Max Entropy and EM

七月在线 褚则伟

2017年5月21日

Plan

- Entropy
- Principle of Max Entropy and Max Entropy Model
- Review (maybe preview?) of logistic regression and SoftMax regression
- Max Entropy and SoftMax models
- Expectation Maximization (EM)
- EM in Gaussian Graphical Models (GMM)

Entropy

- Entropy是对一个random variable的不确定性的描述

- 转盘和骰子哪一个的不确定性更高？哪一个Entropy更高？
- 现在有一个正常的骰子和一个作弊的骰子，哪一个Entropy更高？

Entropy

- Entropy是对一个random variable的不确定性的描述
- X: random variable; X takes values $\{x_1, x_2, \ldots, x_n\}$; and is defined by a probability distribution P(X), then we write the Entropy of the random variable as

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x)$$

- If the log is taken to be to the base 2, then the entropy is expressed in bits. natural log: nats.

Entropy

- Example: Compute the entropy of a fair coin.

$$P(X = heads) = \frac{1}{2} \qquad P(X = tails) = \frac{1}{2}$$

$$\begin{aligned} H(P) &= -\sum_{x \in \{heads, tails\}} P(x) \log P(x) \\ &= -\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right] \\ &= -\left[-\frac{1}{2} + -\frac{1}{2}\right] \\ &= 1. \end{aligned}$$

Entropy

- Example: Let X be an unfair 6-sided die with probability distribution defined by P(X = 1) = ½, P(X = 2) = ¼, P(X = 3) = 0, P(X = 4) = 0, P(X = 5) = 1/8, P(X = 6) = 1/8. The entropy is

$$\begin{aligned} H(P) &= -\sum_{x\in\{1,2,3,4,5,6\}} P(x)\log P(x) \\ &= -\left[\frac{1}{2}\log\frac{1}{2}+\frac{1}{4}\log\frac{1}{4}+0\log 0+0\log 0+\frac{1}{8}\log\frac{1}{8}+\frac{1}{8}\log\frac{1}{8}\right] \\ &= -\left[-\frac{1}{2}+-\frac{1}{2}+0+0+-\frac{3}{8}+-\frac{3}{8}\right] \\ &= 1.75. \end{aligned}$$

Properties of Entropy

1. $H \geq 0$ (obvious)
2. H is a concave function of p (we know $x \ln x$ is convex)
3. $H(p) = 0$ iff p deterministic
4. $H(p) = \max$ for $p = u$=the uniform distribution; in this case $H(u) = \ln |\Omega|$

   Proof Let $\Omega = \{0, 1, \ldots m\}$ w.l.o.g. Then, H is a function of the m variables $p_{1:m}$, with $\sum_{x=0}^{m} p_x = 1$ and $p_0 = 1 - \sum_{x=1}^{m} p_x$.

$$H(p) = -\sum_{x=1}^{m} p_x \ln p_x - (1 - \sum_{x=1}^{m} p_x) \ln(1 - \sum_{x=1}^{m} p_x) \tag{3}$$

$$\frac{\partial H}{\partial p_x} = -\ln p_x - 1 + \ln(1 - \sum_{x=1}^{m} p_x) + 1 = 0 \tag{4}$$

$$p_x = (1 - \sum_{x'=1}^{m} p_{x'}) = p_0 \tag{5}$$

   In other words, all p_x must be equal.

Joint Entropy

- Joint entropy is the entropy of a joint probability distribution, or a multi-valued random variable.

$$H(P(E,C)) = -\sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}} P(e,c) \log P(e,c)$$

Conditional Entropy

- Conditional Entropy is defined as

$$
\begin{aligned}
H(X|Y) &= -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(y)} \\
&= \sum_{y} p(y) H(X|Y=y).
\end{aligned}
$$

- It is easy to see that H(X|Y) = H(X, Y) – H(Y)

Mutual Information

- Mutual information measures a relationship between two random variables that are sampled simultaneously.
- It measures how much information is communicated, on average, in one variable about another. How much does one variable tell me about another?
- For example, suppose X represents the roll of a fair 6-sided die, and Y represents whether the roll is even (0 even, 1 odd). Clearly, the value of Y tells us something about the value of X and vice versa. X and Y share **mutual information**.
- Definition of Mutual Information:

