Generative Models, Maximum Likelihood, Soft Clustering, and Expectation–Maximization

Aris Anagnostopoulos

We will see why we design models for data, how to learn their parameters, and how by considering a mixture model of Gaussians can give us an algorithm for soft clustering. I have included many of the derivations that we omitted in class for those that are interested in the details; there are a few formulae, but the underlying ideas are simple. Feel free to skip the calculations of Section 5 if you'd like, but make sure you understand the main ideas.

1 Modeling

An important part of understanding our data is that of modeling. *Modeling* is generally a way to capture a phenomenon in the real world. There are several different ways to model a phenomenon, and there are probabilistic, game-theoretic, agent-based, statistical, and other types of models. Here we will see some examples of statistical models, which can also be seen as generative models. A good model has a few desired properties:

- It creates data that have similar statistical properties with the ones observed in reality.
- The generation process seems natural and corresponding in some sense to what actually happens in reality.
- It is amenable to analysis.

These properties are rather vague and not always possible. For instance, very often a model to create data that are realistic it has to become complicated; this means that it will probably be hard to analyze it. Thus, we often consider models that replicate only *some* of the characteristics of the underlying data, precisely the characteristics that we are interested in understanding in the given time. This is why the whole process of modeling is usually hard, requires experience, and it is often both a science and an art.

Why are we interested in designing models for our data? First, designing a good model can help us *making sense* of our data, and the process with which the data are created. For instance, the model of Watts and Strogatz [2], even though it is not realistic, can help us explain the *small-world phenomenon* in networks. The Barabási–Albert preferential-attachment model [1], can explain the power law observed in the degree distribution.

Second, a good model can help us *make predictions*. Assume that we have a history of data, say 100 days of transportation information. What we will often do is design a model (i.e., come up with a family of models, which are characterized by a set of *parameters*) and use the largest part of the past data, say 90 days, to *train* the model, that is, find the values of of the parameters of the model. Then using the part of the data that was left out we will evaluate the model: we will see if it can create data for the 10 days that are somehow similar to what the real data are in these 10 days. Or we will check what probability does the model give to the real data in these 10 days. This process (which is often iterative, and usually more complicated than what was just described) allows us to evaluate the model and see whether the family of the models that we have chosen and the parameters that we have estimated are proper. Assuming that this is the case, then we can use the model to make predictions for the future: if the model that was trained in

90 days in the past is able to predict the last 10 days, then we assume (hope!) that it can predict the next 10 days as well. This of course requires several assumptions for our real-life scenarios.

Taking this one step further, we can now *take decisions* for the future. For instance, if the model predicts that we will have a lot of people wanting to go from Colosseum to San Peter's Cathedral, we can increase the number of buses that make this route. Or, if a particular flu strain is predicted to infect a lot of people on February, a country may try to increase its stoc of flu vaccines and publicize vaccination.

Finally, a model can also be used to *design tools*. For instance, we will see that *PageRank*, the algorithm first used by Google (along with many other features) to assign scores to web pages, is actually the value obtained in the *random surfer model*: In this model, we consider a user who visits pages and follows outgoing links randomly, whereas with some probability he stops and restarts browsing at a random page; then the pagerank score of a given web page p is the frequency, in the long term, that this user visit page p.

In the next section we will see some of the above concepts by considering a very simple model for modeling the height of a part of the population. This will bring us to the question of how we can learn the parameters of the model using maximum likelihood. Then, by generalizing this example and including heights of people from different parts of the population, we will introduce a more complicated model, which, in the process of learning its parameters will also allow us to perform soft clustering of the underlying data.

