Lecture 13: Unsupervised Learning I
Clustering

Unsupervised Learning

Supervised learning algorithms have clear goal: produce desired outputs for given inputs

Goal of unsupervised learning algorithms (no explicit feedback whether outputs of system are correct) less clear:

- reduce dimensionality
- find clusters
- model data density
- find hidden causes

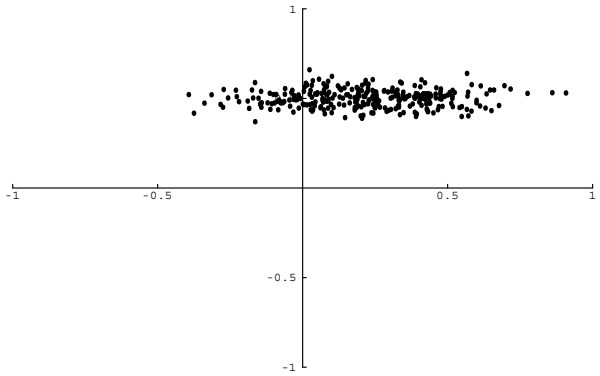
Key utility:

- compress data
- detect outliers
- facilitate other learning

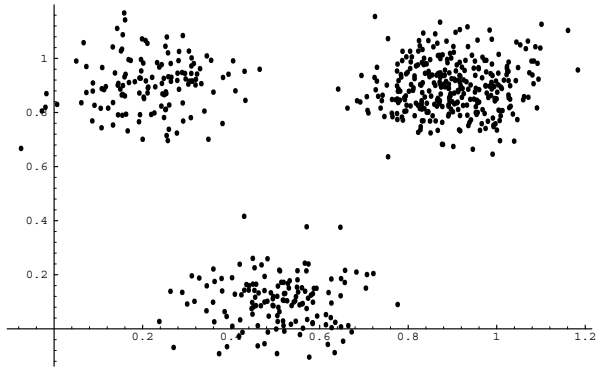
Major classes

Primary problems, approaches in unsupervised learning fall into different classes:

1. Dimensionality reduction: represent each input case using a small number of continuous variables (e.g., principal components analysis, factor analysis, independent component analysis)



2. Clustering: represent each input case using a prototype example (e.g., mixture models, competitive learning)



3. Density estimation: represent each example by the degree to which it fits a density estimated from the dataset

Clustering

grouping T objects into K clusters one of canonical problems in unsupervised learning

motivation:

1. predictive power: fill in unspecified attributes
2. lossy compression: convey approximate description
3. outlier detection: cluster misfits are outliers
4. neural models: learning processes in neural systems

K-means clustering

algorithm for putting T data points in N-dimensional input space into K clusters

each cluster parametrized by $\mathbf{m}^k$, its *mean*

data points: $\mathbf{x}^t, t = 1...T$

metric defines distances between points, e.g.,:

$$d(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2$$

iterative two-step algorithm:

1. assignment: each data point t assigned to nearest mean
2. update: means adjusted to match sample means of data points they are responsible for

K-means algorithm

Initialization: Set K means $\{\mathbf{m}^k\}$ to random values

Assignment: Each data point t assigned to nearest mean

$$\hat{k}^t = \arg\min_k \{d(\mathbf{m}^k, \mathbf{x}^t)\}$$

responsibilities:

$$r_k^t = 1 \iff \hat{k}^t = k$$

Update: Model parameters, means, are adjusted to match sample means of data points they are responsible for

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

$$R^k = \sum_t r_k^t$$

is total responsibility of mean k

Repeat assignment and update steps until assignments do not change

Questions about K-means

- why does update step set $\mathbf{m}^k$ to mean of assigned points?
- where did distance d come from?
- what if we used a different distance measure?
- how can we choose best distance?
- how to choose K
- how can we choose between alternative clusterings?

hard cases – unequal spreads, non-circular spreads, inbetween points

Soft K-means clustering

introduce additional parameter β

Initialization: Set K means $\{\mathbf{m}^k\}$ to random values

Assignment: Each data point t given soft 'degree of assignment' to each of means, based on responsibilities:

$$r_k^t = \frac{\exp[-\beta d(\mathbf{m}^k, \mathbf{x}^t)]}{\sum_{k'} \exp[-\beta d(\mathbf{m}^{k'}, \mathbf{x}^t)]}$$

Update: Model parameters, means, are adjusted to match sample means of data points they are responsible for

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

$$R^k = \sum_t r_k^t$$

is total responsibility of mean k

Repeat assignment and update steps until assignments do not change

Questions about soft K-means

- how to set β?
- what about problems with elongated clusters?
- clusters with unequal weight and width

Overview of Clustering Algorithms

1. hard K-means: adapt cluster means, binary responsibilities

2. soft K-means: soft (softmax) responsibilities

3. adaptive mixture-of-Gaussians: adapt cluster proportions & width(s)

can derive updates based on maximum-likelihood objective, for a mixture-of-Gaussians model

optimization algorithm: Expectation-Maximization

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

Mixture-of-Gaussians clustering

can derive algorithm as particular optimization method applied to fit parameters of mixture-of-Gaussian model to data

example: $N = 1$; 2 components in mixture:

$$\begin{aligned} P(x|m_1, \sigma_1, \pi_1, m_2, \sigma_2, \pi_2) &= \pi_1 (2\pi\sigma_1^2)^{-1/2} \exp[-\frac{(x-m_1)^2}{2\sigma_1^2}] \\ &+ \pi_2 (2\pi\sigma_1^2)^{-1/2} \exp[-\frac{(x-m_2)^2}{2\sigma_2^2}] \end{aligned}$$

more generally,

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k (2\pi\sigma_k^2)^{-N/2} \exp[-d(\mathbf{m}^k, \mathbf{x})/\sigma_k^2]$$

where $\theta = \{\pi_k, \mathbf{m}^k, \sigma_k^2\}$