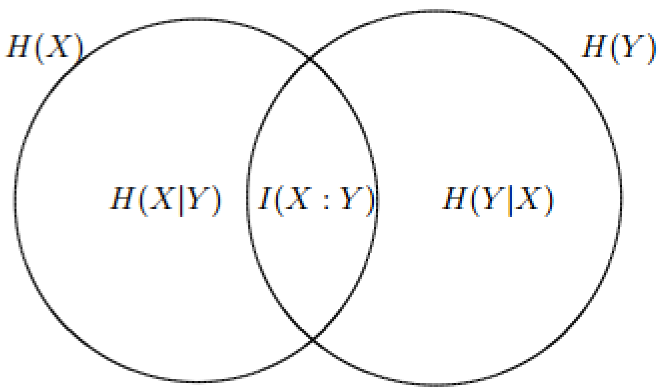
$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

- It's not hard to see that

$$I(X:Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y).$$

Summary

- Relationship of entropy, conditional entropy, mutual information



Kullback-Leibler Divergence

- Also named relative entropy. It measures the closeness of two probability distributions.

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

- KL divergence经常被作为学到的分布好坏的指标

Max Entropy Model

- 最大熵原理：在学习概率模型的时候，在所有可能的概率模型中，熵最大的模型是最好的模型。
- Example: 若随机变量X有4个取值{A, B, C, D}，要顾及各个值的概率{P(A), P(B), P(C), P(D)}。

约束条件

$$P(A)+P(B)+P(C)+P(D)=1$$

可行解

$$P(A)=P(B)=P(C)=P(D)=1/4$$

加入先验

$$P(A)+P(B)=3/10$$

$$P(A)+P(B)+P(C)+P(D)=1$$

可行解

$$P(A)=P(B)=3/20$$

$$P(C)=P(D)=7/20$$

Max Entropy Model

- Imagine that we have a dataset *D* of MNIST (32x32 images of digits). We want to assign probabilities to new images. If we let *X* represent the space of all possible binary 32x32 images, we want probability distribution *p(X)*.
- Now the question is how to constrain this probability distribution. One thing we could d is to constrain our probability distribution to match the empirical distribution. This is a poor choice, since if we flip even a single pixel in a training image, we assign 0 probability to the new image. So we want some sort of smoothness in our semantic space.
- We can use a more relaxed constraint – we want the expectation of each feature on the empirical distribution to match the expectation of each feature on our model's distribution.

Max Entropy Model

Assume that we have a set of N observations $\mathcal{D} = \{x^{(1)}, \ldots x^{(N)}\} \subseteq \Omega$ from an unknown distribution p. The observation define the **empirical distribution** $\tilde{p}$

$$\tilde{p}\,(x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(j)}}(x)$$

where

$$\delta_{x^{(j)}}(x) \;=\; \begin{cases} 1 & x = x^{(j)} \\ 0 & \text{otherwise} \end{cases}$$

Max Entropy Model

We also have a set of **features** $f_i(x)$ of the data; they are functions $f_i : \Omega \rightarrow (-\infty, \infty)$, $i = 1, \ldots K$.

The **Maximum Entropy Principle** states that the "best" model of the data is the distribution q^* representing the solution to the following problem

$$\max_q H(q) \quad \text{s.t.} \quad E_q[f_i] = E_{\tilde{p}}[f_i] \quad \text{for all} \quad i = 1, \ldots K.$$

- We are looking for a distribution that has the same marginal as the empirical distribution

Max Entropy Model

- 定义Lagrangian

$$L(q,\lambda) = -H(q) - \sum_i \lambda_i (E_q[f_i] - E_{\tilde{p}}[f_i]) - \lambda_0 (\sum_{x\in\Omega} q - 1)$$

- 两边求导

$$\frac{\partial L}{\partial q(x)} = \log q(x) + 1 - \sum_i \lambda_i f_i(x) - \lambda_0$$