2 A simple Gaussian model

To become familiar with some of the concepts of modeling, let us assume that we observe the heights of n Italian women. Let x_i be the height of the ith person. A reasonable model for such a population is the Gaussian model: we assume that each x_i is drawn from some Gaussian distribution with mean μ and variance σ^2, each draw being independent. We can think that when "God" decides the height of the ith person, he picks a value x_i distributed according to a normal distribution $\mathcal{N}(\mu, \sigma^2)$. Define the entire set of data to be the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

Recall that the *Gaussian* or *normal distribution*, symbolized by $\mathcal{N}(\mu, \sigma^2)$, is the continuous probability distribution with range $\mathbb{R}$ and probability density function given by

$$p(x) = \mathcal{N}(x; \mu, \sigma^2) \triangleq \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}},$$

where the meaning of $\triangleq$ is that we define $\mathcal{N}(x; \mu, \sigma^2)$ to be the expression on the right. The expected value of the distribution is μ and its variance is σ^2.

> Recall that when we draw a value from a continuous probability distribution (a distribution whose *cumulative distribution function* (CDF) is continuous), the probability that we obtain a particular value x is 0. Thus, we can characterize the distribution by its *probability density function* (PDF), which can give us the probability that we obtain a value in a given interval $[a, b]$. The probability that we obtain a value in a very small interval δx around x is approximately equal to $p(x)\delta x$, and so—using the same approach that we use when we define integrals—the probability that we obtain a value in the range $[a, b]$ is
>
> $$\mathbf{Pr}\big(x \in [a, b]\big) = \int_a^b p(x)\,\mathrm{d}x.$$
>
> Nevertheless, it is often easy to think of $p(x)$ as the likelihood that we will obtain the value x, and many of the properties that hold when we talk about probabilities of discrete random variables hold also for PDFs. For instance, if two random variables X, Y with a *joint PDF* $p_{X,Y}(x, y)$ are independent, then we have that
>
> $$p_{X,Y}(x, y) = p_X(x)p_Y(y),$$
>
> where $p_X(x)$ and $p_Y(y)$ are the PDFs of the random variables X and Y, respectively.

We call the values μ and σ^2 the *parameters* of our model. Often we denote the collection of all our parameters with the vector $\boldsymbol{\theta}$, so here we have $\boldsymbol{\theta} = (\mu, \sigma^2)$.

Given the data and the model, a natural question is "what are the best set of parameters for the data that we have?". This is often called *fitting the model to the data*. There are different ways to define *best*, and in the next section we will see one of the most natural and most commonly used.

3 Maximum Likelihood

Assuming that the data came from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, the likelihood that we observe point x_i is

$$p_{\boldsymbol{\theta}}(x_i) = \mathcal{N}(x_i; \mu, \sigma^2) = \frac{e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}},$$

where we have put as a subscript $\boldsymbol{\theta}$ to make explicit the dependence on the parameters $\boldsymbol{\theta} = (\mu, \sigma^2)$.

Assuming also that all the x_is are jointly independent, we have that the likelihood that we observe the values $x_1, x_2, \ldots, x_n$ is

$$\prod_{i=1}^{n} p_{\boldsymbol{\theta}}(x_i) = \prod_{i=1}^{n} \frac{e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}.$$

Note that this expression is a function of the data $\mathbf{x} = (x_1, \ldots, x_n)$ and of the parameters of the model $\boldsymbol{\theta} = (\mu, \sigma^2)$. Because the data are given and because we want to find the best value for $\boldsymbol{\theta}$, we can see it as a function of $\boldsymbol{\theta}$. We thus, define the *likelihood function* $\mathrm{L}(\boldsymbol{\theta}; \mathbf{x})$ as

$$\mathrm{L}(\boldsymbol{\theta}; \mathbf{x}) = \mathrm{L}((\mu, \sigma^2); \mathbf{x}) \stackrel{\triangle}{=} \prod_{i=1}^{n} p_{\boldsymbol{\theta}}(x_i).$$

Therefore, the likelihood function tells us what is the likelihood that we observe the data for a given set of model parameters $\boldsymbol{\theta}$.

The *maximum-likelihood* approach to finding the best model is to find the value $\hat{\boldsymbol{\theta}} = (\hat{\mu}, \hat{\sigma}^2)$ that maximizes $\mathrm{L}(\boldsymbol{\theta}; \mathbf{x})$. In some sense it is the most probable model for the given data, among all the models of the family that we have selected (here, a Gaussian distribution).

