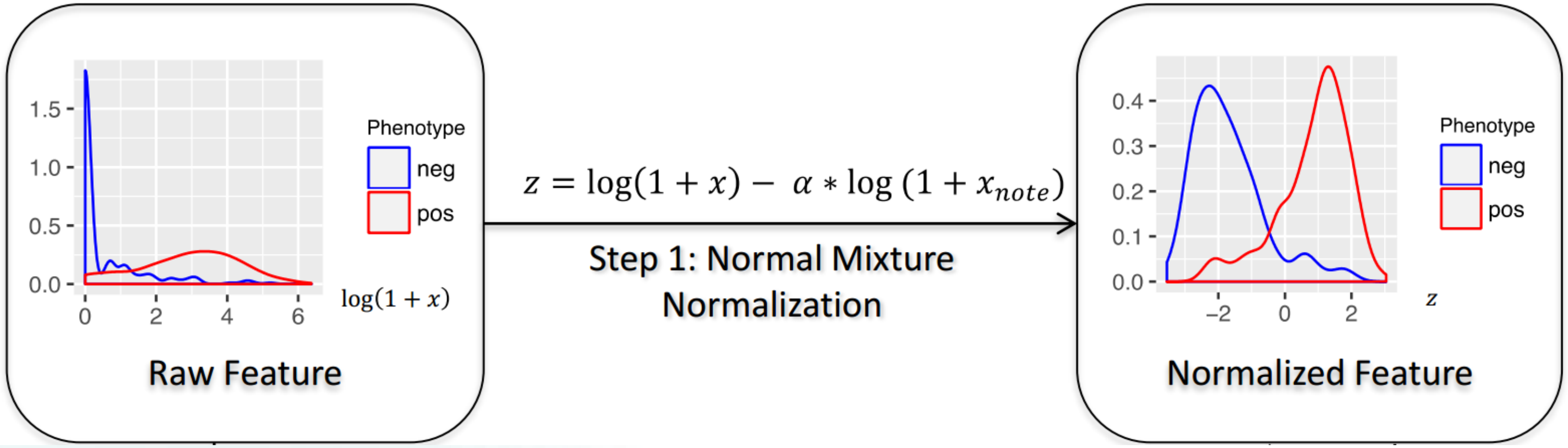
Mixture models and the EM algorithm

俞 声
清华大学统计学研究中心

Example: unsupervised classification

Example: unsupervised classification



Example: unsupervised classification

- Key observation: the main predictors approximately follow a normal mixture distribution after a certain transformation:

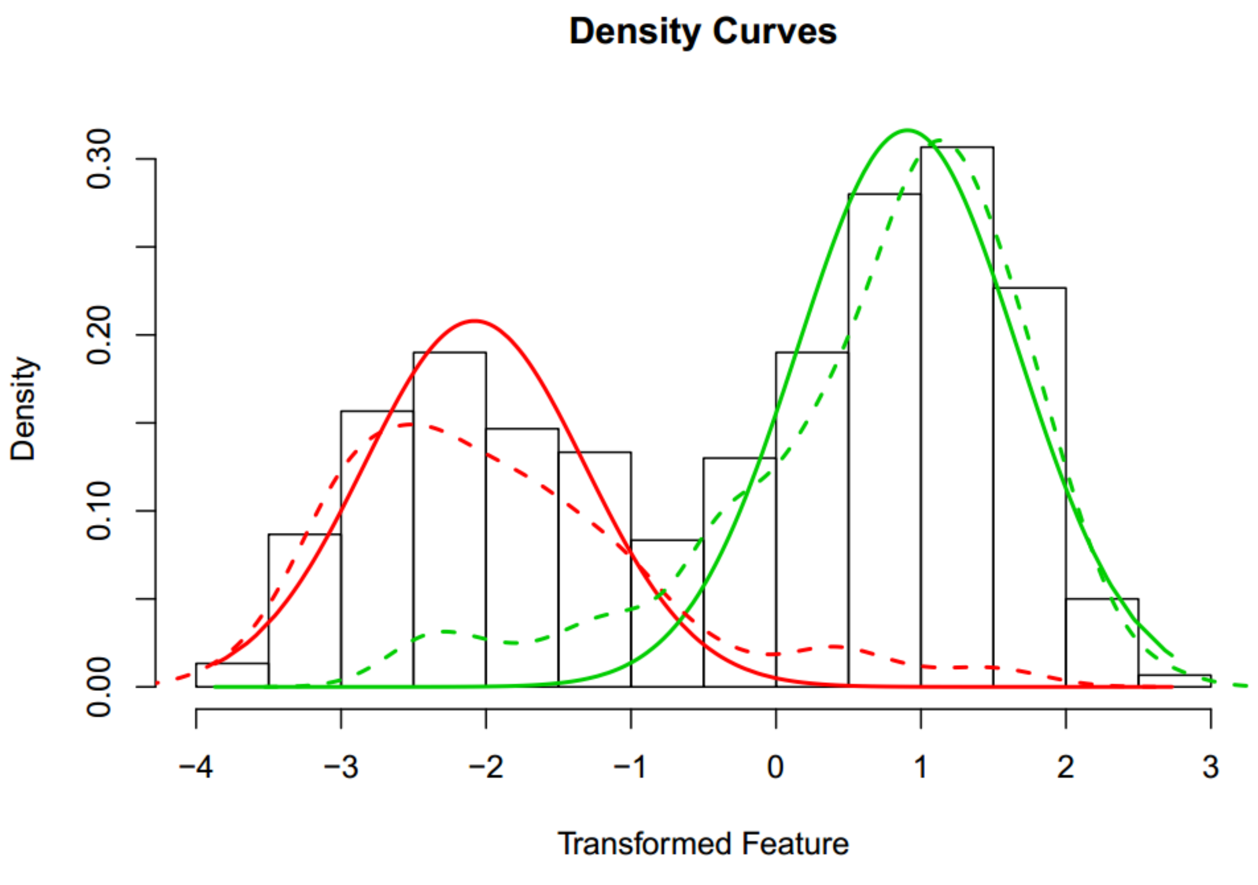
$$z = \log(1 + x) - \alpha \log(1 + x_{note})$$

- Finding the appropriate α: minimize $\mathbb{D}(\alpha)$ with

$$\mathbb{D}(\alpha) = \int_{-\infty}^{+\infty} \left| F_n^\alpha(z) - \lambda\Phi\left(\frac{z-\mu_1}{\sigma}\right) - (1-\lambda)\Phi\left(\frac{z-\mu_0}{\sigma}\right) \right| dz$$

where F_n^α is the empirical CDF of z given parameter α, and Φ is the CDF of the standard normal distribution. $\lambda\Phi\left(\frac{z-\mu_1}{\sigma}\right) + (1-\lambda)\Phi\left(\frac{z-\mu_0}{\sigma}\right)$ is the normal mixture distribution that best fits z given parameter α.

Example: unsupervised classification



Mixture models

- A mixture model is a linear superposition of basic distributions.

- For example, a univariate normal mixture is a linear superposition of normal distributions:

$$\sum_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k^2)$$

where $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$.

Mixture models

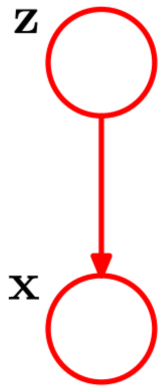
- We can formulate mixture models in terms of latent variables: let z be a random variables and can take values in $\{1 \dots K\}$
$$P(z = k) = \pi_k.$$

- z represents the sub-distribution that x is generated from. For example, for a normal mixture,
$$p(x \mid z = k) = N(x; \mu_k, \sigma_k^2).$$

- Together,
$$p(x) = \sum_{k=1}^{K} p(z = k) p(x \mid z = k) = \sum_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k^2)$$

Mixture models

Graphical representation of a mixture model, in which the joint distribution is expressed in the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$.