Max Entropy Model

- 化简后得到

$$
\begin{aligned}
\log q(x) &= \sum_i \lambda_i f_i(x) + \lambda_0 - 1 \\
&\text{or} \\
q(x) &\propto e^{\sum_i \lambda_i f_i(x)} \\
&\text{or} \\
q &= \frac{1}{Z_\lambda} e^{\lambda^T f}
\end{aligned}
$$

$$
Z_\lambda = \sum_{x \in \Omega} e^{\sum_i \lambda_i f_i(x)}
$$

Improved Iterative Scaling

- 考虑一个conditional exponential model

$$p_\Lambda(y \mid x) \equiv \frac{1}{Z_\Lambda(x)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(x, y)\right)$$

- 现在的问题是如何优化λ的值，使用Improved Iterative Scaling. (refer to the notes attached if you are interested.)
- 其实更简单的方法是直接用Gradient Descent或者Stochastic Gradient Descent，各种Deep Learning Framework都有提供。

What is Max Entropy Model?

- Log Linear model, maximum entropy model, exponential family model, energy based model, Bboltzmann distribution, conditional random fields. They are all essentially equivalent. They all have the following form:

$$p(y \mid x) = \frac{1}{Z_x} \exp \vec{\theta} \cdot \vec{f}(x, y)$$

Logistic Regression

- Remember logistic regression?

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^\top x)}$$

- Train to minimize the cross entropy cost function:

$$J(\theta) = -\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

- Remember the definition of cross entropy?

$$\mathrm{E}_p[-\log q] = H(p) + D_{\mathrm{KL}}(p\|q)$$

SoftMax Regression

- Multi-class case: SoftMax Regression!

$$h_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)} \begin{bmatrix} \exp(\theta^{(1)\top} x) \\ \exp(\theta^{(2)\top} x) \\ \vdots \\ \exp(\theta^{(K)\top} x) \end{bmatrix}$$

- Still cross entropy loss:

$$J(\theta) = -\left[\sum_{i=1}^{m} \sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})} \right]$$

SoftMax Regression

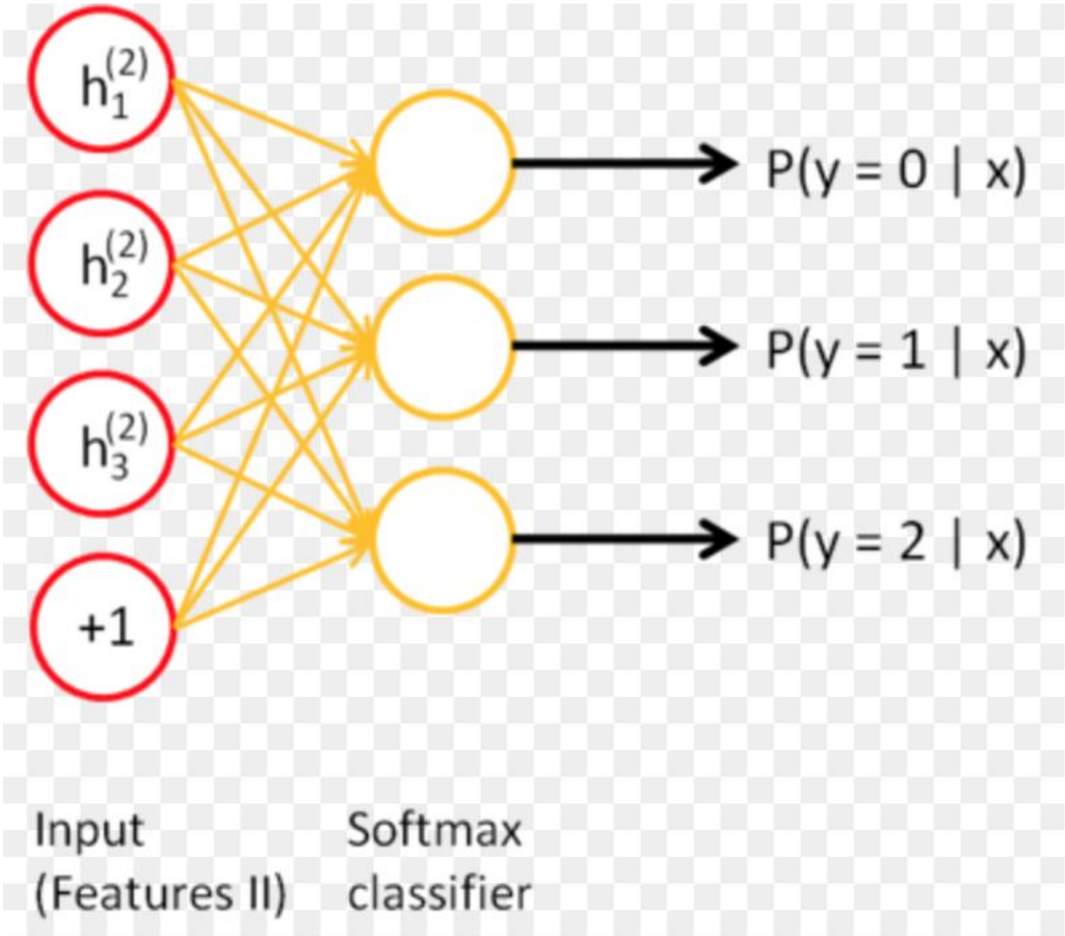
- Classifier:

$$P(y^{(i)} = k | x^{(i)}; \theta) = \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}$$

- How to train? Stochastic Gradient Descent!

$$\nabla_{\theta^{(k)}} J(\theta) = -\sum_{i=1}^{m} \left[x^{(i)} \left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta) \right) \right]$$

SoftMax Regression



Mixture Model and EM

- Unsupervised Learning, used in clustering
- Mixture Models: Assume data generated using the following procedure:
  - Pick one of *k* components according to $z = \pi()$
  - Generate a data point by sampling from $p(x|z)$

Multivariate Gaussians

$$\mathcal{N}(\mathbf{x};\,\boldsymbol{\mu},\boldsymbol{\Sigma}) \;=\; \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- Gaussian Likelihood:

$$\log\mathcal{N}(\mathbf{x};\,\boldsymbol{\mu},\boldsymbol{\Sigma}) \;=\; -\frac{d}{2}\log 2\pi \;-\; \frac{1}{2}\log|\boldsymbol{\Sigma}| \;-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

Maximum likelihood for the mean:

$$\widehat{\boldsymbol{\mu}}_{ML} \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

Maximum likelihood for the covariance:

$$\widehat{\boldsymbol{\Sigma}}_{ML} \;=\; \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^{\intercal}.$$

Generative Models

Construct for each class c

$$\delta_c(\mathbf{x}) \triangleq \log p(\mathbf{x} \mid y = c) + \log p(y = c)$$

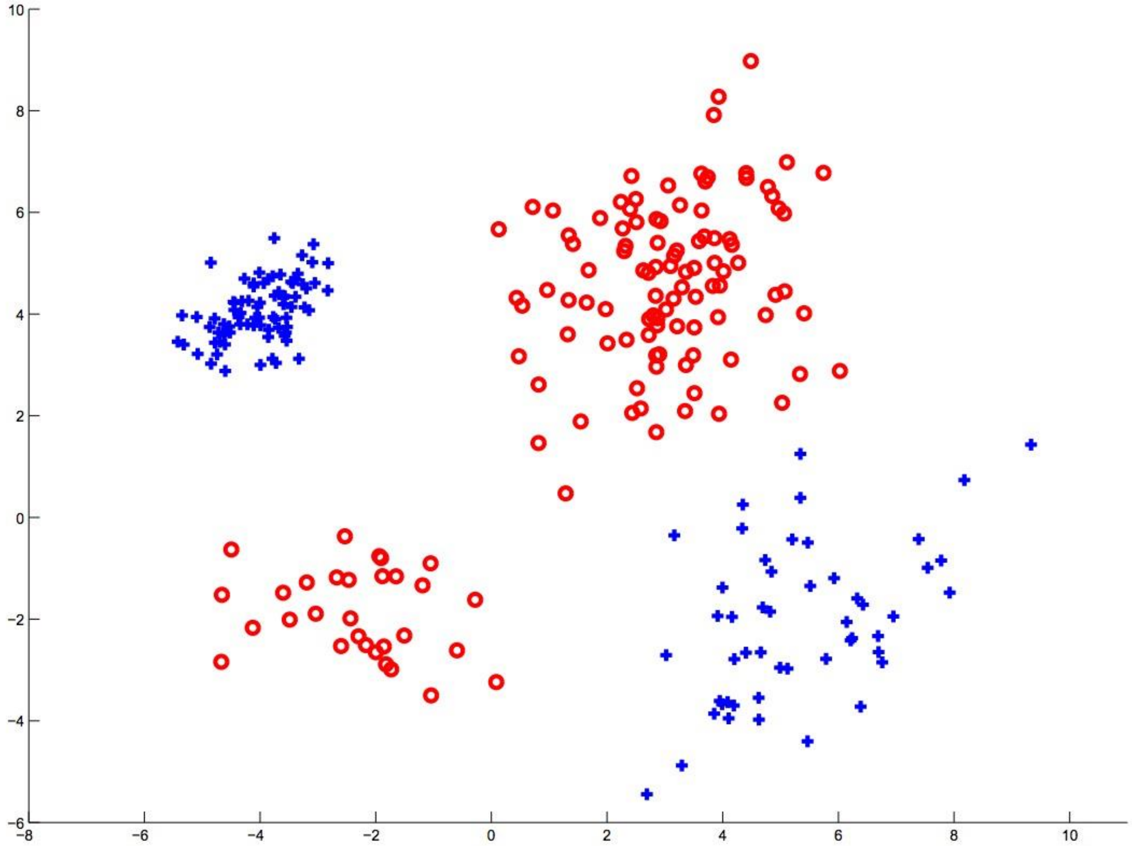
based on our per-class (class-conditional) model $p(\mathbf{x} \mid y = c)$

