Algorithm 7.1: Fitting mixture of Gaussians

The mixture of Gaussians (MoG) is a probability density model suitable for data $\mathbf{x}$ in D dimensions. The data is described as a weighted sum of K normal distributions

$$Pr(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k],$$

where $\boldsymbol{\mu}_{1\ldots K}$ and $\boldsymbol{\Sigma}_{1\ldots K}$ are the means and covariances of the normal distributions and $\lambda_{1\ldots K}$ are positive valued weights that sum to one.

The MoG is fit using the EM algorithm. In the E-step, we compute the posterior distribution over a hidden variable h_i for each observed data point $\mathbf{x}_i$. In the M-step, we iterate through the K components, updating the mean $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ for each and also update the weights $\{\lambda_k\}_{k=1}^{K}$.

Algorithm 7.1: Maximum likelihood learning for mixtures of Gaussians

```
Input  : Training data {x_i}_{i=1}^I, number of clusters K
Output: ML estimates of parameters θ = {λ_{1...K}, μ_{1...K}, Σ_{1...K}}
begin
    Initialize θ = θ_0 ^a
    repeat
        // Expectation Step
        for i=1 to I do
            for k=1 to K do
                l_ik = λ_k Norm_{x_i}[μ_k, Σ_k]                    // numerator of Bayes' rule
            end
            // Compute posterior (responsibilities) by normalizing
            for k=1 to K do
                r_ik = l_ik / (Σ_{k=1}^K l_ik)
            end
        end
        // Maximization Step ^b
        for k=1 to K do
            λ_k^[t+1] = (Σ_{i=1}^I r_ik) / (Σ_{k=1}^K Σ_{i=1}^I r_ik)
            μ_k^[t+1] = (Σ_{i=1}^I r_ik x_i) / (Σ_{i=1}^I r_ik)
            Σ_k^[t+1] = (Σ_{i=1}^I r_ik (x_i − μ_k^[t+1])(x_i − μ_k^[t+1])^T) / (Σ_{i=1}^I r_ik).
        end
        // Compute Data Log Likelihood and EM Bound
        L = Σ_{i=1}^I log[Σ_{k=1}^K λ_k Norm_{x_i}[μ_k, Σ_k]]
        B = Σ_{i=1}^I Σ_{k=1}^K r_ik log[λ_k Norm_{x_i}[μ_k, Σ_k] / r_ik]
    until No further improvement in L
end
```

[a] One possibility is to set the weights $\lambda_\bullet = 1/K$, the means $\boldsymbol{\mu}_\bullet$ to the values of K randomly chosen datapoints and the variances $\boldsymbol{\Sigma}_\bullet$ to the variance of the whole dataset.
[b] For a diagonal covariance retain only the diagonal of the $\boldsymbol{\Sigma}_k$ update.