Mathematical Methods for Data Analysis
Seminar 11

1 EM-algorithm

Let us recall some expressions that were considered in the lecture.
The Kullback-Leibler divergence is called the functional:

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx. \tag{1.1}$$

This function has the meaning of $<<$ distances» between distributions and has the following properties:

- $\mathrm{KL}(p\|q) \geqslant 0, \ \forall p, \ q;$
- $\mathrm{KL}(p\|q) = 0 \iff \text{supp } p = \text{supp } q.$

EM-algorithm — iterative method for maximizing sample likelihood. Let there be the following problem:

$$\log p(X|\Theta) \to \max_{\Theta} \tag{1.2}$$

Let the model have hidden variables Z that describe its internal state. For any distribution of $q(Z)$ on hidden variables, it is true:

$$\begin{aligned}\log p(X|\Theta) = \int q(Z) \log p(X|\Theta) dZ = \{p(X|\Theta)p(Z|X\Theta) = p(X,Z|\Theta)\} = \\ \int q(Z) \log \frac{p(X,Z|\Theta)}{p(Z|X,\Theta)} dZ = \int q(Z) \log \frac{p(X,Z|\Theta)q(Z)}{p(Z|X,\Theta)q(Z)} dZ = \\ \int q(Z) \log \frac{p(X,Z|\Theta)}{q(Z)} dZ + \int q(Z) \log \frac{q(Z)}{p(Z|X,\Theta)} dZ = \\ \mathcal{L}(q,\Theta) + \mathrm{KL}(q\|p).\end{aligned}$$

Since $\mathrm{KL}(q\|p) \geqslant 0$, then $\log p(X|\Theta) \geqslant \mathcal{L}(q,\Theta)$.

Recall that we would like to maximize the left side of the resulting inequality, independent of the distribution q, which, in turn, can be chosen arbitrarily, so the «correct» q is chosen, the more accurate the lower bound on the right side of the inequality will be. Instead of solving the original problem 1.2 , we will maximize the lower bound of $\mathcal{L}(q,\Theta)$ alternately by q and Θ.

E-step. Maximize by q.

It follows from the above that the maximum of $\mathcal{L}(q, \Theta)$ by q is reached when the minimum of $\mathrm{KL}(q\|p)$ is reached, that is, when $q = p$:

$$q^*(Z) = \arg\max_q \mathcal{L}(q, \Theta^{\text{old}}) = \arg\min_q \int q(Z) \log \frac{q(Z)}{p(Z|X, \Theta^{\text{old}})} dZ = p(Z|X, \Theta^{\text{old}})$$

M-step. Maximize by Θ.

$$\begin{aligned}\Theta^{\text{new}} &= \arg\max_\Theta \int q^*(Z) \log \frac{p(X, Z|\Theta)}{q^*(Z)} dZ = \arg\max_\Theta \int q^*(Z) \log p(X, Z|\Theta) dZ \\ &= \arg\max_\Theta \mathbb{E}_{Z \sim q^*(Z)} \log p(X, Z|\Theta)\end{aligned}$$

Problem 1.1. *Why is it necessary to reduce the original optimization problem 1.2 to the optimization problem at the M-step?*

Solution. Optimized function in the problem

$$\log p(X|\Theta) \to \max_\Theta \tag{1.3}$$

It often turns out to be non-convex. Due to the fact that we can enter the hidden variables Z in an arbitrary way, we can select them so that the task

$$\Theta^* = \arg\max_\Theta \mathbb{E}_Z \log p(X, Z|\Theta)$$

has a convenient form for optimization, for example, so that the distribution $p(X, Z|\Theta)$ in the class of exponential family distributions.

