Learning From Data
Lecture 19
A Peek At Unsupervised Learning

$k$-Means Clustering
Probability Density Estimation
Gaussian Mixture Models

M. Magdon-Ismail
CSCI 4100/6100
Recap: Radial Basis Functions

Nonparametric RBF

\[ g(x) = \sum_{n=1}^{N} \left( \frac{\alpha_n(x)}{\sum_{m=1}^{N} \alpha_m(x)} \right) \cdot y_n \]

\[ \alpha_n(x) = \phi \left( \frac{\| x - x_n \|}{r} \right) \] (bump on \( x \))

Parametric \( k \)-RBF-Network

\[ h(x) = w_0 + \sum_{j=1}^{k} w_j \cdot \phi \left( \frac{\| x - \mu_j \|}{r} \right) \]

\[ = w^T \Phi(x) \] (bump on \( \mu_j \))

linear model given \( \mu_j \)
choose \( \mu_j \) as centers of \( k \)-clusters of data

No Training

\( k = 4, r = \frac{1}{k} \)

\( k = 10, \) regularized
Unsupervised Learning

- Preprocessor to organize the data for supervised learning:
  - Organize data for faster nearest neighbor search
  - Determine centers for RBF bumps.

- Important to be able to organize the data to identify patterns.
  - Learn the patterns in data, e.g. the patterns in a language before getting into a supervised setting.
  - Amazon.com organizes books into categories
Clustering Digits

21-NN rule, 10 Classes

Symmetry

Average Intensity

10 Clustering of Data
A cluster is a collection of points $S$.

A $k$-clustering is a partition of the data into $k$ clusters $S_1, \ldots, S_k$.

$$\bigcup_{j=1}^{k} S_j = D$$

$$S_i \cap S_j = \emptyset \quad \text{for } i \neq j$$

Each cluster has a center $\mu_j$. 
How good is a clustering?

Points in a cluster should be similar (close to each other, and the center)

Error in cluster $j$:
$$E_j = \sum_{x_n \in S_j} \| x_n - \mu_j \|^2.$$ 

$k$-Means Clustering Error:

$$E_{\text{in}}(S_1, \ldots, S_k; \mu_1, \ldots, \mu_k) = \sum_{j=1}^{k} E_j$$

$$= \sum_{n=1}^{N} \| x_n - \mu(x_n) \|^2$$

$\mu(x_n)$ is the center of the cluster to which $x_n$ belongs.
**$k$-Means Clustering**

You get to pick $S_1, \ldots, S_k$ and $\mu_1, \ldots, \mu_k$ to minimize $E_{\text{in}}(S_1, \ldots, S_k; \mu_1, \ldots, \mu_k)$

If centers $\mu_j$ are known, picking the sets is easy:

Add to $S_j$ all points closest to $\mu_j$

If the clusters $S_j$ are known, picking the centers is easy:

Center $\mu_j$ is the centroid of cluster $S_j$

$$
\mu_j = \frac{1}{|S_j|} \sum_{x_n \in S_j} x_n
$$
Lloyd’s Algorithm for $k$-Means Clustering

\[ E_{\text{in}}(S_1, \ldots, S_k; \mu_1, \ldots, \mu_k) = \sum_{n=1}^{N} \| x_n - \mu(x_n) \|^2 \]

1. **Initialize** Pick well separated centers $\mu_j$.

2. **Update** $S_j$ to be all points closest $\mu_j$.
   \[ S_j \leftarrow \{ x_n : \| x_n - \mu_j \| \leq \| x_n - \mu_\ell \| \text{ for } \ell = 1, \ldots, k \} \]

3. **Update** $\mu_j$ to the centroid of $S_j$.
   \[ \mu_j \leftarrow \frac{1}{|S_j|} \sum_{x_n \in S_j} x_n \]

4. Repeat steps 2 and 3 until $E_{\text{in}}$ stops decreasing.
Lloyd’s Algorithm for $k$-Means Clustering

$$E_{in}(S_1, \ldots, S_k; \mu_1, \ldots, \mu_k) = \sum_{n=1}^{N} \| x_n - \mu(x_n) \|^2$$

1. **Initialize** Pick well separated centers $\mu_j$.

2. **Update** $S_j$ to be all points closest $\mu_j$.

   $$S_j \leftarrow \{ x_n : \| x_n - \mu_j \| \leq \| x_n - \mu_\ell \| \text{ for } \ell = 1, \ldots, k \}.$$ 

3. **Update** $\mu_j$ to the centroid of $S_j$.

   $$\mu_j \leftarrow \frac{1}{|S_j|} \sum_{x_n \in S_j} x_n$$

4. Repeat steps 2 and 3 until $E_{in}$ stops decreasing.
Lloyd’s Algorithm for $k$-Means Clustering

$$E_{in}(S_1, \ldots, S_k; \mu_1, \ldots, \mu_k) = \sum_{n=1}^{N} \| x_n - \mu(x_n) \|^2$$

1. **Initialize** Pick well separated centers $\mu_j$.