Generative classifier:

$$h^*(\mathbf{x}) = \operatorname*{argmax}_c \delta_c(\mathbf{x}).$$

If assume equal priors $p(y = c) = 1/C$, then
$h^*(\mathbf{x}) = \operatorname{argmax}_c \log p(\mathbf{x} \mid y = c)$.

Mixture Models



Mixture of Gaussians

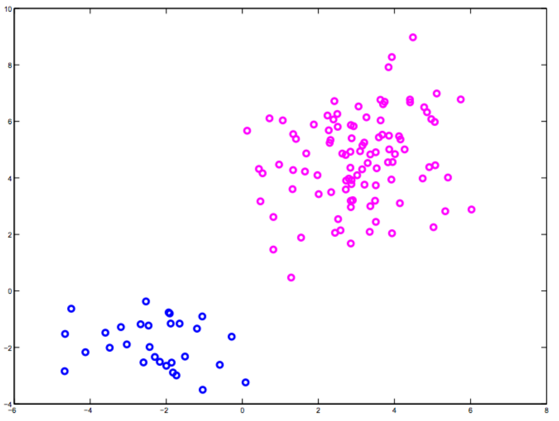
k underlying types (components);

Each component is Gaussian;

y_i is the identity of the component "responsible" for $\mathbf{x}_i$;

y_i is a *hidden* (*latent*) variable: never observed.

A *Gaussian mixture model*:



$$p(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{c=1}^{k} \pi_c \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c).$$

π_cs are the *mixing probabilities*, $\pi_c = p(y = c)$

Gaussian Mixture Model

- Gaussian Mixture Model:

$$p(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{c=1}^{k} \pi_c \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

π_cs are the *mixing probabilities*, $\pi_c = p(y = c)$

Mixture density estimation

Suppose that we do observe $y_i \in \{1, \ldots, k\}$ for each $i = 1, \ldots, N$.

Let us introduce a set of binary *indicator variables* $\mathbf{z}_i = [z_{i1}, \ldots, z_{ik}]$ where

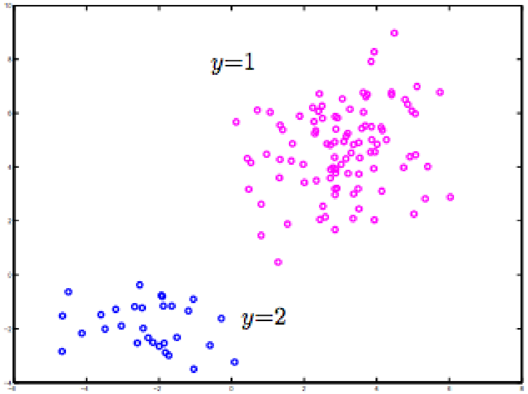
$$z_{ic} = 1 \;=\; \begin{cases} 1 & \text{if } y_i = c, \\ 0 & \text{otherwise.} \end{cases}$$

