STAT 535: Statistical Machine Learning **Autumn 2024**

Lecture 10: Dimension Reduction and Manifold Learning

Instructor: Yen-Chi Chen

10.1 Introduction

Dimension reduction is an important topic in multivariate statistics and unsupervised learning. The problem is very simple. Given observations $X_1, \cdots, X_n \in \mathbb{R}^d$, where d could be potentially very large, we want to create a low-dimensional version $Y_1, \cdots, Y_n \in \mathbb{R}^m$ that m is much smaller than d while the low-dimensional versions have a *similar property* to the original data. Namely, the reduction process preserves some properties of the original data

The key is the property we want to preserve. It turns out that preserving different properties leads to different dimension reduction techniques. Therefore, there is no single method that is optimal in every case. The final choice of the method depends on the properties that we want to preserve.

10.2 Principle component analysis

Principle component analysis (PCA) is a very popular approach to dimension reduction. The principle components (PCs) are the directions that explains the majority of the sample covariance. Given the data $X_1, \cdots, X_n \in \mathbb{R}^d$, we first compute the sample covariance matrix

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)(X_i - \bar{X}_n)^T.$$

Then we perform spectral decomposition of the matrix $\widehat{\Sigma}$ as

$$\widehat{\Sigma} = \sum_{\ell=1}^{d} \widehat{\lambda}_\ell \widehat{v}_\ell \widehat{v}_\ell^T,$$

where $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_d$ are the eigenvalues of $\widehat{\Sigma}$ and $\widehat{v}_1, \cdots, \widehat{v}_d \in \mathbb{R}^d$ are the corresponding eigenvectors.

As a dimension reduction technique, the PCA will choose the top m eigenvectors $\widehat{v}_1, \cdots, \widehat{v}_m$ and compute the projection $Y_{i,\ell} = (X_i - \bar{X}_n)^T \widehat{v}_\ell$ for $\ell = 1, \cdots, m$. Then new observation $Y_i = (Y_{i,1}, \cdots, Y_{i,m})^T \in \mathbb{R}^m$ is the reduced dimension version of X_i.

The reasoning behind the PCA is the reconstruction property of the eigenvectors. Now consider a population level analysis that a random variable $X \in \mathbb{R}^d$ has a mean vector μ and a covariance matrix Σ. The covariance matrix admits a spectral decomposition:

$$\Sigma = \sum_{\ell=1}^{d} \lambda_\ell v_\ell v_\ell^T.$$

Now we consider reconstructing X using an m-dimensional linear subspace. Given an orthonormal basis

$e = (e_1, \cdots, e_m)$ with $e_i \in \mathbb{R}^d$, we can reconstruct X using

$$T_e(X) = \mu + \sum_{\ell=1}^{m} (X - \mu)^T e_\ell e_\ell.$$

We measure the reconstruction error using

$$R(e) = \mathbb{E}\|X - T_e(X)\|^2$$

It turns out that if we want to minimize the reconstruction error among all possible linear basis, the optimal choice will be

$$e_\ell = v_\ell.$$

Namely, the eigenvectors are the ones that minimizes the reconstruction error (in fact, you can prove this by projection property of eigenvectors).

Moreover, the smallest reconstruction error will be

$$R_m^* = R(v) = \sum_{\ell=m+1}^{d} \lambda_\ell,$$

the summation of remaining eigenvalues. Thus, the ratio

$$\frac{\sum_{\ell=1}^{m} \lambda_\ell}{\sum_{j=1}^{d} \lambda_j}$$

is often called the variance explained by the top m principle components.

