Model-based Clustering by Probabilistic Self-Organizing Maps

Shih-Sian Cheng (鄭士賢)

Outline

- Overview
  - Model-based clustering
  - Kohonen’s self-organizing map (SOM)
  - Motivation of probabilistic self-organizing map (PbSOM)

- Review for Model-Based Clustering
  - Maximum mixture likelihood and EM algorithm
  - Maximum classification likelihood and classification EM (CEM) algorithm

- PbSOM and Its Learning Algorithms
Model-Based Clustering

- Model-based clustering is achieved by learning a (Gaussian) mixture model

- We usually apply EM (expectation maximization) type algorithm to learn a GMM
Model-Based Clustering

- Model-based clustering is achieved by learning a (Gaussian) mixture model

We usually apply EM (expectation maximization) type algorithm to learn a GMM.
Kohonen’s Self-Organizing Map (SOM)

High-dimensional data space

Low-dimensional (<=3) network

Data clustering

After learning

Random initial mapping

Data clustering

Topology-preserving & data visualization
Kohonen’s Sequential Learning Algorithm

High-dimensional data space

Low-dimensional (<=3) network

X

r1 r2 r3

r4 r5 r6

□ : Data sample
○ : Network neuron
☆ : Reference vector
Kohonen’s Sequential Learning Algorithm

High-dimensional data space

Low-dimensional (<=3) network

- $X$: Data sample
- $\circ$: Network neuron
- $\star$: Reference vector
- $h_{kl}$: The given neighborhood function

August 31, 2009
Kohonen’s Sequential Learning Algorithm

Learning with lateral interactions between neurons

Topology preserved, but data clustering NOT finished
Kohonen’s Sequential Learning Algorithm

When zero neighborhood Kohonen’s algorithm → Sequential K-means clustering

Topology preserved & data clustering finished

$h_{kl} = \delta_{kl} r_4 r_5 r_6$

$k = l, \delta_{kl} = 0; \text{otherwise, } \delta_{kl} = 1$
Issues of Model-Based Clustering

- Conventional model-based clustering cannot preserve the topological relationships among clusters after the clustering procedure.

- How to initialize the GMM for a EM-type algorithm?

  Initialization is critical to performance.

- How many components (clusters)?

  We won’t address this issue in this work.
Issues of Kohonen’s SOM

- **On the perspective of learning**: its learning algorithm is heuristic and lacks an objective (cost) function.

- **On the perspective of clustering**: the clustering of Kohonen’s SOM is Euclidean distance-based K-means clustering, which is a special case of model-based clustering.

For the case that all the covariance matrices are identical and of small variance (i.e., $\Sigma = \lambda \mathbf{I}$, $\lambda \to 0$)

EM for GMM $\equiv$ K-means clustering
Motivation of Probabilistic SOM (PbSOM)

- We want a model (algorithm) that has the properties
  - Model-based clustering
  - Topology-preserving (data visualization)
  - Learning process based on optimization, rather than heuristic rules as Kohonen’s algorithm

Unify model-based clustering and SOM

PbSOM

- Data sample
- Network neuron
- Multivariate Gaussian distribution

After EM-type learning
Outline

- Overview
  - Model-based clustering
  - Kohonen’s self-organizing map (SOM)
  - Motivation of probabilistic self-organizing map (PbSOM)

- Review of Model-Based Clustering
  - Maximum mixture likelihood and EM algorithm
  - Maximum classification likelihood and classification EM (CEM) algorithm

- PbSOM and Its Learning Algorithms
Gaussian Mixture Model (GMM)

\[ p(x; \Theta_G) = \sum_{k=1}^{G} w(k) r_k(x; \theta_k) \]

Gaussian likelihood

\[ r_k(x; \theta_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \times \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right) \]

\[ \sum_{k=1}^{G} w(k) = 1 \quad \text{s. t.} \quad 0 \leq w(k) \leq 1 \]
Maximum Mixture Likelihood and EM Algorithm

• Data set: \( X = \{x_1, x_2, \ldots, x_N\} \)

• The GMM: \( p(x; \Theta_G) = \sum_{k=1}^{G} w(k) r_k(x; \theta_k) \) where \( \theta_k = \{\mu_k, \Sigma_k\} \)

• Learning the model ⇔ Maximizing the following log mixture likelihood function

\[
L(\Theta_G; X) = \log p(X; \Theta_G) = \log \prod_{i=1}^{N} p(x_i; \Theta_G) = \sum_{i=1}^{N} \log \left( \sum_{k=1}^{G} w(k) r_k(x_i; \theta_k) \right)
\]

Assume \( \{x_1, x_2, \ldots, x_N\} \) are i.i.d.

