MetaGen Inference

Zhangir Azerbayev

2022

1 Inference Framework

In our baseline model, we have N observations $X = \{x^n\}$, where each x^n is a coordinate in 3-space. In the full model, we have N observations $X = \{(x^n, c^n)\}$, where the x^n are locations and the c_n are object categories. We are trying to infer the world-state $\Theta = \{(\mu_k, c_k)\}_{k=1}^K$, where the μ_k are object locations and the c_k are object categories.

Suppose we augment are model with a latent variable $Z = \{z_k^n\}$, where z^n is a one-hot vector that describes which compnent of the GMM generated X^n. Thus $P(X|\Theta)$ factors as $P(X|\Theta) = P(Z)P(X|Z, \Theta)$.

The *expectation-maximization* (EM) algorithm hinges on two criteria:

1. It is easy to optimize $P(X, Z|\Theta)$ wrt Θ (see later section)
2. Although we have poor "global" information about Z, i.e we don't know $P(Z|X)$, we have "local" information about Z, i.e we know $P(Z|X, \Theta)$ for a fixed Θ.

The idea of the algorithm is this: instead of maximizing log-likelihood, maximize the expectation of log-likelihood under the posterior of Z. More precisely, we iterate the two following steps:

1. E-step: Using Θ^{old} from the previous iteration, evaluate $P(Z|X, \Theta^{\text{old}})$.
2. M-Step: Solve for $\Theta = \text{argmax}_\Theta \sum_Z P(Z|X, \Theta) \log P(X, Z|\Theta^{\text{old}})$, which is the expectation of log-likelihood under the posterior of Z.

2 Inference for MetaGen

Define the *responsibility* of z_k^n for x^n as

$$\gamma(z_k^n) = P(z_k^n = 1|x^n, \Theta) = \frac{P(x^n|z_k^n = 1, \Theta)}{\sum_{j=1}^K P(x^n|z_j^n, \Theta)}.$$

Note that the E-step is equivalent to computing the responsibilities for each n and k. The M-step then becomes

$$\text{argmax}_\Theta \sum_{n=1}^N \sum_{k=1}^K \gamma(z_k^n) \log P(x^n, z_k^n = 1|\Theta^{\text{old}}). \tag{1}$$

In the case of the baseline model, this is a continuous optimization problem. In the full MetaGen case, we can simply sample category assignments and condition on them, yielding multiple continuous optimization problems.

3 Optimization for M-step

I currently do not know whether (1) has a closed form solution (my guess is that it doesn't). Worst case, we do gradient-based optimizaton.