Soft-clustering

- A very useful quantity is the posterior probability of the latent label z given the observed value x:

$$P(z=j \mid x) = \frac{p(z=j)p(x \mid z=j)}{\sum_{k=1}^{K} p(z=k)p(x \mid z=k)} = \frac{\pi_j N(x;\mu_j,\sigma_j^2)}{\sum_{k=1}^{K} \pi_j N(x;\mu_k,\sigma_k^2)}$$

MLE for mixture models

- Consider a normal mixture

$$L(\theta; x) = \prod_{i=1}^{N} \sum_{k=1}^{K} \frac{\pi_k}{\sqrt{2\pi\sigma_k^2}} \exp\left\{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right\}$$

- It is difficult to find the maximizers π_k, μ_k, σ_k^2 even for $\log L$.

MLE for mixture models

- On the other hand, we find it easy to find the MLE if we knew z:

$$L(\theta; x, z) = \prod_{i=1}^{N} \frac{\pi_{z_i}}{\sqrt{2\pi\sigma_{z_i}^2}} \exp\left\{-\frac{\left(x_i - \mu_{z_i}\right)^2}{2\sigma_{z_i}^2}\right\}$$

$$\log L(\theta; x, z) = \sum_{i=1}^{N} \log \pi_{zi} - \frac{1}{2} \log 2\pi\sigma_{z_i}^2 - \frac{\left(x_i - \mu_{z_i}\right)^2}{2\sigma_{z_i}^2}$$

- Unfortunately, z is a latent variable that is not observed.

MLE for mixture models

- Since we don' t know the value of z, we can take an expectation with regard to z given x (i.e. the best guess of z) and optimize the expected likelihood:

$$E_{z|x} \log L(\theta; x, z) = \sum_{i=1}^{N} E_{z_i|x_i}\left[\log \pi_{zi} - \frac{1}{2}\log 2\pi\sigma_{z_i}^2 - \frac{\left(x_i - \mu_{z_i}\right)^2}{2\sigma_{z_i}^2}\right]$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} P(z_i = k \mid x_i)\left[\log \pi_k - \frac{1}{2}\log 2\pi\sigma_k^2 - \frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right]$$

- This form is easy to optimize. The problem is that we don' t know $P(z_i = k \mid x_i)$ because there are unknown parameters.

EM algorithm

To solve this problem, the expectation-maximization (EM) algorithm estimates $P(z_i = k \mid x_i)$ and optimizes $E_{z|x} \log L(\theta; x, z)$ in a iterative fashion.

Initialize the parameters with some reasonable guess or random values.

- E-step: estimate $\hat{z}_{ik} = P(z_i = k \mid x_i)$ with the current estimates of the parameters $\hat{\theta}$.
- M-step: update the parameters with

