

Computer Vision Notes

Expectation Maximization and Latent Semantic Analysis

Faisal Z. Qureshi
<http://vclab.science.uoit.ca>

Faculty of Science
Ontario Tech University

December 22, 2025



Copyright information and license

© Faisal Z. Qureshi



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Finite Mixture Models

- ▶ $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$.
- ▶ Assumptions: points are drawn in an i.i.d. fashion from a density function $p(\mathbf{x})$, which is defined as a finite mixture model with K components:

$$p(\mathbf{x}|\theta) = \sum_{k=1}^K \alpha_k p_k(\mathbf{x}|z_k, \theta_k)$$

Here

- ▶ $p_k(\mathbf{x}|z_k, \theta_k)$ are mixture components. Each distribution is parameters θ_k .
- ▶ $\alpha_k = p(z_k)$ are mixture weights. These represent the probability that a random sample \mathbf{x} was generated by component k . Note that $\sum_{k=1}^K \alpha_k = 1$.
- ▶ $z = (z_1, z_2, \dots, z_K)$ is a vector of K binary indicators z_k . Only one of these can be non-zero.

Generating Data from a Finite Mixture Model

- ▶ Pick a model k with probability α_k
- ▶ Sample a data from that model
- ▶ Repeat

Likelihood of Mixture Model

Under the i.i.d. assumption

$$L = \prod_{i=1}^N \sum_{k=1}^K \alpha_k P_k(\mathbf{x}^{(i)} | z_k, \theta_k)$$

Log likelihood

$$\log L = \sum_{i=1}^N \log \sum_{k=1}^K \alpha_k P_k(\mathbf{x}^{(i)} | z_k, \theta_k)$$

Parameters for finite mixture models with K components are

$$\Theta = \{\alpha_1, \dots, \alpha_K, \theta_1, \dots, \theta_K\}.$$

Model fitting

How do we fit our model to data using the maximum likelihood principle?

Membership weights

Recall Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)}$$

The membership weight of data point $\mathbf{x}^{(i)}$ for component k given parameter Θ :

$$w_{ik} = p(z_{ik} = 1 | \mathbf{x}^{(i)}, \Theta) = \frac{\alpha_k p_k(\mathbf{x}^{(i)} | z_k, \theta_k)}{\sum_{m=1}^k \alpha_m p_m(\mathbf{x}^{(i)} | z_m, \theta_m)}$$

The membership weights capture our uncertainty about which k components generated the sample $\mathbf{x}^{(i)}$. We continue to assume that $\mathbf{x}^{(i)}$ is generated by a single component. Meaning these don't imply that the $\mathbf{x}^{(i)}$ is a result of weighted sum of K samples generated by K components.

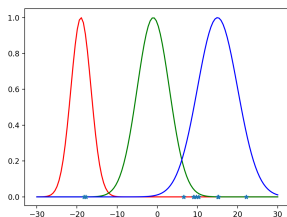
Gaussian Mixture Model (GMM)

We set

$$p(\mathbf{x}|\theta) = \sum_{k=1}^K \alpha_k \mathcal{N}(\theta_k)$$

where $\theta_k = (\mu_k, \Sigma_k)$ and

$$\mathcal{N}(\theta_k) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1} (\mathbf{x}-\mu_k)}.$$



Generating Data from a GMM (1D)

```
mixture_of_gaussians = [(-19, 2.5, 0.2, 'r'),  
                        (-1, 4, 0.3, 'g'),  
                        (15, 5, 0.5, 'b')]
```

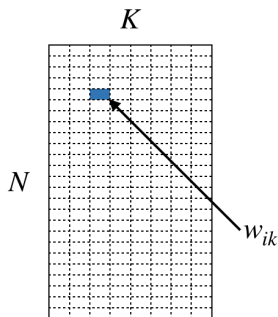
```
K = len(mixture_of_gaussians)  
a = np.empty(K)  
a = [mixture_of_gaussians[k][2] for k in range(K)]
```

```
N = 10  
x = np.empty(N)  
for i in range(N):  
    k = np.random.choice(K, 1, a)[0]  
    mu, sig, _ = mixture_of_gaussians[k]  
    x[i] = np.random.normal(mu, sig, 1)
```

EM for GMM

E-Step: Given current parameter values Θ , compute w_{ik} for each data point - mixture component pair.

$$w_{ik} = p(z_{ik} = 1 | \mathbf{x}^{(i)}, \Theta) = \frac{\alpha_k \mathcal{N}(\mathbf{x}^{(i)}; \theta_k)}{\sum_{m=1}^k \alpha_m \mathcal{N}(\mathbf{x}^{(i)}; \theta_m)}.$$



EM for GMM

M-Step: Use membership weights and data to calculate the new parameter Θ .