10.3 Multidimensional scaling

Multidimensional scaling (MDS) is a simple but popular dimension reduction technique. The idea is very simple, given $X_1, \cdots, X_n \in \mathbb{R}^d$, we try to find a map $T : \mathbb{R}^d \to \mathbb{R}^m$ with $m << d$ but the distance

$$\|Z_i - Z_j\| \approx \|X_i - X_j\|.$$

We can view this as a minimization problem where the objective/loss function is

$$L(T) = \sum_{i \neq j} (\|X_i - X_j\|^2 - \|Z_i - Z_j\|^2)^2.$$

If we choose T to be a linear mapping, i.e., $T(x) = Sx$ for some matrix $S \in \mathbb{R}^{m \times d}$, then you can show that the resulting observations $Z_1, \cdots, Z_n$ will be the same as using the PCA.

Note that MDS can be generalized to other metric space. For instance, if observations $X_1, \cdots, X_n$ are not in Euclidean space but in some other metric space $\mathbb{M}$ with a metric d, then we can replace the loss function by

$$L(T) = \sum_{i \neq j} (d(X_i, X_j)^2 - \|Z_i - Z_j\|^2)^2$$

and the map $T : \mathbb{M} \to \mathbb{R}^k$.

10.4 Isomap

The Isomap is based on a similar idea to the MDS but it tries to recover local geometry in the data. Note that here observations $X_1, \cdots, X_n \in \mathbb{R}^d$. The Isomap consists of the following three steps:

1. *Local graph construction.* We first find the k-NN (or ϵ-) graph G of the data. Namely, $G = (V, E)$, where the vertex set V is the collection of all observations and E is based on k-NN edge or ϵ-edge.
2. *Geodesic distance approximation.* We measure the pairwise distance D_{ij} between two observations X_i and X_j using the shortest path distance in the graph G.
3. *MDS.* We use the distance matrix D and apply the MDS. Namely, we try to find $Z_1, \cdots, Z_n \in \mathbb{R}^m$ such that
$$L(Z_1, \cdots, Z_n) = \sum_{i \neq jj} (D_{ij}^2 - \|Z_i - Z_j\|^2)^2$$
is minimized.

In a sense, we can view the Isomap as a metric-based MDS where the distance between a pair X_i, X_j is measured by the graph distance. Suppose observations $X_1, \cdots, X_n$ is supported on an s-dimensional manifold in $\mathbb{R}^d$. When n is large and k is small relative to n, the graph distance is approximating the geodesic distance on the s-dimensional manifold. So the Isomap can be viewed as MDS with an approximated geodesic distance.

10.5 Local linear embedding

The local linear embedding (LLE) is another popular dimension reduction technique. Its idea is very simple. First, we find a suitable neighborhood of each observation and represent an observation by a linear combination of its neighboring points. The loading of such linear combination forms a weight matrix that represents the local structure of all observations. Finally, we try to find a lower-dimensional representation of the original sample that has a similar local structure.

Formally, the LLE consists of the following three steps:

1. *Local graph construction.* We first find the k-NN (or ϵ-) graph G of the data. These k points are the neighbors of each observation.
2. *Local weighting matrix.* Let $W \in \mathbb{R}^{n \times n}$ be a weight matrix from solving the following problem:
$$\min_W \sum_{i=1}^n \|X_i - \sum_{j=1}^n Wij X_j\|_2^2$$
with constraints that $W_{ij} = 0$ if j is not in the k-NN of i and $1 = \sum_j W_{ij}$.
3. *Dimension reduction.* Finally, we try to find $Y_1, \cdots, Y_n \in \mathbb{R}^m$ such that
$$\Phi(Y_1, \cdots, Y_n) = \sum_{i=1}^n \|Y_i - \sum_{j=1}^n W_{ij} Y_j\|_2^2 \tag{10.1}$$
is minimized with constraints that
$$\sum_{i=1}^n Y_i = 0, \quad \frac{1}{n} \sum_{i=1}^n Y_i Y_i^T = \mathbf{I}_m.$$

As can be seen from the above process, the weight matrix W contains information on the local structure. Its element informs us how each observation is associated with its neighborhoods. The final step is to find a lower dimensional representation with a similar local structure.

