Lecture 21 Gaussian mixture models and t-SNE

In the previous lecture we introduced various algorithms to perform clustering. In this lecture we will study the relationship between clustering and latent or hidden variables. We can think of the cluster identity of each data point (i.e. which cluster does a data point belong to) as a latent variable. Latent variables are a way of representing correlations between data points. We can therefore think of clustering as an algorithm to learn the most probable value of a latent variable. Calculating this latent variable requires additional assumptions about the structure of our data set, in particular we must make an assumption about the underlying probability distribution from which the data was generated. A model for how the data is generated is called GENERATIVE MODEL. We do the following

- assume that data points are assigned a cluster, with each cluster characterised by some cluster-specific probability distribution (e.g. a Gaussian with some mean and variance)
- specify a procedure to find the value of the latent variable, typically by choosing the values of the latent variable that minimise some cost function.

In MLE we choose the values of the latent variables that maximise the likelihood of the observed data under our generative model.

21.1 Gaussian mixture models

Gaussian mixture models (GMM) are a generative model often used in the context of clustering. In GMM, points are drawn from one of K Gaussians, each with its own mean $\vec{\mu}_k$ and covariance matrix Σ_k

$$\mathcal{N}(\vec{x}\,|\,\vec{\mu}, \Sigma) \sim \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})\,\Sigma^{-1}(\vec{x}-\vec{\mu})^T\right]$$

Let us denote the probability that a point is drawn from mixture k by π_k. Then the probability of generating a point $\vec{x}$ in a GMM is given by

$$p(\vec{x}\,|\,\{\vec{\mu}_k, \Sigma, \pi_k\}) = \sum_{k=1}^{K} \mathcal{N}(\vec{x}\,|\,\vec{\mu}_k, \Sigma)\,\pi_k$$

Given a data set $X = \{\vec{x}_1, \ldots, \vec{x}_N\}$, with N the number of data points, we can write the likelihood of the data set as

$$p(X\,|\,\{\vec{\mu}_k, \Sigma, \pi_k\}) = \prod_{i=1}^{N} p(\vec{x}_i\,|\,\{\vec{\mu}_k, \Sigma_k, \pi_k\})$$

Let us denote, for brevity, the set of parameters (of K Gaussians in the model) by $\vec{\vartheta} = \{\vec{\mu}_k, \Sigma_k, \pi_k\}$

Let us introduce discrete binary K-dimensional latent variables $\vec{z}$ for each data point $\vec{x}$ whose k-th component is 1 if point $\vec{x}$ was generated from the k-th Gaussian, and zero otherwise. Example: suppose that we consider a Gaussian mixture with $K = 3$, we would have, for each data point, three possible values of $z \equiv (z_1, z_2, z_3)$: $(1,0,0)$, $(0,1,0)$, $(0,0,1)$. We cannot directly observe the variable $\vec{z}$. It is a latent variable that encodes the cluster identity of point $\vec{x}$. We denote all the (N) latent variables corresponding to a data set X by Z.

Viewing the GMM as a generative model, we can write the probability $p(\vec{x}/\vec{z})$ of observing a data point $\vec{x}$ given $\vec{z}$ as

$$p(\vec{x}|\vec{z};\{\vec{\mu}_k, \Sigma_k\}) = \prod_{k=1}^{K} \mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k)^{z_k}$$

and the probability of observing a given value of the latent variable

$$p(\vec{z}|\{\pi_k\}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Using Bayes' rule, we can write the joint probability of a clustering assignment $\vec{z}$ and a data point $\vec{x}$ given the GMM parameters as

$$p(\vec{x}, \vec{z}, \vec{\vartheta}) = p(\vec{x}/\vec{z}; \{\vec{\mu}_k, \Sigma\})\, p(z|\{\pi_k\})$$