The count of examples from c-th component:

$$N_c \;=\; \sum_{i=1}^{N} z_{ic}.$$

Mixture density estimation: known labels

If we know $\mathbf{z}_i$, the ML estimates of the Gaussian components, just like in class-conditional model, are



$$\widehat{\pi}_c = \frac{N_c}{N},$$

$$\widehat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i=1}^{N} z_{ic}\mathbf{x}_i,$$

$$\widehat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c}\sum_{i=1}^{N} z_{ic}(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)^T.$$

Credit assignment

When we don't know $\mathbf{z}_i$, we face a *credit assignment* problem: which component is responsible for $\mathbf{x}_i$?

Suppose for a moment that we do know component parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ and mixing probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$.

Then, the posterior of each label using Bayes rule:

$$\gamma_{ic} \;=\; \widehat{p}(y = c \,|\, \mathbf{x};\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;=\; \frac{\pi_c \cdot p(\mathbf{x};\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l \cdot p(\mathbf{x};\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

We will call γ_{ic} the *responsibility* of the c-th component for $\mathbf{x}$.

- Note: $\sum_{c=1}^{k} \gamma_{ic} = 1$ for each i.

Expected log likelihood

$$\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \text{const} + \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c + \log \mathcal{N} \left(\mathbf{x}_i ; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c \right) \right).$$

Expectation of z_{ic}:

$$E_{z_{ic} \sim \gamma_{ic}} \left[z_{ic} \right] = \sum_{z \in 0,1} z \cdot \gamma_{ic}^{z} = \gamma_{ic}.$$

The expected likelihood of the data:

$$\begin{aligned} E_{z_{ic} \sim \gamma_{ic}} \left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \right] &= \text{const} \\ &+ \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c + \log \mathcal{N} \left(\mathbf{x}_i ; \mu_c, \boldsymbol{\Sigma}_c \right) \right). \end{aligned}$$

Expectation Maximization

$$E_{z_{ic}\sim\gamma_{ic}}\left[\log p(X_N, Z_N; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)\right] = \sum_{i=1}^{N}\sum_{c=1}^{k}\gamma_{ic}\left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i; \mu_c, \boldsymbol{\Sigma}_c\right)\right)$$

We can find $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize this *expected* likelihood – by setting derivatives to zero and, for $\boldsymbol{\pi}$, using Lagrange multipliers to enforce $\sum_c \pi_c = 1$.

$$\hat{\pi}_c = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ic},$$

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{\sum_{i=1}^{N}\gamma_{ic}}\sum_{i=1}^{N}\gamma_{ic}\mathbf{x}_i,$$

$$\widehat{\boldsymbol{\Sigma}}_c = \frac{1}{\sum_{i=1}^{N}\gamma_{ic}}\sum_{i=1}^{N}\gamma_{ic}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T.$$

What do we have now?

If we know the **parameters** and **indicators** (assignments) we are done.

If we know the **indicators** but not the parameters, we can do ML estimation of the parameters – and we are done.

If we know the **parameters** but not the indicators, we can compute the posteriors of indicators;

- With known posteriors, we can estimate parameters that maximize the *expected* likelihood – and then we are done.

But in reality we know neither the parameters nor the indicators.

The EM algorithm

Start with a guess of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$.

- Typically, random Gaussians and $\pi_c = 1/k$.

Iterate between:

E-step Compute values of expected assignments, i.e. calculate γ_{ic}, using current estimates of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$.