It turns out that instead of maximizing $\mathrm{L}(\boldsymbol{\theta}; \mathbf{x})$, it is more convenient to maximize its logarithm. Thus, we define the *log-likelihood function*

$$\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x}) \stackrel{\triangle}{=} \ln(\mathrm{L}(\boldsymbol{\theta}; \mathbf{x})),$$

and we try to maximize this one. Because $\ln(\cdot)$ is an increasing function, the values $\hat{\boldsymbol{\theta}}$ that maximize $\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x})$ are the values that maximize $\mathrm{L}(\boldsymbol{\theta}; \mathbf{x})$. In our case we have:

$$\begin{aligned}
\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x}) &= \ln\left(\prod_{i=1}^{n} \frac{e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\right) \\
&= \sum_{i=1}^{n} \ln\left(\frac{e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\right) \\
&= \sum_{i=1}^{n} \left(-\frac{1}{2}\ln(2\pi\sigma^2) + \ln\left(e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)\right) \\
&= \sum_{i=1}^{n} \left(-\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln\sigma^2 - \frac{(x_i-\mu)^2}{2\sigma^2}\right) \\
&= -\sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} - \frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln(2\pi).
\end{aligned}$$

To find where $\mathrm{LL}((\mu, \sigma^2); \mathbf{x})$ attains its maximum, we set the partial derivatives with respect to μ and σ^2 equal to 0. Then we obtain

$$\frac{\partial LL}{\partial \mu} = 0,$$

or, equivalently,

$$\begin{aligned}
-\sum_{i=1}^{n} \tfrac{\partial}{\partial \mu} \frac{(x_i - \mu)^2}{2\sigma^2} &= 0 \\
\sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2} &= 0 \\
\sum_{i=1}^{n} x_i - n\mu &= 0 \\
\mu &= \frac{1}{n} \sum_{i=1}^{n} x_i,
\end{aligned}$$

and

$$\frac{\partial LL}{\partial \sigma^2} = 0,$$

or,

$$\begin{aligned}
-\sum_{i=1}^{n} \tfrac{\partial}{\partial \sigma^2} \frac{(x_i - \mu)^2}{2\sigma^2} - \tfrac{\partial}{\partial \sigma^2} \frac{n}{2} \ln \sigma^2 &= 0 \\
\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2(\sigma^2)^2} - \frac{n}{2\sigma^2} &= 0 \\
\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^2} - n &= 0 \\
\sigma^2 &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2.
\end{aligned}$$

These give what we expect, that the best values for the model's mean

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and variance

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

equal the sample mean and variance, respectively. (Actually, to show that it is a maximum and not a minimum or a saddle point, we have to check that the determinant of the Hessian matrix is positive and so on; check your multi-variable calculus books, if interested.)

4 Gaussian Mixture Model and Soft Clustering

Now we will see how, using the ideas of the previous section, we can soft cluster a dataset. When we *soft cluster* a dataset, instead of assigning each point to a cluster, we give to each point for every cluster a probability that it belongs to that cluster.

Assume that we have a population of n women who are either from Italy or from China, and whose weights are given, as before, by a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. We would like to try to cluster them based on their height. We will do this by using a more complicated model for generating the data, which will be a *mixture of Gaussian distributions*. The data-generation process for a height x_i has two steps:

1. We flip a biased coin and with probability π_{I} we choose to have an Italian woman and with probability π_{C} a Chinese one.

2. If the coin selected an Italian woman, then we choose the height x_i distributed as $\mathcal{N}(\mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)$. If the coin selected a Chinese one, then we choose the height x_i distributed as $\mathcal{N}(\mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)$.

All the choices are mutually independent. In the above model we have six parameters $\boldsymbol{\theta} = \{\pi_{\mathrm{I}}, \pi_{\mathrm{C}}, \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2, \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2\}$, and we have the constraint that $\pi_{\mathrm{I}} + \pi_{\mathrm{C}} = 1$, so that the coin flip at the first step be properly defined.