- ▶ The effective number of data points attached to component k are $\alpha_k^{\text{new}} = \sum_{i=1}^N w_{ik}/N$.
- ▶ Compute new means

$$\mu_k^{\text{new}} = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} \mathbf{x}^{(i)}.$$

- ▶ Compute new covariance

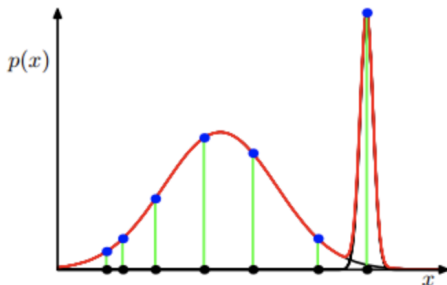
$$\Sigma_k^{\text{new}} = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} (\mathbf{x}^{(i)} - \mu_k^{\text{new}})(\mathbf{x}^{(i)} - \mu_k^{\text{new}})^T.$$

- ▶ Mean and covariance is computed similar to how we would compute these quantities empirically.

EM for GMM: Practical Considerations

When using GMM, EM can often lead to singularities. These occur when a Gaussian begins to account for a single data point. In this case variance goes to zero, resulting in an overfitted model. This case doesn't arise in situations where we fit a single Gaussian to the data. (Why?)

One way to solve such singularities is to reset the mean and the variance of the culprit Gaussian. Specifically, set the new mean to be a well-behaved local maxima of the likelihood function. Set the variance to some large value.



Courtesy: Pattern Recognition

Mean and Variance (1D)

Consider 1D data $\{x^{(1)}, \dots, x^{(N)}\}$

Mean

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x^{(i)}$$

Variance

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x^{(i)} - \langle x \rangle)^2$$

Mean and Covariance

Consider 1D data $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$, $\mathbf{x}^{(i)} \in \mathbb{R}^D$

Mean

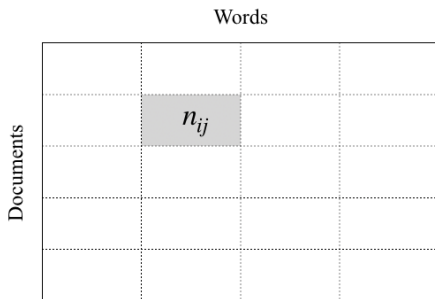
$$\langle \mathbf{x} \rangle = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}, \langle \mathbf{x} \rangle \in \mathbb{R}^D$$

Variance

$$\Sigma = \frac{1}{N} (\mathbf{x}^{(i)} - \langle \mathbf{x} \rangle)(\mathbf{x}^{(i)} - \langle \mathbf{x} \rangle), \Sigma \in \mathbb{R}^{D \times D}$$

Probabilistic Latent Semantic Analysis (pLSA)

- ▶ Documents $D = \{d_1, \dots, d_N\}$
- ▶ Vocabulary $W = \{w_1, \dots, w_M\}$
- ▶ Co-occurrence table of counts $n_{ij} = \text{count}(d_i, w_j)$, which captures how many times word $w^{(j)}$ occurs in document $d^{(i)}$.
- ▶ Associate an unobserved variable $z_k \in \{z_1, \dots, z_K\}$ with each pair $\langle d_i, w_j \rangle$. We will refer to these as *topics*.

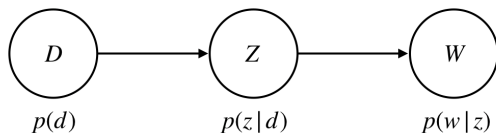


Generative model for pair $\langle d_i, w_j \rangle$

- ▶ Select a document d_i with probability $p(d_i)$
- ▶ Pick a topic z_k with probability $p(z_k|d_i)$
- ▶ Generate a word w_j with probability $p(w_j|z_k)$

We can write the probability of pair $\langle d_i, w_j \rangle$ as follows

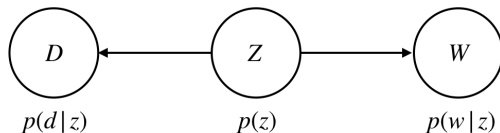
$$p(d_i, w_j) = \sum_{k=1}^K p(w_j|z_k)p(z_k|d_i)p(d_i)$$



Implicit conditional independence assumption

d_i and w_j are independent conditioned on the state of the associated latent variable z_k .

$$p(d_i, w_j) = \sum_{k=1}^K p(w_j|z_k)p(d_i|z_k)p(z_k)$$



Likelihood

Under i.i.d. assumption, we write likelihood as follows

$$L = \prod_{i=1}^N \prod_{j=1}^M p(d_i, w_j)^{n_{ij}}$$

Log-likelihood

$$\log L = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \log p(d_i, w_j)$$

Learn the hidden variables z_k in a maximum likelihood fashion.

