Mixture Models for Learning Low-dimensional Roles in High-dimensional Data

Manas Somaiya[1] Christopher Jermaine[2] Sanjay Ranka[1]

[1]CISE Department
University of Florida

[2]CS Department
Rice University

manas@acm.org
http://www.cise.ufl.edu/~mhs/kdd2010Talk.pdf

July 27, 2010

Outline

Mixture models

Motivating examples

POWER model

Learning the POWER model

Experimental evaluation

Related work

What are mixture models?

In statistics, a probability mixture model is a probability distribution that is a convex combination of other probability distributions.

Suppose that the random variable X is a mixture of n component random variables $Y_1 \cdots Y_n$. Then,

$$fX(x) = \sum_{i=1}^{n} a_i \cdot fY_i(x)$$

for some mixture proportions $0 < a_i < 1$ such that $\sum_i a_i = 1$.

For example, the distribution of the height of students in a class can be thought of as a mixture of the distribution of the height of female students and the distribution of the height of the male students.

Using mixture models

- Using a mixture of random variables to model data is very common technique in data mining, machine learning, and statistics
- Given a set of k components $C = \{C_1, C_2, \cdots, C_k\}$, it is assumed that each data point was produced by first randomly selecting a component C_i from C, and generating attributes according to the distribution specified by C_i
- Classic application is the *Gaussian Mixture Model*. Data is seen as being produced by taking a set of samples from a mixture of k Gaussians
- Often possible to accurately model even complex and multi-modal data using very simple components

Shortcomings of classical mixture model

- A data point is produced by a single component
- A component must provide a generative distribution for all attributes of the data space
- Conflicts with the underlying reality of many datasets
  - $\longrightarrow$ Multiple generative components may influence a data point
  - $\longrightarrow$ A generative component may have influence over only a subset of data attributes
  - $\longrightarrow$ A generative component may have varying influence over data attributes

Example scenario

Real life situation: Retail store, items, customers
Goal: Build an informative model for buying patterns of different classes of customers

With the classical mixture model:

- Each customer belongs to only one class
- Each customer class should attempt to completely describe all the buying patterns of its members
- Highly unrealistic considering the diversity of customers and items for sale

Example scenario · · ·

More accurate and natural to explain the behavior of each customer as resulting from influence of several customer classes:

- Each customer class may influence purchase of an item to a varying degree:
  - → For example, a customer is an *action-movies-fan*, *horror-movies-fan*, and *parent*
  - → One of the items for sale is the animated movie *Teenage Mutant Ninja Turtles*
  - → Being a *parent* will have a stronger influence on purchase of this item than the other two classes.
  - → Being a *action-movies-fan* will have a stronger influence than being a *horror-movies-fan*
- Each data point can be modeled with high precision
- However allows learning very general classes such as *parent* that are important, and yet cannot describe any data point completely

Formal definition of the model

POWER (PrObabilstic Weighted Ensemble of Roles) model

The proposed model consists of a mixture of k components $C = \{C_1, C_2, \cdots, C_k\}$. Associated with each component C_i is:

- An appearance probability α_i
- A d-dimensional parameter vector Θ_i
  - $\rightarrow$ d is the number of data attributes
  - $\rightarrow$ Θ_{ij} parameterizes the probability density function f_j corresponding to the j^{th} data attribute A_j
  - $\rightarrow$ For example, if f_j is a normal random variable, then Θ_{ij} is the mean μ_{ij} and std dev σ_{ij}
- A vector of positive real numbers "parameter weights" W_i
  - $\rightarrow$ w_{ij} specifies the strength of influence of component C_i over attribute A_j
  - $\rightarrow$ $\sum_j w_{ij} = 1$

Data generation process

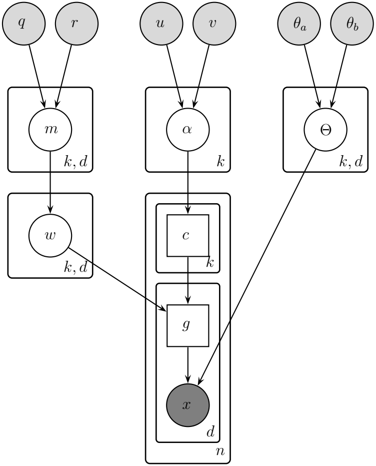
Each data point is generated by the following three step process:

- First, one or more of the k components are marked as "active" by performing a Bernoulli trial with their appearance probabilities
- Second, for each attribute a "dominant" component is selected by performing a weighted multinomial trial (using the parameter weights) amongst active components
- Finally, each data attribute is generated using its parameterized density function by borrowing the parameters from its dominant component

Data generation process · · ·

- Issue – because of Bernoulli trials, non-zero probability of selecting no components
- Solution – make one of the components a special *default* component that is always selected
  - $\longrightarrow$ Set default's appearance probability $\alpha = 1$
  - $\longrightarrow$ Acts as a "catch-all" or background distribution
  - $\longrightarrow$ Want the default component to actually generate a data attribute only when no other component can
  - $\longrightarrow$ Set default's all parameter weights to small user-defined constant ϵ
  - $\longrightarrow$ User can limit/strengthen its role by changing ϵ

Hierarchical Bayesian model



Bayesian inference can be accomplished via a Gibbs sampling algorithm.