■

Problem 1.2. *Why does the EM algorithm need to converge to the local maximum of incomplete likelihood $p(X|\Theta)$?*

Solution. Consider the next iteration of the EM algorithm from the initial approximation Θ^{old}. Let us recall the decomposition of the logarithm of incomplete likelihood into the KL-divergence and the lower bound for some q:

$$\log p(X|\Theta^{\text{old}}) = \mathcal{L}(q, \Theta^{\text{old}}) + \mathrm{KL}(q(Z)\|p(Z|X, \Theta^{\text{old}}))$$

In the E-step, we choose $q^*(Z)$ equal to the posterior distribution on the hidden variables Z under the condition of the observed and current parameters of the model $p(Z|X, \Theta^{\text{old}})$, thus taking the KL-divergence, that is:

$$\log p(X|\Theta^{\text{old}}) = \mathcal{L}(q^*(Z), \Theta^{\text{old}})$$

In the M-step, we optimize the left side of the equality by the parameters Θ when $q^*(Z)$ is fixed, i.e. $\mathcal{L}(q^*(Z), \Theta^{\text{new}}) > \mathcal{L}(q^*(Z), \Theta^{\text{old}})$ (assuming that Θ^{old} is no longer the optimum point), then the following chain of inequalities holds:

$$\log p(X|\Theta^{\text{new}}) = \mathcal{L}(q^*, \Theta^{\text{new}}) + \mathrm{KL}(q^*(Z)\|p(Z|X, \Theta^{\text{new}})) > \mathcal{L}(q^*, \Theta^{\text{old}}) = \log p(X|\Theta^{\text{old}})$$

Thus, at each iteration of the EM algorithm, we increase the partial likelihood value of $p(X|\Theta)$.

■

2 Separation of a mixture of normal distributions

Consider a mixture of normal distributions. In this case, the probability density of our sample is described as follows:

$$p(X \mid \Theta) = \prod_{i=1}^{\ell} p(x_i \mid \Theta) = \prod_{i=1}^{\ell} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k),$$

where i is the index of the sample object, k is the index of the mixture components, $\pi_1, \ldots, \pi_K$ are the prior probabilities of the components.

Let's introduce the hidden variables Z. The variable z_{ik} has the meaning of the object belonging to the mixture component: it takes the value 1 if the ith object of the training sample belongs to the kth component of the mixture, and 0 otherwise, $\sum_k z_{ik} = 1$.

$$p(X, Z \mid \Theta) = \prod_{i=1}^{\ell} \prod_{k=1}^{K} \Big[\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)\Big]^{z_{ik}}$$

At the E-step, the a posterior distribution on the hidden variables is calculated:

$$p(Z \mid X, \Theta^{\text{old}}) \propto p(X, Z \mid \Theta^{\text{old}}) = \prod_{i=1}^{\ell} \prod_{k=1}^{K} \Big[\pi_k^{\text{old}} \mathcal{N}(x_i \mid \mu_k^{\text{old}}, \Sigma_k^{\text{old}})\Big]^{z_{ik}}$$

Note that this distribution is factorized into the product of distributions corresponding to individual objects $p(z_i \mid x_i, \Theta^{\text{old}})$:

$$p(Z \mid X, \Theta^{\text{old}}) = \prod_{i=1}^{\ell} p(z_i \mid x_i, \Theta^{\text{old}}) = \prod_{i=1}^{\ell} \frac{\prod_{k=1}^{K} \Big[\pi_k^{\text{old}} \mathcal{N}(x_i \mid \mu_k^{\text{old}}, \Sigma_k^{\text{old}})\Big]^{z_{ik}}}{\sum_{k=1}^{K} \pi_k^{\text{old}} \mathcal{N}(x_i \mid \mu_k^{\text{old}}, \Sigma_k^{\text{old}})}$$

Let's introduce the notation:

$$g_{ik} \equiv p(z_{ik} = 1 \mid x_i, \Theta^{\text{old}}) = \frac{\pi_k^{\text{old}} \mathcal{N}(x_i \mid \mu_k^{\text{old}}, \Sigma_k^{\text{old}})}{\sum_{s=1}^{K} \pi_s^{\text{old}} \mathcal{N}(x_i \mid \mu_s^{\text{old}}, \Sigma_s^{\text{old}})}.$$