We can re-arrange this expression as

$$\gamma(z_k) \equiv p(z_k = 1 | \vec{x}; \vec{\vartheta}) = \frac{\pi_k \mathcal{N}(\vec{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}|\mu_j, \Sigma_j)}$$

which gives the conditional probability of the data point $\vec{x}$ being in the k-th cluster. The $\gamma(z_k)$ are often referred to as the RESPONSIBILITY that mixture k takes for explaining $\vec{x}$. The complication is that we do not know the parameters $\vec{\vartheta}$ of the underlying GMM – we must learn them from the data. Naively, one may do this by maximizing the likelihood

$$\hat{\vec{\vartheta}} = \underset{\vec{\vartheta}}{\arg\max} \ln p(X, \vec{\vartheta}) \qquad \vec{\vartheta} = \{\vec{\mu}_k, \Sigma_k, \pi_k\}$$

Once we know the MLEs $\hat{\vec{\vartheta}}$, we could compute the $\gamma(z_k)$. In practice, it is very complicated to find the maximum of the likelihood due to its complexity. It is simpler to find a local maximum, e.g. by means of gradient descent. An alternative powerful (iterative) procedure is EXPECTATION

MAXIMISATION (EM). Given an initial guess for the parameters $\vartheta^{(0)}$, the EM algorithm iteratively generates new estimates for the parameters $\vartheta^{(1)}, \vartheta^{(2)}, \ldots$ The central observation underlying EM is that it is often much easier to compute the conditional likelihoods of the latent variables $\tilde{p}^{(t)}(Z) = p(Z \mid X; \vartheta^{(t)})$ given some choice of parameters and the maximum of the expected log likelihood given an assignment of the latent variables: $\vartheta^{(t+1)} = \arg\max_{\vartheta} \mathbb{E}_{p(Z|X,\vartheta^{(t)})}\left[\ln p(X, Z; \vartheta)\right]$

Note that we can write

$$\mathbb{E}_{\tilde{p}^{(t)}}\left[\ln p(X, Z; \vartheta)\right] = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik}^{(t)} \left[\ln \mathcal{N}(\vec{x}_i \mid \vec{\mu}_k, \Sigma_k) + \ln \pi_k\right]$$

where we have used the shorthand notation $\gamma_{ik}^{(t)} = p(z_{ik} \mid X; \vartheta^{(t)})$ with z_{ik} the k-th component of $\vec{z}_i$. Taking the derivative of this equation w.r.t. the parameters $\vec{\mu}_k, \Sigma_k, \pi_k$ (subject to the constraint $\sum_k \pi_k = 1$) and setting this to zero yields

$$\vec{\mu}_k^{(t+1)} = \frac{\sum_i^N \gamma_{ik}^{(t)} x_i}{\sum_i \gamma_{ik}^{(t)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_i^N \gamma_{ik}^{(t)} (\vec{x}_i - \vec{\mu}_k)(\vec{x}_i - \vec{\mu}_k)^T}{\sum_i \gamma_{ik}^{(t)}}$$

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_i^N \gamma_{ik}^{(t)}$$

These are the usual estimates for the mean and variance with each data point weighed according to our current best guess for the probability that it belongs to cluster k. We can then use $\vartheta^{(t+1)}$ to compute $\gamma_{ik}^{(t+1)}$ and repeat the process.

Remarks:

1 It is often useful to think of the visible correlations between features in the data as resulting from hidden latent variables.

2 We will often posit a generative model that encodes the structure we think exists in the data and then find parameters that maximise the likelihood of the observed data.

3 Often we will not be able to directly estimate the MLE, and will have to look for ways to find local minima.

4 Clustering data in high dimension can be very challenging, the reason being the accumulation of noise. It is common practice to de-noise the data before proceeding with usual clustering algorithms. Simple feature selection like PCA can be insufficient.

