LECTURE 9: THE EM (EXPECTATION-MAXIMIZATION) ALGORITHM

STAT 545: Intro. to Computational Statistics

Vinayak Rao
Purdue University

September 18, 2019

The Multivariate normal (MVN) density on $\mathbb{R}^d$:

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Given N i.i.d. observations $X \equiv \{x_1, \ldots, x_N\}$, the likelihood is

$$\mathcal{L}(X|\mu, \Sigma) = \prod_{i=1}^{N} p(x_i|\mu, \Sigma)$$

NORMAL

The Multivariate normal (MVN) density on $\mathbb{R}^d$:

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

Given N i.i.d. observations $X \equiv \{x_1, \ldots, x_N\}$, the likelihood is

$$\mathcal{L}(X|\mu, \Sigma) = \prod_{i=1}^{N} p(x_i|\mu, \Sigma)$$

Maximum likelihood estimation (MLE): learn parameters by maximizing $\mathcal{L}(X|\mu, \Sigma)$ w.r.t μ and Σ.

How? Calculate derivatives and set to 0.

MLE for the MVN

More convenient is the log-likelihood $\ell(X|\mu,\Sigma) = \log \mathcal{L}(X|\mu,\Sigma)$:

$$\ell(X|\mu,\Sigma) = \sum_{i=1}^{N} \log p(x_i|\mu,\Sigma)$$

For the Gaussian,

$$\ell(X|\mu,\Sigma) = -\frac{1}{2}\sum_{o=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) - \frac{N}{2}\log|\Sigma| - \text{const}$$

MLE for the MVN

More convenient is the log-likelihood $\ell(X|\mu,\Sigma) = \log \mathcal{L}(X|\mu,\Sigma)$:

$$\ell(X|\mu,\Sigma) = \sum_{i=1}^{N} \log p(x_i|\mu,\Sigma)$$

For the Gaussian,

$$\ell(X|\mu,\Sigma) = -\frac{1}{2}\sum_{o=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) - \frac{N}{2}\log|\Sigma| - \text{const}$$

$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \Sigma_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_{ML})(x_i-\mu_{ML})^\top$$

MLE for the MVN

More convenient is the log-likelihood $\ell(X|\mu,\Sigma) = \log \mathcal{L}(X|\mu,\Sigma)$:

$$\ell(X|\mu,\Sigma) = \sum_{i=1}^{N} \log p(x_i|\mu,\Sigma)$$

For the Gaussian,

$$\ell(X|\mu,\Sigma) = -\frac{1}{2}\sum_{o=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) - \frac{N}{2}\log|\Sigma| - \text{const}$$

$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \Sigma_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_{ML})(x_i-\mu_{ML})^\top$$

MLE: moment matching (set mean/covariance to that of data)

Holds for *exponential family distributions* (later)

Discrete distribution

Consider a K-component discrete distribution $\pi = (\pi_1, \dots, \pi_K)$

- for $X \sim \pi$, $p(X = c) = \pi_c$.

Discrete distribution

Consider a K-component discrete distribution $\pi = (\pi_1, \ldots, \pi_K)$

- for $X \sim \pi$, $p(X = c) = \pi_c$.
- Equivalently,

$$p(X) = \prod_{c=1}^{K} \pi_i^{\delta(X=c)} = \exp(\sum_{c=1}^{K} \delta(X = c) \log \pi_c)$$

Discrete distribution

Consider a K-component discrete distribution $\pi = (\pi_1, \ldots, \pi_K)$

- for $X \sim \pi$, $p(X = c) = \pi_c$.
- Equivalently,

$$p(X) = \prod_{c=1}^{K} \pi_i^{\delta(X=c)} = \exp(\sum_{c=1}^{K} \delta(X = c) \log \pi_c)$$

Given data, what is MLE of π?

$$\pi_c = \frac{1}{N} \sum_{i=1}^{N} \delta(x_i = c)$$

Back to clustering

Last week we saw a few clustering algorithms.

We also saw some limitations:

- Limited control on the cluster shapes (e.g. spherical clusters in k-means).
- Cannot capture variability across clusters.
- Cannot capture uncertainty in cluster assignments.
- Cannot capture information about relative cluster sizes.

Model-based clustering

We could adjust loss-function/optimization algorithm.