The minimization in equation (10.1) can be done easily using eigen-decomposition. To see this, we can rewrite equation (10.1) as

$$\Phi(Y_1, \cdots, Y_n) = \sum_{i=1}^{n} \|Y_i - \sum_{j=1}^{n} W_{ij} Y_j\|_2^2 = \mathbb{Y}^T (I - W)^T (I - W) \mathbb{Y},$$

where $\mathbb{Y} = (Y_1, \cdots, Y_n)^T \in \mathbb{R}^{n \times m}$. It turns out that the solution to the constraint minimization problem will be the m-smallest non-zero eigenvectors of $(I-W)^T(I-W)$. Namely, let $u_1, \cdots, u_m$ be the eigenvectors corresponding to the m-smallest non-zero eigenvalues of $(I - W)^T(I - W)$. Then $Y_i = (u_{1,i}, \cdots, u_{m,i})$ for each $i = 1, \cdots, n$. Thus, the step 3 can be done quickly by solving the eigenvalue/eigenvector problem of $(I - W)^T(I - W)$.

10.6 Laplacian-based approach

Dimension reduction can also be achieved via spectral methods, i.e., graph Laplacian. Here we will discuss two popular idea along this direction.

10.6.1 Laplacian eigenmap

The Laplacian eigenmap is a popular approach to perform dimension reduction using graph Laplacian. It uses a procedure that is very similar (and almost identical) to spectral clustering.

Given observations $X_1, \cdots, X_n \in \mathbb{R}^d$, we first compute the similarity matrix $S \in \mathbb{R}^{n \times n}$ for every pair of observation. This similarity matrix can be based on either k-NN approach, ϵ-neighborhood, or kernel approach, just like the case of spectral clustering. Given the matrix S, we then compute the degree matrix $D = \mathsf{diag}(D_{11}, \cdots, D_{nn})$, such that $D_{ii} = \sum_{j=1}^{n} S_{ij}$.

Recall that the unnormalized graph Laplacian $L_{\mathsf{un}} = D - S$. We then perform spectral analysis of L_{un} and let

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m,$$

where $\lambda_1, \cdots, \lambda_m$ are the m-smallest *non-zero* eigenvalues. of L_{un}. Let $u_1, \cdots, u_m \in \mathbb{R}^n$ be the corresponding eigenvectors.

We then map $X_i \in \mathbb{R}^d$ into $Y_i = (u_{1,i}, \cdots, u_{m,i})^T \in \mathbb{R}^m$. The coordinate $Y_1, \cdots, Y_n \in \mathbb{R}^m$ will be the reduced dimension coordinate.

The intuition of Laplacian eigenmap is based on the following fact of smallest eigenvalues/eigenvectors. The eigenvector u_1 can be constructed based on the following process:

$$u_1 = \mathsf{argmin}_v v^T L_{\mathsf{un}} v \text{ s.t. } v^T D v = 1, \quad v^T D \mathbf{1} = 0.$$

The constraint $u^T D \mathbf{1} = 0$ is due to the fact that the 0 eigenvalue has an eigenvector $\frac{1}{\sqrt{n}}\mathbf{1} = \frac{1}{\sqrt{n}}(1, 1, \cdots, 1)^T$.

As is argued in the spectral clustering lecture, the graph Laplacian is approximating a Laplacian operator on a manifold. If the observations $X_1, \cdots, X_n$ are uniformly distributed over a manifold $\mathcal{M}$, then

$$v^T L_{\mathsf{un}} v \approx \int_{\mathcal{M}} \|\nabla f(x)\|^2 dx,$$

where $v_i = f(X_i)$. Thus, eigenvector via minimization process will try to pick values that are slowly changing along the manifold. Namely, for pairs X_i, X_j that are close on the manifold, the corresponding values $v_i = f(X_i), v_j = f(X_j)$ will also be close. So the eigenvetor u_1 will try to preserve such local information when representing observations in a new coordinate. A similar argument also applies to other eigenvectors.