Hierarchical Bayesian model ···

$$
\begin{array}{rcll}
\alpha_i | a, b & \sim & \beta(\cdot | a, b) & i = 1 \cdots k \\
m_{i,j} | q, r & \sim & \gamma(\cdot | q, r) & i = 1 \cdots k,\ j = 1 \cdots d \\
w_{i,j} & = & \dfrac{m_{i,j}}{\sum_j m_{i,j}} & \\
c_{a,i} | \alpha_i & \sim & Bernoulli(\cdot | \alpha_i) & i = 1 \cdots k \\
e_{a,j} & = & \displaystyle\sum_{i=1}^{k} c_{a,i} \cdot w_{i,j} & a = 1 \cdots n,\ j = 1 \cdots d \\
f_{a,j,i} & = & \dfrac{c_{a,i} \cdot w_{i,j}}{e_{a,j}} & a = 1 \cdots n,\ j = 1 \cdots d,\ i = 1 \cdots k \\
g_{a,j} & \sim & Multinomial(1, \overrightarrow{f_{a,j}}) & a = 1 \cdots n,\ j = 1 \cdots d \\
x_{a,j} & \sim & f_j(\cdot | \theta_{g_{a,j}, j}) & a = 1 \cdots n,\ j = 1 \cdots d
\end{array}
$$

Conditional distributions for model parameters

$$
\begin{aligned}
F(\alpha_i|\cdot) &\propto \beta(\alpha_i|a,b)\cdot\alpha_i^{nactive_i}\cdot(1-\alpha_i)^{n-nactive_i}\\
nactive_i &= \sum_a I(c_{a,i}=1)\\
c_{a,i} &= 1 \quad \text{if} \quad \exists j,\, g_{a,j}=i\\
F(c_{a,i}=0|\cdot) &\propto (1-\alpha_i)\cdot\prod_j f_j(x_{a,j}|\theta_{g_{a,j},j})\cdot F(g_{a,j}|c_{a,\star},c_{a,i}=0,m)\\
F(c_{a,i}=1|\cdot) &\propto \alpha_i\cdot\prod_j f_j(x_{a,j}|\theta_{g_{a,j},j})\cdot F(g_{a,j}|c_{a,\star},c_{a,i}=1,m)\\
F(g_{a,j}=i|\cdot) &\propto f_j(x_{a,j}|\theta_{g_{a,j},j})\cdot\frac{w_{i,j}\cdot I(c_{a,i}=1)}{\sum_i w_{i,j}\cdot I(c_{a,i}=1)}\\
F(m_{i,j}|\cdot) &\propto \gamma(m_{i,j}|q,r)\cdot\prod_a\prod_j\frac{w_{g_{a,j},j}\cdot I(c_{a,g_{a,j}}=1)}{\sum_i w_{i,j}\cdot I(c_{a,i}=1)}
\end{aligned}
$$

Challenges

- Assigning proper prior distributions for all model parameters
- Deriving analytical expressions for all the conditional distributions
- Update to *parameter weights* was very slow because of compute intensive conditional

  $\longrightarrow$ It can be easily approximated by a beta-pdf
- Difficult to visualize and identify results

  $\longrightarrow$ Innovative scheme using KL Divergence

NIPS Papers Dataset

Dataset: NIPS full papers dataset – 1500 papers, 12419 unique words, 6.4 million total words. We consider 1000 most frequent words. Each document is modeled as vector of 0s/1s based on absence/presence of word. So, input is 1500 x 1000 0/1 matrix.

Model: 21-component model with Bernoulli generators. Non-informative priors for appearance probability and parameter weights. $\epsilon = \frac{1}{1000}$.

Iterations: 2000 Gibbs iterations. Results are average over last 1000 iterations.

Details: `http://www.cise.ufl.edu/~ranka/power/`

NIPS Papers Dataset ···

Table 1: Highly-ranked words for some of the components learned from the NIPS dataset. Plain text indicates high importance to the word, as well as a high Bernoulli probability. Bold text indicates high importance but a low Bernoulli probability.

id	α	Words
1	0.3374	arbitrary, assume, asymptotic, bound, case, consider, define, exist, implies, proof, theorem, theory
3	0.1497	acoustic, amplitude, auditory, channel, filter frequency, noise, signal, sound, speaker, speech
5	0.1901	activity, brain, cortex, excitatory, firing, inhibition, membrane, neuron, response, spike, stimuli, synapse
6	0.4025	activation, backpropagation, feedforward, hidden, input, layer, network, neural, output, perceptron, training
7	0.1293	adaptive, control, dynamic, environment, exploration, motor, move, positioning, robot, trajectory, velocity

NIPS Papers Dataset ···

id	α	Words
9	0.2785	class, classifier, clustering, data, dimensionality, features, label, table, testing, training, validation
13	0.2597	dot, edges, field, horizontal, images, matching, object, orientation, perception, pixel, plane, projection, retina, rotation, scene, shape, spatial, vertical, vision, visual
14	0.2557	bayesian, conditional, covariance, density, distribution, estimate, expectation, gaussian, inference, likelihood, mixture, model, parameter, posterior, prior, probability
17	0.1192	analog, bit, chip, circuit, design, diagram, digital, gate, hardware, implement, integrated, output, power, processor, pulse, source, transistor, vlsi, voltage
19	0.3976	**acknowledgement**, **department**, **foundation**, **grant**, **institute**, **research**, **support**, **thank**, **university**

Related work

- Other hierarchical mixture models (Cadez et al., etc)
- Indian Buffet Process (Griffiths and Ghahramani, Heller and Ghahramani)
- Parsimonious mixtures (Graham and Miller)
- Latent Dirichlet Allocation LDA topic models (Blei et al.)