

LECTURE 9: THE EM (EXPECTATION-MAXIMIZATION) ALGORITHM

STAT 545: INTRO. TO COMPUTATIONAL STATISTICS

Vinayak Rao

Purdue University

September 16, 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 $X \equiv \{x_1, \dots, x_N\}$, the likelihood is

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

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 $X \equiv \{x_1, \dots, x_N\}$, 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.

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,

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

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,

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

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

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,

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

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

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

Holds for *exponential family distributions* (later)

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

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

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

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

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

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

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

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

Given data, what is MLE of π ?

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

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.

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.

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

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

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

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:

$$c_i \sim \pi$$

Sample it's cluster assignment

$$x_i \sim \mathcal{N}(x_i | \mu_{c_i}, \Sigma_{c_i})$$

Sample it's value

MIXTURE OF GAUSSIANS (MoG)

To generate the i th observation:

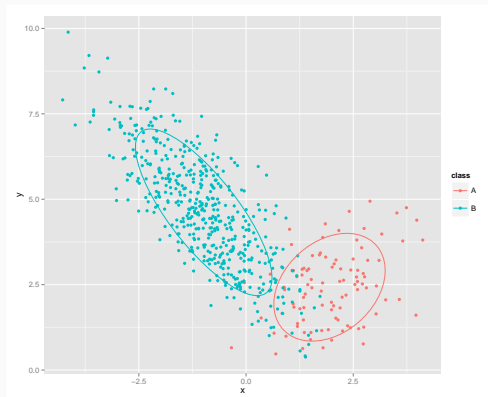
$c_i \sim \pi$ Sample it's cluster assignment

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

Joint probability:

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

MODEL-BASED CLUSTERING



Given observations $X = \{x_1, \dots, x_N\}$, 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)

Imagine we had the cluster assignments C . We saw:

$$\begin{aligned}
 P(X_1, \dots, X_N, C_1, \dots, C_N | \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^N \prod_{j=1}^K [\pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)]^{\mathbb{1}(c_i=j)} \\
 &= \left(\prod_{j=1}^K (\pi_j)^{N_j} \right) \left(\prod_{j=1}^K \prod_{\{i \text{ s.t. } c_i=j\}} \mathcal{N}(x_i | \mu_j, \Sigma_j) \right)
 \end{aligned}$$

Conveniently separates out into π and component parameters.

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

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

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?

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

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

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

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

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

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

- 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_K ?

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

- 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 .

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

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

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

- 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);
```


How do we update parameters given these probabilities?

$$\begin{aligned}\mu &= \frac{\sum_{i=1}^N r_{ic} x_i}{\sum_{i=1}^N r_{ic}} \\ \Sigma &= \frac{\sum_{i=1}^N r_{ic} x_i x_i^T}{\sum_{i=1}^N r_{ic}} \\ \pi_c &= \frac{1}{N} \sum_{i=1}^N r_{ic}\end{aligned}$$

How do we update parameters given these probabilities?

$$\begin{aligned}\mu &= \frac{\sum_{i=1}^N r_{ic} x_i}{\sum_{i=1}^N r_{ic}} \\ \Sigma &= \frac{\sum_{i=1}^N r_{ic} x_i x_i^T}{\sum_{i=1}^N r_{ic}} \\ \pi_c &= \frac{1}{N} \sum_{i=1}^N r_{ic}\end{aligned}$$

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:

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