

CS-667 Advanced Machine Learning

Nazar Khan

PUCIT

Gaussian Mixture Models

Gaussian Mixture Models

- ▶ We have already seen that multi-modal densities cannot be modelled via a uni-modal Gaussian.
- ▶ They can be modelled via *mixtures of Gaussians* which are simply *linear superpositions of uni-modal Gaussians*

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where the *mixing coefficients* π_k satisfy

$$0 \leq \pi_k \leq 1$$

$$\sum_{k=1}^K \pi_k = 1$$

- ▶ We will now derive the mixture density $p(\mathbf{x})$ using latent variables.

Gaussian Mixture Models

Latent Variable View

- ▶ Similar to the K-means approach, let us append our observed variable \mathbf{x} with a latent variable \mathbf{z} using 1-of- K coding.
- ▶ Using elementary probability

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$$

- ▶ However, this time we use probabilities (soft assignment)

$$p(z_k = 1) = \pi_k$$

- ▶ Due to the 1-of- K representation

$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$$

and conditional probability

$$p(\mathbf{x}|\mathbf{z}) = p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Mixture Models

Latent Variable View

- ▶ Therefore, the latent variable view also yields the Gaussian Mixture Model (GMM)

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- ▶ More importantly, complicated/multi-modal $p(\mathbf{x})$ has been modelled using simple/uni-modal $p(\mathbf{x}|\mathbf{z})$.
- ▶ This powerful idea extends beyond Gaussian mixtures.
 - ▶ Mixtures of insert_your_favourite distribution.
 - ▶ Mixtures of linear regression.
 - ▶ Mixtures of logistic regression.
 - ▶ ...

Gaussian Mixture Models

Responsibilities

- ▶ $p(\mathbf{x})$ is the *marginal density* that we are looking to model.
- ▶ $p(\mathbf{x}|z_k = 1)$ is the *component conditional density*. That is, probability density of \mathbf{x} according to component k .
- ▶ $p(z_k = 1)$ is the *prior probability* of component k .
- ▶ $p(z_k = 1|\mathbf{x})$ is the *posterior probability* of component k .
 - ▶ Can be viewed as the *responsibility* that component k takes for explaining observation \mathbf{x} .
 - ▶ Can be computed via Bayes' theorem

$$\begin{aligned} p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \end{aligned}$$

- ▶ We will denote responsibility by $r_k = p(z_k = 1|\mathbf{x})$.

Gaussian Mixture Models

Parameter Estimation

- ▶ The latent variable view of GMMs suggests iterative, alternating optimisation.
- ▶ Given i.i.d. data $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and an integer $K > 1$, find mixing coefficients $\{\pi_k\}$ and Gaussian parameters $\{\boldsymbol{\mu}_k\}$ and $\{\boldsymbol{\Sigma}_k\}$.
- ▶ Likelihood is given by $\prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- ▶ Log-likelihood is given by $\sum_{n=1}^N \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$.
 - ▶ Notice that the summation in the mixture model prevents the natural logarithm from cancelling out the Gaussian exponential. So, no closed form solution.
 - ▶ Solution 1: Gradient ascent.
 - ▶ Solution 2: Alternating optimisation.

Gaussian Mixture Models

Estimation of μ_k

- ▶ Using maximum likelihood

$$\begin{aligned} \mathbf{0} &\equiv \frac{\partial \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} \\ \implies \mathbf{0} &= - \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\underbrace{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}_{r_{nk}}} \boldsymbol{\Sigma}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \\ \implies \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}} = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{N_k} \end{aligned}$$

- ▶ Notice the similarity with K-means. The only difference here is that the hard assignments have been replaced by soft responsibilities.

Gaussian Mixture Models

Estimation of Σ_k

- ▶ Similarly

$$\mathbf{0} \equiv \frac{\partial \ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k}$$
$$\implies \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

- ▶ This is similar to the result for fitting a single Gaussian but now each data point is weighted by the responsibility r_{nk} .

Gaussian Mixture Models

Estimation of π_k

- ▶ Maximisation with respect to π_k is a constrained maximisation since π_k correspond to probability values.
- ▶ So we maximise the Lagrangian

$$L(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda) = \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right)$$

by setting the gradient to 0

$$\begin{aligned} 0 &\equiv \frac{\partial L(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \pi_k} \\ \implies 0 &= \sum_{n=1}^N \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda \end{aligned}$$

- ▶ Multiplying both sides by π_k and then summing both sides over k yields $\lambda = -N$.
- ▶ Substituting $\lambda = -N$ and rearranging yields

$$\pi_k = \frac{N_k}{N}$$

- ▶ In words, mixing coefficient for component k is given by the *average responsibility that it takes for explaining the training data points*.

Gaussian Mixture Models

Parameter Estimation

- ▶ Notice that solutions for μ_k , Σ_k and π_k are dependent on the responsibilities r_{nk} .
- ▶ However, the responsibilities depend on μ_k , Σ_k and π_k .
- ▶ We can now present the alternating optimisation algorithm for GMMs.

Alternating Optimisation for GMMs

Data: Data points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, integer $K > 1$.

Result: Component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, mixing coefficients $\{\pi_k\}$

1. Choose some initial values for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
2. Fix parameters, update responsibilities $r_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$
3. Fix responsibilities, update parameters

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{N_k} \text{ where } N_k = \sum_{n=1}^N r_{nk}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

Alternating Optimisation for GMMs

4. Evaluate log-likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

and check for convergence of either log-likelihood or parameters. If not converged, return to step 2.

Alternating Optimisation for GMMs

- ▶ Each iteration increases (or retains) the value of the log-likelihood.
- ▶ Therefore, convergence to (local) maximum is guaranteed.
- ▶ This algorithm has a name – *Expectation Maximisation (EM)*.
 - ▶ We will cover it in detail in next lecture.
- ▶ Converges slower than K-means and performs more computations per-iteration.
- ▶ Usually a good idea to initialise EM by the result of K-means.
 - ▶ Set μ_k to the k -th K-means cluster center.
 - ▶ Set Σ_k to the data covariance matrices for k -th K-means cluster.
 - ▶ Set π_k to the fraction of points assigned by K-means to cluster k .

Alternating Optimisation for GMMs

Singularity Avoidance

- ▶ Log-likelihood equals infinity if any Gaussian component 'collapses' to a training data point. (Why?)
- ▶ This represents a pathological condition or *singularity*.
- ▶ Care must be taken to check if that has happened or is close to happening.
- ▶ If so, the collapsing component should be reset to some other randomly chosen μ_k and large Σ_k and the optimisation should proceed as before.

Assignment 6

EM for Gaussian Mixture Model

- ▶ Density estimation via Gaussian Mixture Model (GMM).
 - ▶ Code up a generic implementation of learning a GMM via the EM algorithm in function
`[mixing_coefs, means, covariance_mats]=learn_gmm(X,K)`
where X is a $D \times N$ data matrix and K is the number of Gaussian components.
 - ▶ Regenerate Figure 9.8 in Bishop's book.
- ▶ Submit your `_roll_number_GMM.zip` containing
 - ▶ code,
 - ▶ generated image, and
 - ▶ report.txt/pdf explaining your results.
- ▶ Due Wednesday, May 16, 2018 before 5:30 pm on `\\printsrv`.