Given this model, it is natural to ask the same question as before: what is the best value of $\boldsymbol{\theta}$? We will try to apply again the maximum-likelihood approach. For this it is convenient to define the hidden or *latent* random variable Z_i as

$$Z_i = \begin{cases} \mathbf{I} & \text{if the } i\text{th person is Italian, and} \\ \mathbf{C} & \text{if the } i\text{th person is Chinese.} \end{cases}$$

We have

$$\begin{aligned} p_{\boldsymbol{\theta}}(x_i) &= \mathbf{Pr}(Z_i = \mathbf{I}) \cdot p_{\boldsymbol{\theta}}(x_i \mid Z_i = \mathbf{I}) + \mathbf{Pr}(Z_i = \mathbf{C}) \cdot p_{\boldsymbol{\theta}}(x_i \mid Z_i = \mathbf{C}) \\ &= \pi_{\mathrm{I}} \cdot \frac{e^{-\frac{(x_i - \mu_{\mathrm{I}})^2}{2\sigma_{\mathrm{I}}^2}}}{\sqrt{2\pi\sigma_{\mathrm{I}}^2}} + \pi_{\mathrm{C}} \cdot \frac{e^{-\frac{(x_i - \mu_{\mathrm{C}})^2}{2\sigma_{\mathrm{C}}^2}}}{\sqrt{2\pi\sigma_{\mathrm{C}}^2}} \\ &= \pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2). \end{aligned}$$

As before, using the fact that the different x_is are independent, we have that the likelihood function is

$$\mathrm{L}(\boldsymbol{\theta}; \mathbf{x}) = \prod_{i=1}^{n} \left(\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2) \right),$$

and we can also consider the log-likelihood function,

$$\begin{aligned} \mathrm{LL}(\boldsymbol{\theta}; \mathbf{x}) &= \sum_{i=1}^{n} \ln \left(\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2) \right) \\ &= \sum_{i=1}^{n} \ln \left(\pi_{\mathrm{I}} \cdot \frac{e^{-\frac{(x_i - \mu_{\mathrm{I}})^2}{2\sigma_{\mathrm{I}}^2}}}{\sqrt{2\pi\sigma_{\mathrm{I}}^2}} + \pi_{\mathrm{C}} \cdot \frac{e^{-\frac{(x_i - \mu_{\mathrm{C}})^2}{2\sigma_{\mathrm{C}}^2}}}{\sqrt{2\pi\sigma_{\mathrm{C}}^2}} \right). \end{aligned} \tag{1}$$

This is much harder to analyze than before because the sum prevents the logarithm to be applied to the exponentials and we cannot obtain a closed formula. In principle we can try to maximize it numerically, and, indeed, this is the approach used sometimes. In the next section we will introduce an alternative method, which can be used with mixture models and more generally when we have to infer the parameters of a model that contains latent variables.

5 Expectation–Maximization for Gaussian Mixtures and Soft Clustering

As we will see, we will be able to infer the values of the parameters with an algorithm similar to k-means. For those interested in the details I have included the calculations. They are actually a bit pedantic, but if you follow them slowly not too hard. Nevertheless, feel free to skip them if you want but make sure that you understand the high-level idea.

But first, let us assume that we knew the parameter vector $\boldsymbol{\theta}$. We will now see how, using these values, we could create a soft clustering. This means, that for every value x_i we will compute probabilities $\mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i)$ and $\mathbf{Pr}(Z_i = \mathbf{C} \,|\, X_i = x_i)$, which represent the probability that the ith person is Italian and Chinese, respectively, conditional that their weight is x_i. (We define the random variable X_i to be the height of the ith person when we create the data with our model.) Note that we condition on an event with probability 0, which, even though we have not defined it for discrete probability spaces, is fine but needs some technical work. Nevertheless, by some hand-waving arguments we can chose a very small value $\delta > 0$ and we have:

$$\begin{aligned}
\mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i) &\overset{\text{(a)}}{\approx} \mathbf{Pr}(Z_i = \mathbf{I} \,|\, x_i - \delta \le X_i \le x_i + \delta) \\
&= \frac{\mathbf{Pr}(Z_i = \mathbf{I},\ x_i - \delta \le X_i \le x_i + \delta)}{\mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta)} \\
&= \frac{\mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta \,|\, Z_i = \mathbf{I}) \cdot \mathbf{Pr}(Z_i = \mathbf{I})}{\mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta)} \\
&= \frac{\mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta \,|\, Z_i = \mathbf{I}) \cdot \mathbf{Pr}(Z_i = \mathbf{I})}{\mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta \,|\, Z_i = \mathbf{I}) \cdot \mathbf{Pr}(Z_i = \mathbf{I}) + \mathbf{Pr}(x_i - \delta \le X_i \le x_i + \delta \,|\, Z_i = \mathbf{C}) \cdot \mathbf{Pr}(Z_i = \mathbf{C})} \\
&\overset{\text{(b)}}{\approx} \frac{2\delta \cdot p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \mathbf{Pr}(Z_i = \mathbf{I})}{2\delta \cdot p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \mathbf{Pr}(Z_i = \mathbf{I}) + 2\delta \cdot p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{C}) \cdot \mathbf{Pr}(Z_i = \mathbf{I})} \\
&= \frac{p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \pi_{\mathrm{I}}}{p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \pi_{\mathrm{I}} + p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{C}) \cdot \pi_{\mathrm{C}}},
\end{aligned}$$

where in (a) we approximate the probability that $X_i = x_i$ with the probability that X_i falls in a tiny interval of length 2δ around x_i, in (b) we approximate this probability with 2δ times its PDF at the center of the interval, whereas in the equations in-between we use the definition of conditional expectation. (We are, essentially, proving Bayes' rule.) Actually, by taking the limit $\delta \to 0$, the above approximations hold with equality, and so we have

$$\begin{aligned}
\gamma_{i,\mathrm{I}} \triangleq \mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i) &= \frac{p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \pi_{\mathrm{I}}}{p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) \cdot \pi_{\mathrm{I}} + p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{C}) \cdot \pi_{\mathrm{C}}} \\
&= \frac{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)},
\end{aligned} \tag{2}$$

because, assuming that the person is Italian, her height is distributed according to $\mathcal{N}(\mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)$:

$$p_{\boldsymbol{\theta}}(x_i \,|\, Z_i = \mathbf{I}) = \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) = \frac{e^{-\frac{(x_i - \mu_{\mathrm{I}})^2}{2\sigma_{\mathrm{I}}^2}}}{\sqrt{2\pi\sigma_{\mathrm{I}}^2}}.$$

Notice that if we know $\boldsymbol{\theta}$ we can compute exactly the probability $\gamma_{i,\mathrm{I}} = \mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i)$.

Likewise, we can compute

$$\gamma_{i,\mathrm{C}} \triangleq \mathbf{Pr}(Z_i = \mathbf{C} \,|\, X_i = x_i) = \frac{\pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}. \tag{3}$$

Therefore, if we fix the parameters of the model $\boldsymbol{\theta}$, we can assign a probability for height x_i to be an Italian or a Chinese person, and this assignment induces a soft clustering.