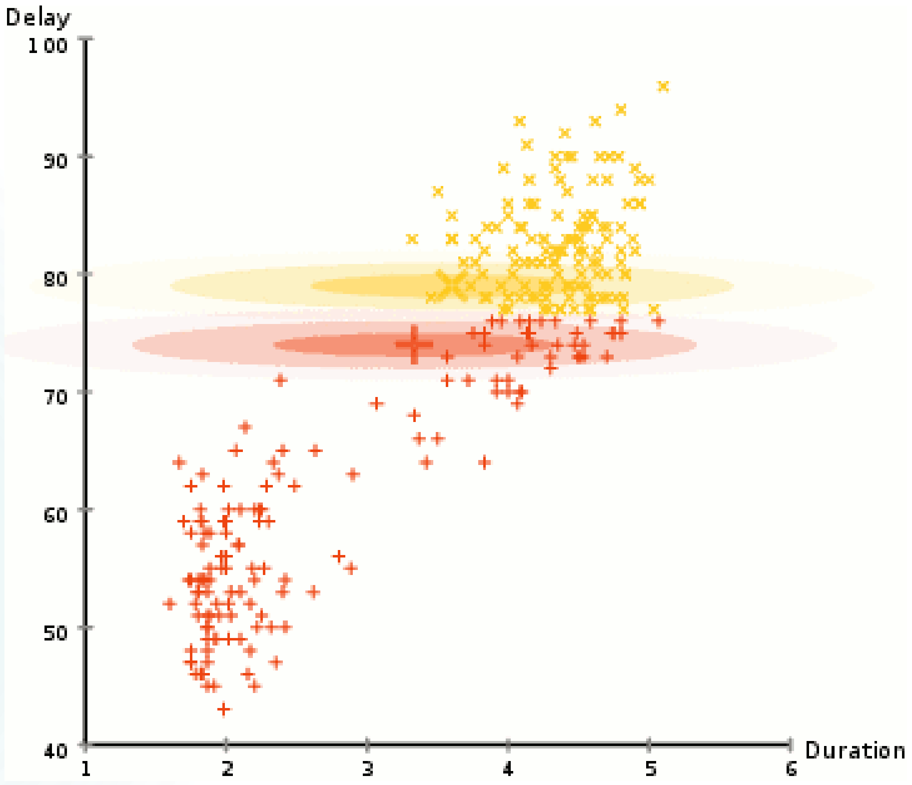
$$\hat{\theta}_{new} = \arg\max E_{z|x} \log L(\theta; x, z) = \sum_{i=1}^{N} \sum_{k=1}^{K} \hat{z}_{ik} \log p(x_i, z_i = k)$$

Theoretical support

- In general, by Jensen' s inequality, $E_{z|x} \log L(\theta; x, z)$ is a lower bound of $\log L(\theta; x)$.

- Setting $P(z_i = k) = P(z_i = k \mid x_i)$ makes the equality hold. Thus, the EM algorithm makes the log-likelihood $\log L(\theta; x)$ grow monotonically and converge.

EM for multivariate normal



Practice

- Develop the EM algorithm for normal mixture distributions with unknown parameters π_k, μ_k, σ_k^2.

- What if $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$?

Example: TCM data mining

- A TCM prescription is a combination of herbs that may look like this:

  柴胡 黄芩 陈皮 竹茹 赤芍 枳壳 元胡 当归 黄芪 郁金 茯苓 三棱 蒺藜 姜半夏

- It is known that physicians prescribe herbs in ‘combos’ known as 配伍组合 rather than in individuals. E.g., 柴胡+黄芩、当归+黄芪 are both common combos.

- Finding the combos in a prescription is useful for dimension reduction, as combos represent treatment to different conditions and have better independence from each other than basic herbs.

Example: TCM data mining

- We have developed a statistical method to generate a collection of possible combos $\{c_1, \ldots, c_M\}$, where individual herbs are also counted as possible combos. We want to know which ones to keep.

- A natural way to identify the good combos is to find those with the highest probability of being used.

- Independence assumption: each combo is used independent of the others. The probability that combo c_m is used is p_m.

Example: TCM data mining

- Estimating p_m would be easy if we knew which combos were used in each prescription. However, the combos are not observed, and each prescription has multiple ways for decomposition into combos.

- We need:
  - a way to estimate $p_1 \dots p_M$ without knowing the combos in each prescription (the EM algorithm)
  - a way for fast decomposition of prescriptions into combos

Practice

- Denote the prescriptions by $O_1, \dots, O_N$, and the actual combo decomposition by $Y_1, \dots, Y_N$. Denote the possible decompositions of O_i by $D(O_i)$, and let $d_{i,k} \in D(O_i)$ be the k-th possible decomposition of O_i. Let $x_{i,k,m} = 0/1$ denote whether c_m is used in $d_{i,k}$.

- Use the EM algorithm to estimate $p_1 \dots p_M$.

Practice

- How do you quickly find $D(O_i)$?

- Requirements
  - For a decomposition $d_{i,k}$, c_m and c_n don' t have any common component if $x_{i,k,m} = x_{i,k,n} = 1$; and
  - $\bigcup_{x_{i.k.m}=1} c_m = O_i$.