Likelihood

$$\begin{aligned}\log L &= \sum_{i=1}^N \sum_{j=1}^M n_{ij} \log p(d_i, w_j) \\ &= \sum_{i=1}^N \sum_{j=1}^M n_{ij} \log \left[\sum_{k=1}^K p(w_j|z_k)p(z_k|d_i)p(d_i) \right] \\ &= \sum_{i=1}^N \sum_{j=1}^M n_{ij} \log \left[p(d_i) \sum_{k=1}^K p(w_j|z_k)p(z_k|d_i) \right] \\ &= \sum_{i=1}^N \sum_{j=1}^M n_{ij} \left[\log p(d_i) + \log \sum_{k=1}^K p(w_j|z_k)p(z_k|d_i) \right] \\ &= \sum_{i=1}^N \sum_{j=1}^M n_{ij} \log p(d_i) + n_{ij} \left[\log \sum_{k=1}^K p(w_j|z_k)p(z_k|d_i) \right]\end{aligned}$$

Likelihood

continued from last slide

$$\log L = \sum_{i=1}^N n_i \log p(d_i) + \sum_{j=1}^M n_{ij} \left[\log \sum_{k=1}^K p(w_j | z_k) p(z_k | d_i) \right]$$

Expectation maximization

E-Step calculates posterior probabilities for latent variables given the observations by using the current estimates of the parameters.

$$\begin{aligned} p(z_k | d_i, w_j) &= \frac{p(w_j, z_k | d_i)}{p(w_j | d_i)} \\ &= \frac{p(w_j | z_k, d_i) p(z_k | d_i)}{p(w_j | d_i)} \\ &= \frac{p(w_j | z_k) p(z_k | d_i)}{\sum_{l=1}^K p(w_j | z_l, d_i) p(z_l | d_i)} \end{aligned}$$

Expectation maximization

E-Step *continued*

In order to maximize $\log L$, set

$$p(w_j|z_k) = \frac{\sum_{i=1}^N n_{ij} p(z_k|d_i, w_j)}{\sum_{l=1}^M \sum_{i=1}^N w_{il} p(z_k|d_i, w_l)}$$

and

$$p(z_k|d_i) = \frac{\sum_{j=1}^M n_{ij} p(z_k|d_i, w_j)}{\text{count}(d_i)}$$

M-Step Given current estimates of z_k , compute $p(d_i)$, n_i , and n_{ij} . n_i refers to the number of words in document d_i .

These quantities can be easily computed using the available data.

Expectation maximization

- ▶ Repeat E-Step and M-Step until convergence.
- ▶ Bag of Words Model: each document is represented using the frequency of word occurrences. The relative ordering of the words is ignored.

Applications of pLSA

- ▶ Topic detection
- ▶ Image classification
- ▶ Action classification

Useful Logarithm Identities

Definition $\log_a(b) = r$ iff $a^r = b$.

Exponent $\log_a x^n = n \log_a x$

By definition $\log_a(x) = r$ iff $a^r = x$. Let $(a^r)^n = x^n$ then implies that $a^{rn} = x^n$. Then from definition $\log_a x^n = rn$. Hence $\log_a x^n = n \log_a x$.

Multiplication $\log_a(xy) = \log_a x + \log_a y$

Division

$$\log_a \left(\frac{x}{y} \right) = \log_a x - \log_a y$$

Change of base

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

Useful Logarithm Identities

One $\log_a(1) = 0$

Base-10 We often use the shortcut \log to indicate \log_{10}

▶ $\log(10^x) = x$

▶ $10^{\ln(x)} = x$

Base-e $\ln = \log_e$

▶ $\ln(e^x) = x$

▶ $e^{\ln(x)} = x$

Caveat \ln always denotes logarithm base e . \log is ambiguous. In computer science, it often refers to logarithm base e or logarithm base 2 depending upon the context. In many situations the logarithm base does not matter. E.g., when doing complexity analysis. It is because different logarithms are simply scalar multiple of each other.