Let us now calculate the total likelihood expectation:

$$\begin{aligned}
&\mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})} \log p(X, Z \mid \Theta) = \\
&= \mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})} \sum_{i=1}^{\ell} \sum_{k=1}^{K} z_{ik} \Big\{\log \pi_k + \log \mathcal{N}(x_i \mid \mu_k, \Sigma_k)\Big\} = \\
&= \sum_{i=1}^{\ell} \sum_{k=1}^{K} \mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})}[z_{ik}] \Big\{\log \pi_k + \log \mathcal{N}(x_i \mid \mu_k, \Sigma_k)\Big\}.
\end{aligned}$$

We will need an auxiliary value:

$$\mathbb{E}_{Z\sim p(Z\,|\,X,\Theta^{\text{old}})}[z_{ik}] = 1\cdot p(z_{ik}=1\,|\,x_i,\Theta^{\text{old}}) + 0 * p(z_{ik}=0\,|\,x_i,\Theta^{\text{old}}) = g_{ik}.$$

We get the following optimization problem:

$$\mathbb{E}_{Z\sim p(Z\,|\,X,\Theta^{\text{old}})}\log p(X,Z\,|\,\Theta) = \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\Big\{\log\pi_k + \log\mathcal{N}(x_i\,|\,\mu_k,\Sigma_k)\Big\} \to \max_{\{\pi_k,\mu_k,\Sigma_k\}}$$

$\pi_{\mathbf{k}}$: There is a restriction on the parameters π_k $\sum_k \pi_k = 1$, so we will use the Lagrange multipliers method:

$$\mathcal{F}(\pi,\lambda) = \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\log\pi_k + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right),$$

$$\nabla_{\pi_k}\mathcal{F} = \sum_i g_{ik}\frac{1}{\pi_k} + \lambda \Rightarrow \pi_k = \frac{1}{\lambda}\sum_i g_{ik}, \quad \lambda = \ell,$$

$$\pi_k = \frac{1}{\ell}\sum_i g_{ik}.$$

$\mu_{\mathbf{k}}$:

$$\mathcal{L}(q^*,\Theta) \propto^+_{\mu_k} \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\left[-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right],$$

$$\nabla_{\mu_k}\mathcal{L} = \sum_{i=1}^{\ell} g_{ik}\Sigma_k^{-1}(x_i-\mu_k) = \Sigma_k^{-1}\sum_{i=1}^{\ell} g_{ik}(x_i-\mu_k) = 0 \Rightarrow \sum_{i=1}^{\ell} g_{ik}(x_i-\mu_k) = \Sigma_k 0 = 0.$$

$$\mu_k = \frac{1}{\ell\pi_k}\sum_{i=1}^{\ell} g_{ik}x_i, \quad \ell\pi_k = \sum_{i=1}^{\ell} g_{ik}.$$

$\mathbf{\Sigma_k}$: Denote $\Lambda_k = \Sigma_k^{-1}$, then:

$$\mathcal{L}(q^*,\Theta) \propto^+_{\Lambda_k} \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\left[-\frac{1}{2}(x_i-\mu_k)^T\Lambda_k(x_i-\mu_k) + \frac{1}{2}\log\det\Lambda_k\right],$$

$$\nabla_{\Lambda_k}\mathcal{L} = \sum_{i=1}^{\ell} g_{ik}\left[-\frac{1}{2}(x_i-\mu_k)(x_i-\mu_k)^T + \frac{1}{2}\Lambda^{-1}\right] = 0,$$

$$\Sigma_k = \Lambda_k^{-1} = \frac{1}{\ell\pi_k}\sum_{i=1}^{\ell} g_{ik}(x_i-\mu_k)(x_i-\mu_k)^T.$$

3 Separation of a mixture of Bernoulli distributions

Consider a mixture of Bernoulli distributions:

$$p(x \mid \mu, \pi) = \sum_{k=1}^{K} \pi_k p(x \mid \mu_k),$$

where $x \in \mathbb{R}^d$, $\mu = \{\mu_1, \ldots, \mu_K\}$, $\mu_k \in [0,1]^d$, $\pi = \{\pi_1, \ldots, \pi_K\}$, $\sum_{i=1}^{K} \pi_k = 1$, and

$$p(x_j \mid \mu_k) = \mu_{kj}^{x_j}(1-\mu_{kj})^{1-x_j},$$
$$p(x \mid \mu_k) = \prod_{j=1}^{d} \mu_{kj}^{x_j}(1-\mu_{kj})^{1-x_j}.$$

In other words, the k-th component of the mixture — is such a distribution on d - dimensional binary vectors that the j-th coordinate of the vector has a Bernoulli distribution with the parameter μ_{kj}.