2. **Update** $S_j$ to be all points closest $\mu_j$.
   $$S_j \leftarrow \{ x_n : \| x_n - \mu_j \| \leq \| x_n - \mu_\ell \| \text{ for } \ell = 1, \ldots, k \}.$$  

3. **Update** $\mu_j$ to the centroid of $S_j$.
   $$\mu_j \leftarrow \frac{1}{|S_j|} \sum_{x_n \in S_j} x_n$$

4. Repeat steps 2 and 3 until $E_{in}$ stops decreasing.
Application to $k$-RBF-Network

10-center RBF-network

300-center RBF-network

Choosing $k$ - knowledge of problem (10 digits) or CV.
Probability Density Estimation

\[ P(x) \]

\( P(x) \) measures how likely it is to generate inputs similar to \( x \).

Estimating \( P(x) \) results in a ‘softer/finer’ representation than clustering.

Clusters are regions of high probability.
Parzen Windows – RBF density estimation

Basic idea: put a bump of ‘size’ (volume) $\frac{1}{N}$ on each data point.

$$P(x) = \frac{1}{N r^d} \sum_{i=1}^{N} \phi \left( \frac{\|x - x_i\|}{r} \right)$$

$$\phi(z) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}z^2}$$
Digits Data

RBF Density Estimate

Density Contours
The Gaussian Mixture Model (GMM)

Instead of $N$ bumps $\rightarrow k \ll N$ bumps.  
(Similar to nonparametric RBF $\rightarrow$ parametric $k$-RBF-network)

Instead of uniform spherical bumps $\rightarrow$ each bump has its own shape.

Bump centers:

\[ \mu_1, \ldots, \mu_k \]

Bump shapes:

\[ \Sigma_1, \ldots, \Sigma_k \]

Gaussian formula for the bump:

\[
\mathcal{N}(x; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x - \mu_j)^T\Sigma_j^{-1}(x - \mu_j)}.
\]
GMM Density Estimate

\[ \mathcal{N}(\mathbf{x}; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2} (\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j)}. \]

\[ \hat{P}(\mathbf{x}) = \sum_{j=1}^{k} w_j \mathcal{N}(\mathbf{x}; \mu_j, \Sigma_j) \]

(Sum of \(k\) weighted bumps).

\[ w_j > 0, \quad \sum_{j=1}^{k} w_j = 1 \]

You get to pick \(\{w_j, \mu_j, \Sigma_j\}_{j=1,...,k}\).
Maximize Likelihood Estimation

Pick \( \{w_j, \mu_j, \Sigma_j\}_{j=1,\ldots,k} \) to best explain the data.

Maximize the likelihood of the data given \( \{w_j, \mu_j, \Sigma_j\}_{j=1,\ldots,k} \)

(We saw this when we derived the cross entropy error for logistic regression)
Expectation-Maximization: The E-M Algorithm

A simple algorithm to get to the local minimum of the likelihood.

Partition variables into two sets. Given one-set, you can estimate the other
‘Bootstrap’ your way to a decent solution.

Lloyd’s algorithm for $k$-means is an example for ‘hard clustering’
Bump Memberships

Fraction of $\mathbf{x}_n$ belonging to bump $j$ (a ‘hidden variable’)

$\gamma_{nj}$
Bump Memberships

Fraction of \( x_n \) belonging to bump \( j \) (a ‘hidden variable’)

\[ \gamma_{nj} \]

\[
N_j = \sum_{n=1}^{N} \gamma_{nj} \quad \text{('number' of points in bump \( j \))}
\]

\[
w_j = \frac{N_j}{N} \quad \text{(probability bump \( j \))}
\]

\[
\mu_j = \frac{1}{N_j} \sum_{n=1}^{N} \gamma_{nj} x_n \quad \text{(centroid of bump \( j \))}
\]

\[
\Sigma_j = \frac{1}{N_j} \sum_{n=1}^{N} \gamma_{nj} x_n x_n^T - \mu_j \mu_j^T \quad \text{(covariance matrix of bump \( j \))}
\]
Re-Estimating Bump Memberships

\[ \gamma_{nj} = \frac{w_j \mathcal{N}(x_n; \mu_j, \Sigma_j)}{\sum_{\ell=1}^{k} w_\ell \mathcal{N}(x_n; \mu_\ell, \Sigma_\ell)} \]

\( \gamma_{nj} \) is the probability that \( x_n \) came from bump \( j \)

- probability of bump \( j \): \( w_j \)
- probability density for \( x_n \) given bump \( j \): \( \mathcal{N}(x_n; \mu_j, \Sigma_j) \)
E-M Algorithm for GMMs:
1. Start with estimates for the bump membership $\gamma_{nj}$.
2. Estimate $w_j, \mu_j, \Sigma_j$ given the bump memberships.
3. Update the bump memberships given $w_j, \mu_j, \Sigma_j$;
4. Iterate to step 2 until convergence.
GMM on Digits Data

10-center GMM

Density Contours