21.2 t-SNE

It is often desirable to preserve local structures in high-dimensional data sets. When dealing with data sets having clusters delimited by complicated surfaces or data sets with a large number of clusters, preserving local structures becomes difficult with PCA. A recent technique, called t-stochastic neighbour embedding (t-SNE), has become promising with high-dimensional data. t-SNE is a non-parametric method that constructs non-linear embeddings. Each high-dimensional training point is mapped to low-dim. embedding coordinates, optimised in a way to preserve the local structure in the data.

The idea of stochastic neighbour embedding is to associate a probability distribution to the neighbourhood of each data (as unvel $\vec{x} \in \mathbb{R}^p$, p is the number of features)

$$p_{i|j} = \frac{\exp\left(-\|\vec{x}_i - \vec{x}_j\|_2^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|\vec{x}_i - \vec{x}_k\|^2 / 2\sigma_i^2\right)}$$

where $p_{i|j}$ is the likelihood that x_j is x_i's neighbour (we expect $p_{i|j} = 0$ if $\vec{x}_i$ is close to $\vec{x}_j$; σ_i are free band-width parameters that are usually determined by fixing the local entropy $H(p_i)$ of each data point

$$H(p_i) \equiv -\sum_j p_{j|i} \ln_2 p_{j|i}$$

The local entropy is then set to equal a constant across all data points $\Sigma = 2^{H(p_i)}$ where Σ is called PERPLEXITY. The perplexity constraint determines $\sigma_i \; \forall i$ and implies that points in region of high-density will have smaller σ_i. Using Gaussian likelihoods in $p_{i|j}$ implies that only points that are nearby x_i contribute to its probability distribution. While this ensures that the similarity for nearby points is well represented, this can be a problem for points that are far away from x_i: they have exponentially vanishing contributions to the distribution, which in turn means that their embedding coordinates are ambiguous.

A way to overcome this issue is to define a symmetrised distribution

$$p_{i|j} \longrightarrow p_{ij} \equiv (p_{i|j} + p_{j|i})/2N$$

This guarantees that $\sum_j p_{ij} > 1/(2N)$ for all data points x_i

making a significant contribution to the cost function. 7

t-SNE constructs a similar probability distribution q_{ij} in a low-dimensional latent space with coordinates

$$Y = \{y_i\}, \; y_i \in \mathbb{R}^{p'}, \; p' < p$$

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}}$$

Note that q_{ij} is a long tail distribution. This preserves short-distance information (relative neighbourhoods) while strongly repelling two points that are far apart in the original space. In order to find the latent-space coordinates, t-SNE minimises the Kullback-Leibler divergence between q_{ij} and p_{ij}

$$C(Y) = D_{KL}(p \| q) = \sum_{ij} p_{ij} \ln \left(\frac{p_{ij}}{q_{ij}} \right)$$

The minimisation is performed via gradient descent. Let us compute the gradient of C w.r.t. y_i

$$\partial_{y_i} C = \sum_{j \neq i} 4\, p_{ij}\, q_{ij}\, Z_i (y_i - y_j) - \sum_{j \neq i} 4\, q_{ij}^2\, Z_i (y_i - y_j)$$

where $Z_i = 1 \Big/ \left(\sum_{k \neq i} (1 + \|y_k - y_i\|^2)^{-1} \right)$. We have separated the gradient into an attractive $F_{attractive}$ and repulsive $F_{repulsive}$ terms. Note that $F_{attractive,\,i}$ induces a significant attractive force only between points that are nearby points (of i) in the original space. Finding the embedding coordinates y_i is thus equivalent to finding the equilibrium configurations of particles interacting through attractive and repulsive forces.

Remarks

1 t-SNE can rotate the data. The KL divergence is invariant under rotations in the latent space.

2 t-SNE results are stochastic. The solution will depend on the initial seed.

3 t-SNE generally preserves short-distance information (it preserves ordination, but not actual distances)

4 scales are deformed in t-SNE (a scale-free distribution is used in the latent space.

5 t-SNE is computationally expensive: it scales as $O(N^2)$.