• Clustering data after model learning:

For \( i=1 \) to \( N \)

\( x_i \in \tilde{P}_j \) if \( j = \arg \max_k p(k|x_i; \hat{\Theta}_G) \)

Then, we obtain the clusters

\( \tilde{P} = \{\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_G\} \)
• Analytic solution for the log mixture likelihood function is currently unavailable

• The expectation-maximization (EM) algorithm is usually employed

Given the initial parameter set $\Theta_G^{(t)}$

Repeat

**E-step:**

for $i=1$ to $N$ and $k=1$ to $G$

$$\gamma_{ki}^{(t)} \equiv p(k \mid x_i; \Theta_G^{(t)}) = \frac{w(k^{(t)} r_k(x_i; \theta_k^{(t)})}{\sum_{j=1}^{G} w(j^{(t)} r_j(x_i; \theta_j^{(t)})}$$

Create the so-called auxiliary function: $Q(\Theta_G; \Theta_G^{(t)}) = \sum_{i=1}^{N} \sum_{k=1}^{G} \gamma_{ki}^{(t)} p(x_i, k; \Theta_G)$

**M-step:**

Setting the derivative of $Q(\Theta_G; \Theta_G^{(t)})$ w.r.t the parameters to zero, we obtain

$$w(k^{(t+1)} = (\sum_{i=1}^{N} \gamma_{ki}^{(t)}) / N$$

$$\mu_k^{(t+1)} = ((\sum_{i=1}^{N} \gamma_{ki}^{(t)} x_i) / (\sum_{i=1}^{N} \gamma_{ki}^{(t)}))$$

$$\Sigma_k^{(t+1)} = [(\sum_{i=1}^{N} \gamma_{ki}^{(t)} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T] / (\sum_{i=1}^{N} \gamma_{ki}^{(t)})$$

Until the convergence criterion is met

August 31, 2009
Maximum Classification Likelihood and Classification EM (CEM) Algorithm

• Learning the model \(\Leftrightarrow\) Maximize the following \textit{classification} log-likelihood function

\[
C(P, \Theta_G; X) = \sum_{k=1}^{G} \sum_{x_i \in P_k} \log w(k) r_k(x_i; \theta_k)
\]

\(x_i\) is the data point, \(w(k)\) is the weight, \(r_k(x_i; \theta_k)\) is the classification log-likelihood function.

\(\rightarrow\) Objective function of a K-means type algorithm

✓ The prototype are Gaussians, rather than reference vectors
✓ With mixture weights

• Clustering data after model learning:

For \(i=1\) to \(N\)

\[x_i \in P_j \text{ if } j = \arg \max_k p(k|x_i; \hat{\Theta}_G)\]

Then, we obtain the clusters

\(\hat{P} = \{\hat{P}_1, \hat{P}_2, ..., \hat{P}_G\}\)
• Analytic solution for the classification log-likelihood function is unavailable

• The **classification EM (CEM) algorithm** is usually employed

Given the initial parameter set $\Theta_G^{(0)}$

Repeat

**E-step:**

for $i=1$ to $N$ and $k=1$ to $G$

$$
\gamma_{ki}^{(t)} \equiv p(k \mid x_i ; \Theta_G^{(t)}) = \frac{w(k)^{(t)} r_{k}^{(t)}(x_i ; \theta_{k}^{(t)})}{\sum_{j=1}^{G} w(j)^{(t)} r_{j}^{(t)}(x_i ; \theta_{j}^{(t)})}
$$

**C-step:**

for $i=1$ to $N$ and $k=1$ to $G$

if $j = \arg\max_k \gamma_{ki}^{(t)}$, $x_i \in P_{G}^{(t)}$

**M-step:**

Setting the derivative of $C(\Theta_G ; X, \Lambda) = \sum_{k=1}^{G} \sum_{x_i \in P_k^{(t)}} w(k) \log r_{k}^{(t)}(x_i ; \theta_k)$ w.r.t the parameters to zero, we obtain

$$
w(k)^{(t+1)} = \left| P_k^{(t)} \right| / N,
$$

$$
\mu_{k}^{(t+1)} = ( \sum_{x_i \in P_k^{(t)}} x_i ) / \left| P_k^{(t)} \right|, \quad \Sigma_{k}^{(t+1)} = \left[ \sum_{x_i \in P_k^{(t)}} (x_i - \mu_{k}^{(t+1)})(x_i - \mu_{k}^{(t+1)})^{T} \right] / \left| P_k^{(t)} \right|,
$$

