights $W = [\mathbf{w}_1 \dots \mathbf{w}_N]$ are orthonormal ($W^T W = I$)

$$y_i = \mathbf{w}_i^T \mathbf{x}$$

dimensionality reduction – keep only $M \ll N$ basis vectors

$$\hat{\mathbf{x}} = \sum_{i=1}^M y_i \mathbf{w}_i + \sum_{i=M+1}^N b_i \mathbf{w}_i$$

$$\mathbf{x}^c - \hat{\mathbf{x}}^c = \sum_{i=M+1}^N (y_i^c - b_i) \mathbf{w}_i$$

min MSE:

$$E = \frac{1}{2} \sum_{c=1}^C \|\mathbf{x}^c - \hat{\mathbf{x}}^c\|^2 = \frac{1}{2} \sum_{c=1}^C \sum_{i=M+1}^N (y_i^c - b_i)^2$$

$$b_i = \frac{1}{C} \sum_{c=1}^C y_i^c = \mathbf{w}_i^T \left(\frac{1}{C} \sum_{c=1}^C \mathbf{x}^c \right) = \mathbf{w}_i^T \bar{\mathbf{x}}$$

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

where Σ is the covariance matrix

$$\Sigma = \sum_{c=1}^C (\mathbf{x}^c - \bar{\mathbf{x}})(\mathbf{x}^c - \bar{\mathbf{x}})^T$$

turns out that the vectors \mathbf{w}_i that minimize E are the eigenvectors \mathbf{e}_i of the covariance matrix

$$\Sigma \mathbf{e}_i = \lambda_i \mathbf{e}_i$$

choosing these weights ($\mathbf{w}_i = \mathbf{e}_i$) 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 \mathbf{y} , under a Gaussian model
- yields similar expression— $\sum_{i=1}^M \mathbf{w}_i^T \Sigma \mathbf{w}_i$ —which can be maximized by selecting the M largest eigenvalues

PCA: Algorithm

1. compute and subtract off the mean of the vectors \mathbf{x}^c
2. calculate the covariance matrix Σ
3. find its eigenvectors and eigenvalues
4. retain the M largest eigenvectors