Now we will look at the inverse question: Assume that we know the probabilities $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$. How can we compute the best value $\boldsymbol{\theta}$? We will try to maximize the log-likelihood function of Equation (1) as we did in Section 3. First we set the partial derivative with respect to μ_I equal to 0:

$$\begin{aligned}
0 &= \tfrac{\partial LL}{\partial \mu_\mathrm{I}} \\
&= \sum_{i=1}^{n} \frac{\pi_\mathrm{I} \cdot \mathcal{N}(x_i; \mu_\mathrm{I}, \sigma_\mathrm{I}^2)}{\pi_\mathrm{I} \cdot \mathcal{N}(x_i; \mu_\mathrm{I}, \sigma_\mathrm{I}^2) + \pi_\mathrm{C} \cdot \mathcal{N}(x_i; \mu_\mathrm{C}, \sigma_\mathrm{C}^2)} \cdot \frac{x_i - \mu_\mathrm{I}}{\sigma_\mathrm{I}^2} \\
&= \sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot \frac{x_i - \mu_\mathrm{I}}{\sigma_\mathrm{I}^2},
\end{aligned}$$

which gives

$$\mu_\mathrm{I} = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} x_i}{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}}}. \tag{4}$$

Similarly we get

$$\mu_\mathrm{C} = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} x_i}{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}}}. \tag{5}$$

Setting the derivative with respect to σ_I^2 equal to 0 we obtain:

$$\begin{aligned}
0 &= \tfrac{\partial LL}{\partial \sigma_\mathrm{I}^2} \\
&= \sum_{i=1}^{n} \frac{1}{\pi_\mathrm{I} \cdot \mathcal{N}(x_i; \mu_\mathrm{I}, \sigma_\mathrm{I}^2) + \pi_\mathrm{C} \cdot \mathcal{N}(x_i; \mu_\mathrm{C}, \sigma_\mathrm{C}^2)} \cdot \frac{\pi_\mathrm{I}}{2\pi\sigma_\mathrm{I}^2} \left(e^{-\frac{(x_i-\mu_\mathrm{I})^2}{2\sigma_\mathrm{I}^2}} (-(x_i-\mu_\mathrm{I})^2)(-\frac{1}{2(\sigma_\mathrm{I}^2)^2})\sqrt{2\pi\sigma_\mathrm{I}^2} - e^{-\frac{(x_i-\mu_\mathrm{I})^2}{2\sigma_\mathrm{I}^2}} \frac{\sqrt{2\pi}}{2\sqrt{\sigma_\mathrm{I}^2}} \right) \\
&= \sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot \frac{\sqrt{2\pi}}{4\pi\sigma_\mathrm{I}^2} \left((x_i - \mu_\mathrm{I})^2 \frac{1}{(\sigma_\mathrm{I}^2)^2} \sqrt{\sigma_\mathrm{I}^2} - \frac{1}{\sqrt{\sigma_\mathrm{I}^2}} \right) \\
&= \sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot \frac{\sqrt{2\pi}\sqrt{\sigma_\mathrm{I}^2}}{4\pi\sigma_\mathrm{I}^4} \left((x_i - \mu_\mathrm{I})^2 \frac{1}{\sigma_\mathrm{I}^2} - 1 \right),
\end{aligned}$$

which gives

$$\sigma_\mathrm{I}^2 = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot (x_i - \mu_\mathrm{I})^2}{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}}} \tag{6}$$

and, similarly,

$$\sigma_\mathrm{C}^2 = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} \cdot (x_i - \mu_\mathrm{C})^2}{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}}}. \tag{7}$$

Finally, because of the constraint $\pi_\mathrm{I} + \pi_\mathrm{C} = 1$, to maximize $\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x})$ with respect to π_I we will use the technique of Lagrange multipliers (check your optimization books for details, if interested). We add the constraint to the objective multiplied by λ and we try to maximize

$$\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x}) + \lambda(\pi_\mathrm{I} + \pi_\mathrm{C} - 1).$$

Setting the derivative with respect to π_I equal to 0 we get:

$$\begin{aligned}
0 &= \tfrac{\partial}{\partial \pi_\mathrm{I}} \big(\mathrm{LL}(\boldsymbol{\theta}; \mathbf{x}) + \lambda(\pi_\mathrm{I} + \pi_\mathrm{C} - 1) \big) \\
&= \sum_{i=1}^{n} \frac{\mathcal{N}(x_i; \mu_\mathrm{I}, \sigma_\mathrm{I}^2)}{\pi_\mathrm{I} \cdot \mathcal{N}(x_i; \mu_\mathrm{I}, \sigma_\mathrm{I}^2) + \pi_\mathrm{C} \cdot \mathcal{N}(x_i; \mu_\mathrm{C}, \sigma_\mathrm{C}^2)} + \lambda.
\end{aligned}$$

