3. ~~K-means~~

~~3. Variational methods: from K-means (as EM algorithm) to~~

3. K-means and EM.

Let us consider a task of putting a set of data points $\{\vec{x}^{(n)}\}_{n=1}^{N}$ into K clusters.

1) Classical K-Means Algo:

1- Set K means $\{\vec{m}^{(k)}\}$ to random values

2- For each $x^{(n)}$: $\underbrace{\hat{k}^{(n)}}_{\text{predicted cluster}} = \arg\min_k \{d(\vec{m}^{(k)}, \vec{x}^{(n)})\}$, d-distance

3- Set $r_k^{(n)} = I(\hat{k}^{(n)} = k)$ and update $\vec{m}^{(k)} \leftarrow \dfrac{\sum_{n=1}^{N} r_k^{(n)} \vec{x}^{(n)}}{\sum_{n=1}^{N} r_k^{(n)}}$ $\forall k$

4- Repeat 2 and 3 until convergence.

2) Problems of K-Means

- too "hard" algorithm: points are assigned ~~to~~ to exactly one cluster, but ~~bottom~~ border points ~~should be presen~~ better be presented in multiple
- the classic distance couldn't represent a narrow clusters with K-Means.
- doesn't have any ~~info~~ represent. of the weight and breadth of each cluster.

3) Soft K-Means Clustering: (Improvement) β - hyper-param.

1- Set K means $\{\vec{m}^{(k)}\}$ to random values

2- For each $x^{(n)}$ and k: $r_k^{(n)} = \dfrac{\exp(-\beta\, d(\vec{m}^{(k)}, \vec{x}^{(n)}))}{\sum_{k'} \exp(-\beta\, d(\vec{m}^{(k')}, \vec{x}^{(n)}))}$ - softmax over distances.

3- Update $\vec{m}^{(k)} \leftarrow \dfrac{\sum_{n=1}^{N} r_k^{(n)} \vec{x}^{(n)}}{\sum_{n=1}^{N} r_k^{(n)}}$

4- Repeat 2 and 3 until convergence.

4) Further soft k-means enhancements:

1- ~~Set to~~ ~~Assign~~ Initialize $\{\vec{m}^{(k)}\}$ - cluster centers, $\{\pi_k\}$ - cluster importances, $\{\sigma_k^2\}$ - cluster dispersions.

2- For each $x^{(n)}$ and k: $$r_k^{(n)} = \frac{\pi_k \left(\frac{1}{\sqrt{2\pi\sigma_k^2}}\right)^d \exp\left(-\frac{1}{\sigma_k^2} d(\vec{m}^{(k)}, \vec{x}^{(n)})\right)}{\sum_{k'=1}^{K} -//-}$$ ⇐ normaliz. constant

3- Adjust $m^{(k)}$, π_k and σ_k^2 to match the data:

$$\cancel{\vec{m}^{(k)} \leftarrow \frac{\sum_n r_k^{(n)} \vec{x}^{(n)}}{\sum_{n'} -//-}, \quad \sigma_k^2 \leftarrow \frac{\sum_n r_k^{(n)} (\vec{x}^{(n)} - \vec{m}^{(k)})^2}{d \sum_{n'} -//-}, \quad \pi_k}$$

$$\vec{m}^{(k)} \leftarrow \frac{\sum_n r_k^{(n)} \vec{x}^{(n)}}{R^{(k)}}, \quad \sigma_k^2 \leftarrow \frac{\sum_n r_k^{(n)} (\vec{x}^{(n)} - \vec{m}^{(k)})^2}{d \cdot R^{(k)}}, \quad \pi_k \leftarrow \frac{R^{(k)}}{\sum_k R^{(k)}},$$

where $R^{(k)} = \sum_n r_k^{(n)}$, d - dimensionality of $\vec{x}$.

5) General Idea: EM.

Let X denote a set of observed data, Z - a set of latent data, Θ - a vector of unknown parameters.

the EM-algorithm is a way to find MLE of the Θ, in other words to maximize $p(X|\Theta) = \int p(X, Z|\Theta)\, dZ$

$p(X,Z|\Theta)$ ″

1- (Expectation step) $Q(\Theta|\Theta^{(t-1)}) = \mathbb{E}_{Z|X,\Theta^{(t-1)}}[\log \mathcal{L}(\Theta, X, Z)]$

2- (Maximization step) $\Theta^{(t)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(t-1)})$.

In example 4) $\Theta^{(t)} = \{\vec{m}^{(k)}, \sigma_k^2, \pi_k\}_{k=1}^{K}$, $Z = \{r_k^{(n)}\}_{k,n=1}^{K,N}$

6) A fatal flaw of maximum likelihood

E.g., let's look at algorithm 4). If $\vec{m}^{(k)} \equiv \vec{x}^{(\tilde{n})}$ during iterations, then $\sigma_k^2 \leftarrow 0$. If σ_k^2 is sufficiently small, then it becomes even smaller during iterations.