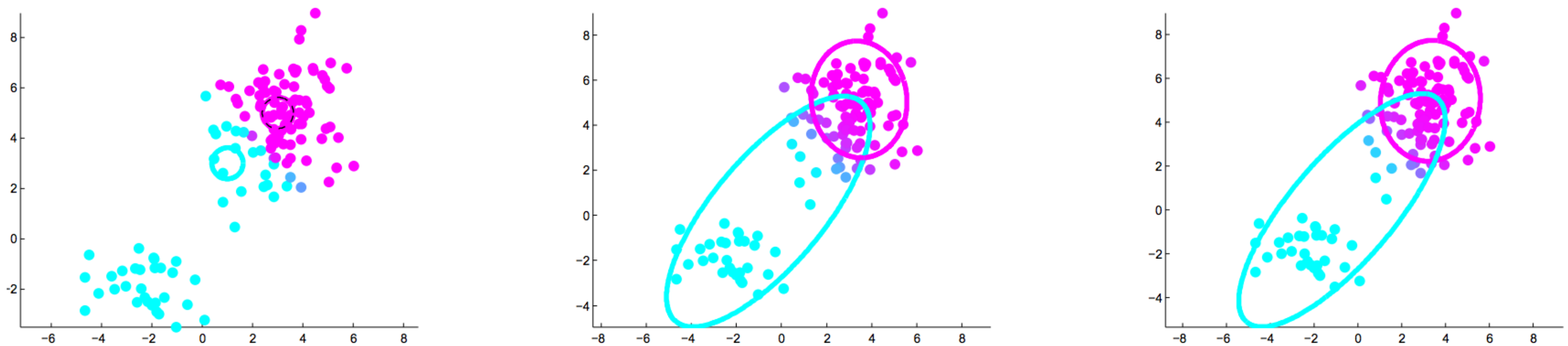
M-step Maximize the *expected* likelihood, under current γ_{ic}.

Repeat until convergence.

EM for Gaussian Mixture: an example

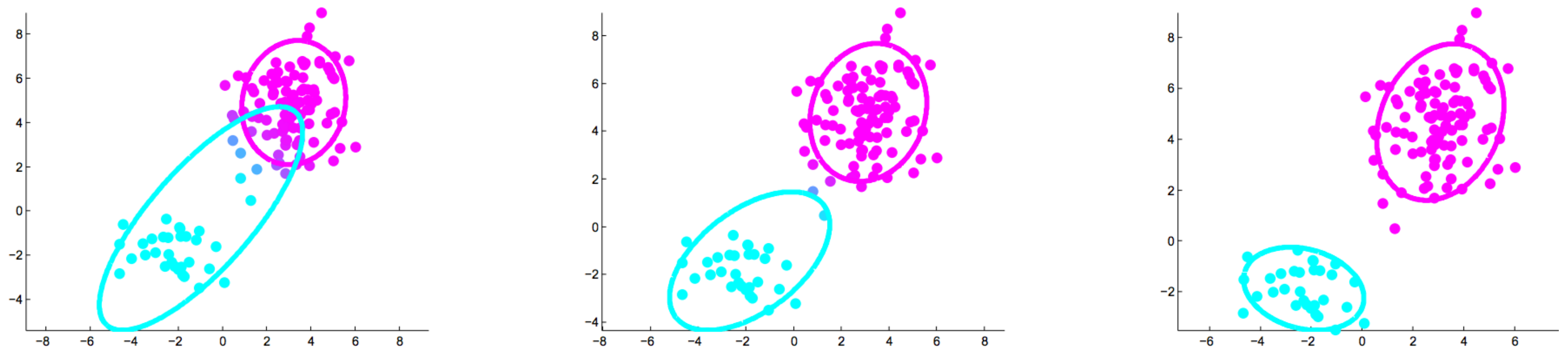
Colors represent γ_{ic} after the E-step.

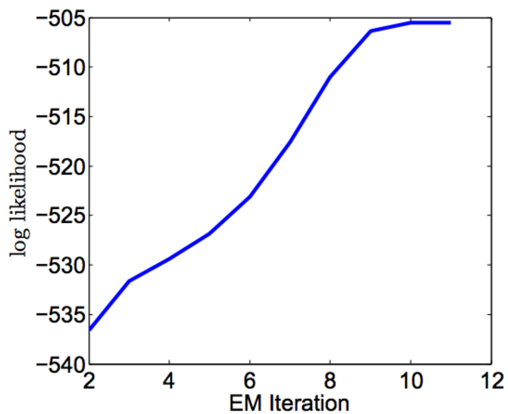
1st iteration　　2nd iteration　　3rd iteration