Different approach: directly model data-generation process

- Can capture much richer structure more intuitively.
- Can make predictions about future data.
- Can deal with missing data naturally.

Finite mixture models

Like k-means, fix the number of clusters to K.

- component c has parameter θ_c
- observations from cluster c distributed as $p(x|\theta_c)$

Finite mixture models

Like k-means, fix the number of clusters to K.

- component c has parameter θ_c
- observations from cluster c distributed as $p(x|\theta_c)$

Draw cluster from π, a K-component probability vector

Finite mixture models

Like k-means, fix the number of clusters to K.

- component c has parameter θ_c
- observations from cluster c distributed as $p(x|\theta_c)$

Draw cluster from π, a K-component probability vector

Today we will consider the mixture of Gaussians (MoG)

- each component is a Gaussian
- $\theta_c = (\mu_c, \Sigma_c)$ is its mean and covariance

Mixture of Gaussians (MoG)

To generate the *i*th observation:

$$c_i \sim \pi \qquad \text{Sample it's cluster assignment}$$

$$x_i \sim \mathcal{N}(x_i | \mu_{c_i}, \Sigma_{c_i}) \qquad \text{Sample it's value}$$

Mixture of Gaussians (MoG)

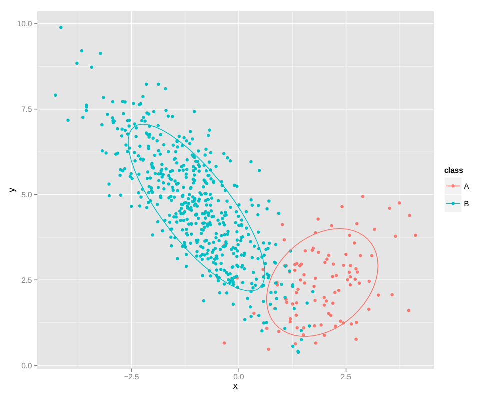
To generate the *i*th observation:

$$c_i \sim \pi \qquad \text{Sample it's cluster assignment}$$
$$x_i \sim \mathcal{N}(x_i|\mu_{c_i}, \Sigma_{c_i}) \qquad \text{Sample it's value}$$

Joint probability:

$$\begin{aligned} P(x_1, \ldots, x_N, c_1, \ldots, c_N|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{N} \pi_{c_i}\mathcal{N}(x_i|\mu_{c_i}, \Sigma_{c_i}) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{K} \left[\pi_j\mathcal{N}(x_i|\mu_j, \Sigma_j)\right]^{\mathbb{1}(c_i=j)} \end{aligned}$$

Model-based clustering



Given observations $X = \{x_1, \ldots, x_N\}$, we face three problems:

- What are the c_i? (inference)
- What is π and $\theta_c = (\mu_c, \Sigma_c)$? (learning)
- What is K? (model selection, not covered here)

Learning

Imagine we had the cluster assignments C. We saw:

$$P(x_1,\ldots,x_N,c_1,\ldots,c_N|\pi,\boldsymbol{\mu},\boldsymbol{\Sigma}) = \prod_{i=1}^{N}\prod_{j=1}^{K}\left[\pi_j\mathcal{N}(x_i|\mu_j,\Sigma_j)\right]^{\mathbb{1}(c_i=j)}$$
$$= \left(\prod_{j=1}^{K}(\pi_j)^{N_j}\right)\left(\prod_{j=1}^{K}\prod_{\{i\ s.t.\ c_i=j\}}\mathcal{N}(x_i|\mu_j,\Sigma_j)\right)$$

Conveniently separates out into π and component parameters.

$$\log P(X,C|\pi,\boldsymbol{\mu},\boldsymbol{\Sigma}) = \left(\sum_{j=1}^{K}N_j\log\pi_j\right)\left(\sum_{j=1}^{K}\sum_{\{i\ s.t.\ c_i=j\}}\log\mathcal{N}(x_i|\mu_j,\Sigma_j)\right)$$

$$\log P(X, C|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\sum_{j=1}^{K} N_j \log \pi_j\right) \left(\sum_{j=1}^{K} \sum_{\{i \text{ s.t. } c_i=j\}} \log \mathcal{N}(x_i|\mu_j, \Sigma_j)\right)$$

MLE requires three sets of 'sufficient statistics':

- The number of observations assigned to each cluster (N_j).
- The empirical mean and mean-square of obs. in each cluster

$$\left(\frac{1}{N_j} \sum_{\{i \text{ s.t. } c_i=j\}} x_i, \frac{1}{N_j} \sum_{\{i \text{ s.t. } c_i=j\}} x_i x_i^\top\right)$$

k-means assigns obs. to clusters given parameters. Good idea?