To compute the value λ we will use the constraint $\pi_{\mathrm{I}} + \pi_{\mathrm{C}} = 1$. Multiplying the above by π_{I} gives

$$\lambda \cdot \pi_{\mathrm{I}} = -\sum_{i=1}^{n} \pi_{\mathrm{I}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} = -\sum_{i=1}^{n} \gamma_{i,\mathrm{I}}. \tag{8}$$

With the same approach we have

$$\lambda \cdot \pi_{\mathrm{C}} = -\sum_{i=1}^{n} \pi_{\mathrm{C}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} = -\sum_{i=1}^{n} \gamma_{i,\mathrm{C}}. \tag{9}$$

Summing the two constraints and using that $\pi_{\mathrm{I}} + \pi_{\mathrm{C}} = 1$ gives

$$\begin{aligned}
\lambda &= -\left(\sum_{i=1}^{n} \pi_{\mathrm{I}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} + \sum_{i=1}^{n} \pi_{\mathrm{C}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} \right) \\
&= -\sum_{i=1}^{n} \left(\pi_{\mathrm{I}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} + \pi_{\mathrm{C}} \frac{\mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)} \right) \\
&= -n.
\end{aligned}$$

Combining this with Equations (8) and (9) we obtain

$$\pi_{\mathrm{I}} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \tag{10}$$

and

$$\pi_{\mathrm{C}} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,\mathrm{C}}. \tag{11}$$

Combining these with Equations (4) and (5) we obtain

$$\mu_{\mathrm{I}} = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot x_i}{n \cdot \pi_{\mathrm{I}}} \tag{12}$$

and

$$\mu_{\mathrm{C}} = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} \cdot x_i}{n \cdot \pi_{\mathrm{C}}}, \tag{13}$$

and combining them with Equations (6) and (7) we obtain

$$\sigma_{\mathrm{I}}^2 = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot (x_i - \mu_{\mathrm{I}})^2}{n \cdot \pi_{\mathrm{I}}} \tag{14}$$

and

$$\sigma_{\mathrm{C}}^2 = \frac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} \cdot (x_i - \mu_{\mathrm{C}})^2}{n \cdot \pi_{\mathrm{C}}}. \tag{15}$$

Therefore, we see that if we know the values $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$ we can compute the best value for $\boldsymbol{\theta}$. (Note that as before, we need some more work to show that the values that we computed indeed give a local maximum, but we omit the details.)

Recall that Equations (2) and (3) tell us how we can compute the estimates $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$ if we know the value of $\boldsymbol{\theta}$, whereas Equations (10)–(15), tell us how we can compute $\boldsymbol{\theta}$ if we know the estimates $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$. This suggests the algorithm of Figure 1 for computing the best $\boldsymbol{\theta}$ and $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$, which is similar to k-means and is called *expectation–maximization* (EM).