EM for Gaussian Mixture: an example

4th iteration | 7th iteration | 10th iteration





Log-likelihood progress with iterations

Generic EM for mixture models

General mixture models: $p(\mathbf{x}) = \sum_{c=1}^{k} \pi_c \, p(\mathbf{x}; \boldsymbol{\theta}_c)$

Initialize $\boldsymbol{\pi}$, $\boldsymbol{\theta}^{old}$, and iterate until convergence:

E-step: compute responsibilities

$$\gamma_{ic} = \frac{\pi_c^{old} \, p(\mathbf{x}_i; \boldsymbol{\theta}_c^{old})}{\sum_{l=1}^{k} \pi_l^{old} \, p(\mathbf{x}_i; \boldsymbol{\theta}_l^{old})}.$$

M-step: re-estimate mixture parameters:

$$\boldsymbol{\pi}^{new}, \boldsymbol{\theta}^{new} = \operatorname*{argmax}_{\boldsymbol{\theta}, \boldsymbol{\pi}} \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c + \log p(\mathbf{x}_i; \boldsymbol{\theta}_c) \right).$$

The EM algorithm in general

Observed data X, hidden variables Z.

- E.g., *missing data*.

Initialize θ^{old}, and iterate until convergence:

E-step: Compute the expected complete data log-likelihood as a function of θ.

$$Q\left(\theta;\, \theta^{old}\right) \;=\; E_{p(Z\,|\,X,\theta^{old})}\left[\log p(X, Z;\, \theta)\,|\,X,\, \theta^{old}\right]$$

M-step: Compute

$$\theta^{new} \;=\; \underset{\theta}{\operatorname{argmax}}\, Q\left(\theta;\, \theta^{old}\right).$$

Why does EM work?

Ultimately, we want to maximize likelihood of the *observed* data

$$\theta^* = \operatorname*{argmax}_\theta \log p(X; \theta).$$

Let $\log p^{(t)}$ be $\log p(X; \theta^{new})$ after t iterations.

Can show:

$$\log p^{(0)} \leq \log p^{(1)} \leq \ldots \leq \log p^{(t)} \ldots$$

A more general case for EM

- Suppose we want to maximize the following function:

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{m} \log p(x;\theta) \\ &= \sum_{i=1}^{m} \log \sum_{z} p(x,z;\theta).\end{aligned}$$

A more general case for EM

- Using Jensen's inequality:

$$
\begin{aligned}
\sum_i \log p(x^{(i)};\theta) &= \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)};\theta) \\
&= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})} \\
&\geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}
\end{aligned}
$$

- Naturally, we want to maximize the lower bound by:

$$
\begin{aligned}
Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)};\theta)}{\sum_z p(x^{(i)}, z;\theta)} \\
&= \frac{p(x^{(i)}, z^{(i)};\theta)}{p(x^{(i)};\theta)} \\
&= p(z^{(i)}|x^{(i)};\theta)
\end{aligned}
$$

A more general case for EM

- Naturally, we want to maximize the lower bound by (E-step):

$$
\begin{aligned}
Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{\sum_z p(x^{(i)}, z; \theta)} \\
&= \frac{p(x^{(i)}, z^{(i)}; \theta)}{p(x^{(i)}; \theta)} \\
&= p(z^{(i)} | x^{(i)}; \theta)
\end{aligned}
$$

- Then we maximize the log likelihood by (M-step)

$$
\theta := \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}
$$

The EM algorithm in general

Repeat until convergence

(E-step) For each i, set

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta).$$

(M-step) Set

$$\theta := \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}.$$

Mixture Model: Example

- Average height of people in different ethnicities (African, Caucasian, Asian, Latino). Assume the height distribution is different within each ethnicity, and it follows a Gaussian distribution. The weighting factor may be the percentage of the population that are from each ethnic group. This would be a 4-point Gaussian Mixture model.

Reading

- Andrew Ng CS229 Lecture Notes http://cs229.stanford.edu/notes/cs229-notes8.pdf
- http://l2r.cs.uiuc.edu/~danr/Teaching/CS598-05/Lectures/Lec8-maxent.pdf
- https://www.stat.washington.edu/courses/stat538/winter12/Handouts/l8-maxent.pdf

Thank you!

Zewei Chu