Inference

k-means assigns obs. to clusters given parameters. Good idea?

For obs. x_i, what is the conditional probability over c_i?

$$P(c_i|x_i, \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{P(x_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

$$\propto P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{j=1}^{K} \left[\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j) \right]^{\mathbb{1}(c_i=j)} \right)$$

Inference

k-means assigns obs. to clusters given parameters. Good idea?

For obs. x_i, what is the conditional probability over c_i?

$$P(c_i|x_i, \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{P(x_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

$$\propto P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{j=1}^{K} \left[\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)\right]^{\mathbb{1}(c_i=j)} \right)$$

- proportional to prior probability of cluster j, π_j
- proportional to compatibility obs. i with parameters θ_j

k-means assigns obs. to clusters given parameters. Good idea?

For obs. x_i, what is the conditional probability over c_i?

$$P(c_i|x_i, \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{P(x_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

$$\propto P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{j=1}^{K} \left[\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j) \right]^{\mathbb{1}(c_i=j)} \right)$$

- proportional to prior probability of cluster j, π_j
- proportional to compatibility obs. i with parameters θ_j

Written as r_{ic}: 'responsibility' of cluster c for obs. i.

Inference

k-means assigns obs. to clusters given parameters. Good idea?

For obs. x_i, what is the conditional probability over c_i?

$$P(c_i|x_i, \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{P(x_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

$$\propto P(x_i, c_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{j=1}^{K} \left[\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j) \right]^{\mathbb{1}(c_i=j)} \right)$$

- proportional to prior probability of cluster j, π_j
- proportional to compatibility obs. i with parameters θ_j

Written as r_{ic}: 'responsibility' of cluster c for obs. i.

```
rr <- rep(0,K)
for(i in 1:K) rr[i] <- pi[i]*dmvnorm(x, mu[[i]],sigma[[i]])
rr <- rr / sum(rr);
```

The mixture of Gaussians

How do we update parameters given these probabilities?

$$\mu = \frac{\sum_{i=1}^{N} r_{ic} x_i}{\sum_{i=1}^{N} r_{ic}}$$

$$\Sigma + \mu\mu^\top = \frac{\sum_{i=1}^{N} r_{ic}\, x_i x_i^\top}{\sum_{i=1}^{N} r_{ic}}$$

$$\pi_c = \frac{1}{N} \sum_{i=1}^{N} r_{ic}$$

How do we update parameters given these probabilities?

$$\mu = \frac{\sum_{i=1}^{N} r_{ic} x_i}{\sum_{i=1}^{N} r_{ic}}$$

$$\Sigma + \mu\mu^\top = \frac{\sum_{i=1}^{N} r_{ic} \, x_i x_i^\top}{\sum_{i=1}^{N} r_{ic}}$$

$$\pi_c = \frac{1}{N} \sum_{i=1}^{N} r_{ic}$$

Compare with when we actually knew the cluster assignments.

The EM algorithm

Initialize parameters $\pi, \{(\mu_c, \Sigma_c)\}$ arbitrarily

Calculate the observation responsibilities r_{ic} given parameters

Update parameters given responsibilities

Repeat till convergence

Suprising fact: EM converges to stationary point of the log-likelihood:

$$\log P(X|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \sum_{C=\mathcal{C}} P(X, C|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

The EM algorithm

Suprising fact: EM converges to stationary point of the log-likelihood:

$$\log P(X|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \sum_{C=\mathcal{C}} P(X, C|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Can directly calculate gradients w.r.t. parameters and optimize.

Doable but messy:

The EM algorithm

Suprising fact: EM converges to stationary point of the log-likelihood:

$$\log P(X|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \sum_{C=\mathcal{C}} P(X, C|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Can directly calculate gradients w.r.t. parameters and optimize.

Doable but messy:

- Sums inside logarthms is inconvenient.
- Need to calculate gradients w.r.t. covariance matrices.
- Need to choose step sizes.