Until the convergence criterion is met
Comparison of EM for GMM and CEM for GMM

(a) EM for GMM

(b) CEM for GMM
Outline

- Overview
  - Model-based Clustering
  - Kohonen’s self-organizing map (SOM)
  - Motivation of probabilistic self-organizing map (PbSOM)

- Review for Model-Based Clustering
  - Maximum mixture likelihood and EM algorithm
  - Maximum classification likelihood and classification EM (CEM) algorithm

- PbSOM and Its Learning Algorithms
Probabilistic Self-Organizing Map (PbSOM)

- The architecture

PbSOM

- Data sample
- Network neuron
- Multivariate Gaussian distribution

After learning
Motivation of PbSOM’s Learning Algorithm

- The conventional model-based clustering
  ⇔ Learning a \((Gaussian)\) mixture model with \(EM\)-type algorithms

\[
p(x; \Theta_G) = \sum_{k=1}^{G} w(k) r_k(x; \theta_k)
\]

Gaussian mixture model

\(x\)

\(\Sigma\)

\(w(1)\)

\(w(2)\)

\(\ldots\)

\(w(G)\)

\(r_1(x; \theta_1)\)

\(r_2(x; \theta_2)\)

\(\ldots\)

\(r_G(x; \theta_G)\)

mixture of Gaussian likelihoods
**Coupling-Likelihood Mixture Model (CLMM)**

- Learning CLMM ⇔ Model-based clustering & topology-preserving

Coupling-likelihood involves lateral interactions between neurons

![Diagram](image)

\[ p_s(x | k; \Theta_G, h) = r(x; \theta_k) h_{kk} \prod_{l=k}^{G} r(x; \theta_l) h_{kl} = \prod_{l=1}^{G} r(x; \theta_l) h_{kl} = \exp\left(\sum_{l=1}^{G} h_{kl} \log r(x; \theta_l)\right) \]

Neighbor likelihoods are jointly considered

**Coupling-likelihood mixture model (CLMM)**

August 31, 2009
Coupling-Likelihood Mixture Model (CLMM)

- Learning CLMM ⇔ Model-based clustering & topology-preserving

We use equal mixture weights in CLMM to take account of the topological order learning induced by the neurons with equal prior importance.

$$p_s(x; \Theta, h) = \sum_{k=1}^{G} \frac{1}{G} p_s(x | k; \Theta, h)$$

When $h_{kl} \rightarrow \delta_{kl}$,

CLMM $\Rightarrow$ GMM with equal mixture weights
Since CLMM is a mixture model, we can apply EM to learn CLMM; here we call it the SOEM algorithm:

Given the initial parameter set \( \Theta_G^{(t)} \)

Repeat

**E-step:**

for \( i=1 \) to \( N \) and \( k=1 \) to \( G \)

\[
\gamma_{k|i}^{(t)} \equiv p_s(k | x_i; \Theta_G^{(t)}, h) = \frac{p_s(x_i | k; \Theta_G^{(t)}, h)}{\sum_{j=1}^{G} p_s(x_i | j; \Theta_G^{(t)}, h)} = \frac{\exp(\sum_{l=1}^{G} h_{kl} \log r(x_i; \theta_l))}{\sum_{j=1}^{G} \exp(\sum_{l=1}^{G} h_{jl} \log r(x_i; \theta_j))}
\]

Create the so-called auxiliary function: \( Q_s(\Theta_G; \Theta_G^{(t)}) = \sum_{i=1}^{N} \sum_{k=1}^{G} \gamma_{k|i}^{(t)} p_s(x_i, k; \Theta_G, h) \)

**M-step:**

Setting the derivative of \( Q_s(\Theta_G; \Theta_G^{(t)}) \) w.r.t the parameters to zero, we obtain

\[
\mu_i^{(t+1)} = \frac{\sum_{i=1}^{N} \sum_{k=1}^{G} \gamma_{k|i}^{(t)} h_{kl} x_i}{\sum_{i=1}^{N} \sum_{k=1}^{G} \gamma_{k|i}^{(t)} h_{kl}}
\]

\[
\Sigma_i^{(t+1)} = \frac{\sum_{i=1}^{N} \sum_{k=1}^{G} \gamma_{k|i}^{(t)} h_{kl} (x_i - \mu_i^{(t+1)})(x_i - \mu_i^{(t+1)})^T}{\sum_{k=1}^{G} \gamma_{k|i}^{(t)} h_{kl}}
\]

Until the convergence criterion is met
The SOCEM Algorithm for Learning CLMM

We can also apply CEM to learn CLMM, we call it SOCEM.