Let's introduce the hidden variables Z in the same way as in the previous problem:

$$p(X, Z \mid \Theta) = \prod_{i=1}^{\ell} \prod_{k=1}^{K} \Big[\pi_k p(x_i \mid \mu_k)\Big]^{z_{ik}}.$$

At the E-step, the a posterior distribution on the hidden variables is calculated:

$$p(Z \mid X, \Theta^{\text{old}}) = \frac{p(X, Z \mid \Theta^{\text{old}})}{p(X \mid \Theta^{\text{old}})} = \frac{\prod_{i=1}^{\ell}\prod_{k=1}^{K}\Big[\pi_k^{\text{old}} p(x_i \mid \mu_k^{\text{old}})\Big]^{z_{ik}}}{\sum_Z \prod_{i=1}^{\ell}\prod_{k=1}^{K}\Big[\pi_k^{\text{old}} p(x_i \mid \mu_k^{\text{old}})\Big]^{z_{ik}}}.$$

Note that this distribution is factorized into the product of distributions corresponding to individual objects $p(z_i \mid x_i, \Theta^{\text{old}})$:

$$p(Z \mid X, \Theta^{\text{old}}) = \prod_{i=1}^{\ell} p(z_i \mid x_i, \Theta^{\text{old}}) = \prod_{i=1}^{\ell} \frac{\prod_{k=1}^{K}\Big[\pi_k^{\text{old}} p(x_i \mid \mu_k^{\text{old}})\Big]^{z_{ik}}}{\sum_{k=1}^{K} \pi_k^{\text{old}} p(x_i \mid \mu_k^{\text{old}})},$$

Let's introduce the notation:

$$g_{ik} \equiv p(z_{ik} = 1 \mid x_i, \Theta^{\text{old}}) = \frac{\pi_k^{\text{old}} p(x_i \mid \mu_k^{\text{old}})}{\sum_{s=1}^{K} \pi_s^{\text{old}} p(x_i \mid \mu_s^{\text{old}})}.$$

Let us now calculate the total likelihood expectation:

$$\mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})} \log p(X, Z \mid \Theta) =$$
$$= \mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})} \sum_{i=1}^{\ell}\sum_{k=1}^{K} z_{ik}\Big\{\log \pi_k + \log p(x_i \mid \mu_k)\Big\} =$$
$$= \sum_{i=1}^{\ell}\sum_{k=1}^{K} \mathbb{E}_{Z \sim p(Z \mid X, \Theta^{\text{old}})}[z_{ik}]\Big\{\log \pi_k + \log p(x_i \mid \mu_k)\Big\}.$$

We will need an auxiliary value:

$$\mathbb{E}_{Z\sim p(Z\,|\,X,\Theta^{\text{old}})}[z_{ik}] = 1 * p(z_{ik} = 1\,|\,x_i, \Theta^{\text{old}}) + 0 * p(z_{ik} = 0\,|\,x_i, \Theta^{\text{old}}) = g_{ik}.$$

We get the following optimization problem:

$$\mathbb{E}_{Z\sim p(Z\,|\,X,\Theta^{\text{old}})} \log p(X, Z\,|\,\Theta) = \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\Big\{\log \pi_k + \log p(x_i\,|\,\mu_k)\Big\} =$$

$$= \sum_{i=1}^{\ell}\sum_{k=1}^{K} g_{ik}\Big\{\log \pi_k + \sum_{j=1}^{d}(x_{ij}\log \mu_{kj} + (1 - x_{ij})\log(1 - \mu_{kj}))\Big\} \to \max_{\{\pi_k,\,\mu_k\}}$$