1. **Function Expectation–Maximization**(**x**)
2. **Initialization step:**
3. Pick (say random) reasonable values for $\boldsymbol{\theta} = \{\pi_{\mathrm{I}}, \pi_{\mathrm{C}}, \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2, \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2\}$
4. **while** (not converged)
5. **Expectation step:**
6. $\gamma_{i,\mathrm{I}} \leftarrow \dfrac{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}$ (Equation (2))
7. $\gamma_{i,\mathrm{C}} \leftarrow \dfrac{\pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}{\pi_{\mathrm{I}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{I}}, \sigma_{\mathrm{I}}^2) + \pi_{\mathrm{C}} \cdot \mathcal{N}(x_i; \mu_{\mathrm{C}}, \sigma_{\mathrm{C}}^2)}$ (Equation (3))
8. **Maximization step:**
9. $\pi_{\mathrm{I}} \leftarrow \dfrac{1}{n}\sum_{i=1}^{n} \gamma_{i,\mathrm{I}}$ $\quad$ $\pi_{\mathrm{C}} \leftarrow \dfrac{1}{n}\sum_{i=1}^{n} \gamma_{i,\mathrm{C}}$ (Equations(10), (11))
10. $\mu_{\mathrm{I}} \leftarrow \dfrac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot x_i}{n \cdot \pi_{\mathrm{I}}}$ $\quad$ $\mu_{\mathrm{C}} \leftarrow \dfrac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} \cdot x_i}{n \cdot \pi_{\mathrm{C}}}$ (Equations(12), (13))
11. $\sigma_{\mathrm{I}}^2 \leftarrow \dfrac{\sum_{i=1}^{n} \gamma_{i,\mathrm{I}} \cdot (x_i - \mu_{\mathrm{I}})^2}{n \cdot \pi_{\mathrm{I}}}$ $\quad$ $\sigma_{\mathrm{C}}^2 \leftarrow \dfrac{\sum_{i=1}^{n} \gamma_{i,\mathrm{C}} \cdot (x_i - \mu_{\mathrm{C}})^2}{n \cdot \pi_{\mathrm{C}}}$ (Equations(14), (15))
12. **end while**
13. **Output** the final values of $\gamma_{i,\mathrm{I}}$, $\gamma_{i,\mathrm{C}}$, and $\boldsymbol{\theta}$

Figure 1: The expectation–maximization (EM) algorithm for learning Gaussian mixtures.

Initially we pick random values for the parameter vector $\boldsymbol{\theta}$. Given $\boldsymbol{\theta}$, we estimate the probability for each height to be of an Italian or a Chinese woman $\gamma_{i,\mathrm{I}} = \mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i)$ and $\gamma_{i,\mathrm{C}} = \mathbf{Pr}(Z_i = \mathbf{C} \,|\, X_i = x_i)$; this is the *expectation* step. Given these values, we proceed to the *maximization* step, in which we update the parameter vector $\boldsymbol{\theta}$. We repeat until we have converged, which can be tested, for instance, by checking that the value of $\boldsymbol{\theta}$ does not change a lot from the previous round. At the end we obtain our estimate for the best parameters of the model $\hat{\boldsymbol{\theta}}$ and our estimates for the probabilities $\mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i)$ and $\mathbf{Pr}(Z_i = \mathbf{I} \,|\, X_i = x_i)$, which induce the soft clustering that we wanted to compute in the first place.

Note that the EM algorithm resembles a lot the k-means algorithm. The expectation step corresponds to assigning points to clusters (there we only allow $\gamma_{i,\mathrm{I}}$ and $\gamma_{i,\mathrm{C}}$ to take the values 0 or 1; instead in the EM algorithm they can take any value in $[0, 1]$ but in both cases we have the constraint that they sum to 1). The maximization step corresponds to the step of k-means in which we compute the best new centers (as the new means), given the assignment of the points to clusters. Indeed, one can show that the k-means can be viewed as a special case of the EM algorithm, if we put the additional constraint that the variances of the two Gaussian distributions σ_{I}^2 and σ_{C}^2 are very small (tend to zero). In this way, we will end up with a hard clustering by performing, essentially, k-means.

Similar to k means we can show that the EM algorithm converges to some value, which is a local maximum but not necessarily a global one. But whereas the k-means algorithm usually converges fast, the EM can be much slower. Therefore, one trick often used when we want to produce a soft clustering, is to first run k-means and obtain a hard cluster, and then use this assignments to initialize the EM algorithm.

Here we saw one particular example of the EM algorithm, that of learning Gaussian mixtures (actually a special case of that). But note that the EM algorithm is much more general and can be used when we want to learn models that contain latent variables. The technical details might be more involved, but the approach is the same.

References

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[2] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.