Given the initial parameter set $\Theta^{(t)}_G$

Repeat

**E-step:**

for $i=1$ to $N$ and $k=1$ to $G$

$$\gamma_{k|i}^{(t)} \equiv p_s(k \mid x_i; \Theta^{(t)}_G, h) = \frac{p_s(x_i \mid k; \Theta^{(t)}_G, h)}{\sum_{j=1}^{G} p_s(x_i \mid j; \Theta^{(t)}_G, h)} = \frac{\exp(\sum_{l=1}^{G} h_{kl} \log r(x_i; \theta_i))}{\sum_{j=1}^{G} \exp(\sum_{l=1}^{G} h_{jl} \log r(x_i; \theta_j))}$$

**C-step:**

for $i=1$ to $N$ and $k=1$ to $G$

if $j = \arg\max_k \gamma_{k|i}^{(t)}$, $x_i \in \tilde{P}_{j}^{(t)}$

**M-step:**

Setting the derivative of $C_s(\Theta_G; \tilde{P}, X, h) = \sum_{k=1}^{G} \sum_{x_i \in P_k} \log p_s(x_i \mid k; \Theta_G, h)$

w.r.t the parameters to zero, we obtain

$$\mu_{l}^{(t+1)} = \left( \sum_{k=1}^{G} \sum_{x_i \in P_k} h_{kl} x_i \right) / \sum_{k=1}^{G} |P_k| h_{kl},$$

$$\Sigma_{l}^{(t+1)} = \left( \sum_{k=1}^{G} \sum_{x_i \in P_k} h_{kl} (x_i - \mu_{l}^{(t+1)})(x_i - \mu_{l}^{(t+1)})^T \right) / \left( \sum_{k=1}^{G} |P_k| h_{kl} \right)$$

Until the convergence criterion is met

When $h_{kl} \to \delta_{kl}$,

SOCEM $\Rightarrow$ CEM for GMM
Comparison of SOEM and SOCEM

(a) SOEM

(b) SOCEM

Winner-take-all

Probabilistic assignment

\[ r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4 \rightarrow \ldots \rightarrow r_G \]

\[ X_i \rightarrow \gamma_{ij} \]

\[ (a) \ SOEM \]

\[ (b) \ SOCEM \]
Comparison of SOCEM and Kohonen’s Algorithm

- The same property: winner-take-all strategy

- The differences
  1) SOCEM considers the neighborhood information when selecting the winning neuron, but Kohonen’s algorithm does not

SOCEM’s winner selection

E-step:
for $i=1$ to $N$ and $k=1$ to $G$

$$\gamma_{k,i}^{(t)} \equiv \frac{\exp(\sum_{l=1}^{G} h_{kl} \log r(x_i; \theta_l))}{\sum_{j=1}^{G} \exp(\sum_{l=1}^{G} h_{jl} \log r(x_i; \theta_l))}$$

C-step:
for $i=1$ to $N$ and $k=1$ to $G$

if $j = \arg \max_k \gamma_{k,i}^{(t)}$, $x_i \in \mathcal{P}_j^{(t)}$

2) SOCEM extends the reference vectors in Kohonen’s algorithm to multivariate Gaussians

for each $x_i$ : Kohonen’s winner selection

find $m_k$ such that $\|x - m_k\| = \min_j \{\|x - m_j\|\}$.

$m_k^{new} = m_k + \alpha(t)[x_i - m_k]$

$m_j^{new} = m_j$ for $j \neq c$
Experiments: Simulations on 2D data

- The network structure: 5 X 5 lattice

- The neighborhood function: \( h_{kl} = \exp\left(-\frac{||r_k - r_l||}{2\sigma^2}\right) \)

- Random initial mapping between neurons and Gaussian components:

Larger \( \sigma \) value, larger neighborhood

\( r_k \) \( r_l \)
Simulations Using SOEM

Initialization of SOEM?
→ Start with a large neighborhood, rand. ini. is fine

EM for GMM
Simulations Using SOCEM

(a) Rand ini.  (b) $\sigma=0.6$  (c) $\sigma=0.45$

(d) $\sigma=0.3$  (e) $\sigma=0.15$  (f) $\sigma=0$ ($h_{kl} \rightarrow \delta_{kl}$)