Differentiating this functional, we can obtain the formulas of the M-step:

$$\pi_k^{\text{new}} = \frac{1}{\ell}\sum_{i=1}^{\ell} g_{ik};$$

$$\mu_{kj}^{\text{new}} = \frac{\sum_i g_{ik}x_{ij}}{\sum_i g_{ik}}.$$

4 Restoring markup using the EM algorithm

Currently, models that require a large amount of marked-up data (for example, neural networks) are increasingly popular. However, marking up a large sample for a new task is an expensive procedure. Now there are services where performers can mark up the customer's data for a small fee (for example, to answer whether the person in the given image is smiling). Examples of such services are Amazon Mechanical Turk and Yandex.Toloka.

The experts who mark up the data are not interested in high-quality markup or do not have sufficient competence, so the resulting data is noisy, which can greatly affect the quality of the final algorithm. The main way to combat this effect is to average the responses across multiple experts, i.e. a simple vote. This method has obvious problems: it does not take into account (1) the competence of each expert and (2) the complexity of the problem being solved.

Consider a dataset ofℓ images of a binary classification problem, for which the answers of experts $l_{ij} \in \{0, 1\}$, are collected, where l_{ij} — the markup of the $i$image by the$j$ expert (the observed variables), and the true labels of the images $z_i \in \{0, 1\}$ are known, that is, the hidden variables that we want to recover. Additionally, we set the parameters $\alpha_j \in (-\infty, +\infty)$ — the level of expertise of the $j$expert and $\beta_i \in (0, +\infty)$ such that $1/\beta_i$ represents the complexity of the $i$problem. The joint distribution is defined as follows (l_i denotes a set of labels for the ith image from all experts):

$$p(z_i, l_i|\alpha, \beta_i) = p(z_i)\prod_j p(l_{ij}|z_i, \alpha_j, \beta_i),$$

$$p(l_{ij} = z_i|z_i, \alpha_j, \beta_i) = \sigma(\alpha_j\beta_i) = \frac{1}{1 + \exp(-\alpha_j\beta_i)},$$

Why do we set parameters and joint distribution this way? It turns out that such a probabilistic model has several properties that are logical for our problem. If we fix β_i and consider $\alpha_j \to +\infty$, we get $p(l_{ij} = z_i|z_i, \alpha_j, \beta_i) \to 1$, that is, such an expert always solves correctly any problem of finite complexity. Similarly, with $\alpha_j \to -\infty$, we get $p(l_{ij} = z_i|z_i, \alpha_j, \beta_i) \to 0$, which means that we get a very harmful expert Advisor that intentionally (or out of confusion) spoils our markup.

Now let's look at the value of $1/\beta_i$. If $1/\beta_i \to 0$, then for a "good"expert ($\alpha_j > 0$) we get $p(l_{ij} = z_i|z_i, \alpha_j, \beta_i) \to 1$, and for a "bad"expert ($\alpha_j < 0$) we get $p(l_{ij} = = z_i|z_i, \alpha_j, \beta_i) \to 0$. That is, any expert easily identifies the correct class of the image and issues the markup based on its harmfulness. In contrast, for $1/\beta_i \to +\infty$, we get the probability $p(l_{ij} = z_i|z_i, \alpha_j, \beta_i) \to 1/2$ for both "good"and "bad"experts. The authors of this probabilistic model called it GLAD (Generative model of Labels, Abilities and Difficulties).

The optimal parameters α and β, as well as the distribution of the true markup z_i, will be searched using the EM algorithm. But first, let's pay attention to the a prior distribution of $p(z_i)$. You can leave it uniform $p(z_i = 0) = p(z_i = 1) = 1/2$, or you can add our ideas about the balance of classes among images to it. Another option is to enter the parameter $\pi = p(z_i = 1)$ and perform optimization on it at the M-step. But we will leave this probability fixed.