Initialization of SOCEM?
$\rightarrow$ Start with a large neighborhood, rand. ini. is fine

CEM for GMM
Comparison of SOEM and SOCEM

SOEM result

\[ \sigma = 0 \quad (h_{kl} \rightarrow \delta_{kl}) \]

EM for GMM

SOCEM result

\[ \sigma = 0 \quad (h_{kl} \rightarrow \delta_{kl}) \]

CEM for GMM
Experiments: High-Dimensional Data

- The network structure: 7 X 7 lattice
- The neighborhood function: \( h_{kl} = \exp\left(-\frac{\|r_k - r_l\|^2}{2\sigma^2}\right) \)
- Data set: **test set of image segmentation** database from UCI (which consists of seven classes, namely brickface: B, sky: S, foliage: F, cement: C, window: W, path: P, and grass: G, each with 300 19-dimensional data samples)

- Use **diagonal** covariance Gaussians in order to avoid singular covariance matrices

Data projection on the network obtained by random initial mapping
Conclusion

- Probabilistic self-organizing maps (PbSOMs) are proposed for model-based clustering and data visualization.

- PbSOM learning algorithms, SOEM and SOCEM, are derived by learning a coupling-likelihood mixture model (CLMM) based on the maximum likelihood criterion.

- SOEM is an EM algorithm for learning CLMM.

- SOCEM is a classification EM (CEM) algorithm for learning CLMM.
Future Research Directions

- Automatic clustering (abstraction) of the network

- How to solve the singularity problem of full covariance Gaussians in SOEM and SOCEM

- Apply PbSOM to Kohonen SOM’s applications

August 31, 2009
WEBSOM - Self-Organizing Maps for Internet Exploration (http://websom.hut.fi/websom/)

- Over million documents from over 80 Usenet newsgroups
- Automatic labeling of the documents
- More bright color, higher document density

Graphical user interface for exploring and searching vast amounts of documents

In this example

- Over million documents from over 80 Usenet newsgroups

Instructions

Explanation of the symbols on the map

- acorn - comp.sys.acorn.hardware
- amiga - comp.sys.amiga.hardware
- books - comp.rec.arts.books
- cdrom - comp.publish.cdrom.hardware
- compilers - comp.compilers
- fuzzy - comp.ai.fuzzy
- genetic - comp.ai.genetic
- hp - comp.sys.hp.hardware
- humor - rec.humor
- lang.eiffel - comp.lang.eiffel
- lang.ml - comp.lang.ml
- linux - comp.os.linux.hardware
- lisp - comp.lang.lisp
- lisp.mcl - comp.lang.lisp.mcl
- mac - comp.sys.mac.hardware
- mac.storage - comp.sys.mac.hardware.storage
- movies - movies
- music - music
- nt - comp.os.ms-windows.nt.setup.hardware
- pc.cdrom - comp.sys.ibm.pc.hardware.cd-rom
- pc.chips - comp.sys.ibm.pc.hardware.chips
- pc.com - comp.sys.ibm.pc.hardware.com
- pc.storage - comp.sys.ibm.pc.hardware.storage
- pc.video - comp.sys.ibm.pc.hardware.video
- philosophy - philosophy
- plant - bioc.biology.plant
- prolog - comp.lang.prolog
- sci.lang - sci.lang
- smalltalk - comp.lang.smalltalk
- speech - comp.speech
- sun - comp.sys.sun.hardware

Full list of newsgroups
Click arrows to move to neighboring areas on the map, and to move up to the overall view.

Click any area on the map to get a zoomed view!

Instructions
Click on one of the white nodes to move to neighboring areas on the map, and to move up to the overall view.

Click any white dot to enter the node.

Instructions

**Explanation of the symbols on the map**

- blues - rec.music.bluenote
- books - rec.arts.books
- classical - rec.music.classical
- lang.dylan - comp.lang.dylan
- lang.eiffel - comp.lang.eiffel
- lisp - comp.lang.lisp
- movies - rec.arts.movies.current-films
- music - music
- philosophy - comp.ai.philosophy
- sci.lang - sci.lang
- smalltalk - comp.lang.smalltalk
WEBSOM node 2e,226

Click arrows
to move to neighboring nodes on the map.

Instructions

rec.arts.books


comp.ai.philosophy

**Re: On Going Beyond The Information Given & 'Cognition'** 🟦 PHIL@daffodif.demon.co.uk, Thu, 17 Aug 1995, Lines: 85.
Question?