To sum up, the Laplacian eigenmap consists of the following three steps:

1. *Construction of similarity matrix.* We compute the similarity matrix $S \in \mathbb{R}^{n\times n}$ for any pair of observations and compute the unnormalized graph Laplacian $L_{\mathsf{un}} = D - S$.
2. *Eigen-decomposition.* We apply eigen-decomposition to L_{un} to obtain $u_1, \cdots, u_m$ that are eigenvectors corresponding to the m-smallest non-zero eigenvalues.
3. *Dimension reduction.* We represent observation X_i as $Y_i = (u_{1,i}, \cdots, u_{m,i})^T \in \mathbb{R}^m$.

Remark. In spectral clustering, we choose m to be the final number of clusters and apply a k-means clustering based on the reduced dimension (also m). Laplacian eigenmap does not have the k-means step but just use the m reduced dimensional coordinate for further analysis.

10.6.2 Diffusion map

Diffusion map is another popular dimension reduction technique based on the spectral information. It is proposed in the following paper:

> [CL2006] Coifman, R. R., & Lafon, S. (2006). Diffusion maps. Applied and computational harmonic analysis, 21(1), 5-30.

Instead of using unnormalized Laplacian, we consider the random walk Laplacian $L_{\mathsf{RW}} = D^{-1}S$.

The random walk Laplacian has an elegant interpretation: the matrix L_{RW} is a transition probability matrix of a Markov chain over observations $X_1, \cdots, X_n$. The transition probability

$$P(i \to j) = L_{\mathsf{RW,ij}} = \frac{S_{ij}}{\sum_{k=1}^n S_{ik}}.$$

Now suppose that we construct our similarity matrix using some kernel function, i.e., $S_{ij} = K(X_i, X_j)$, for some kernel function K. Then the transition probability can be further written as

$$P(i \to j) = \frac{S_{ij}}{\sum_{k=1}^n S_{ik}} = \frac{K(X_i, X_j)}{\sum_{k=1}^n K(X_i, X_k)}.$$

In the continuous limit, a popular analogous to the above transition probability is the transition kernel

$$p(x \to y) = q(y|x) = \frac{K(x,y)}{\int K(x,z)dP(z)} = \frac{K(x,y)}{s(x)},$$

where P is the CDF that generates our observation. The transition kernel $q(y|x)$ defines a continuous time Markov chain (a diffusion process) with a stationary distribution/density $\pi(x) = \frac{s(x)}{\int s(x)dP(x)} \propto s(x)$. One can easily see that when K is symmetric, such transition kernel satisfies the detailed balanced, i.e.,

$$\pi(x)q(y|x) = \frac{s(x)}{\int s(x)dP(x)}\frac{K(x,y)}{s(x)} = \frac{K(x,y)}{c} = \frac{K(y,x)}{c} = \frac{s(y)}{\int s(x)dP(x)}\frac{K(y,x)}{s(y)} = \pi(y)q(x|y),$$

where $c = \int s(x)dP(x)$. So the Markov chain eventually converges to the stationary distribution.

Suppose that we run the Markov chain for time t, this leads to a transition kernel $q_t(y|x)$ (which has a sample analogue $[L^t_{\mathsf{RW}}]_{ij}$). Using the transition kernel, we define the *diffusion distance*

$$D_t(x,y) = \int (q_t(u|x) - q_t(u|y))^2 \frac{1}{\pi(u)} dP(u). \tag{10.2}$$

The diffusion distance measures the distance between x and y in terms of the diffusion process starting at x versus y. If the two points are close on the manifold, we expect that their diffusion process will be similar so the distance will be small.

In [CL2006], the authors further showed that the diffusion distance can be written as follows:

$$D_t(x,y) = \sqrt{\sum_{k=1}^{\infty} \lambda_k^{2t} (\psi_k(x) - \psi_k(y))^2}, \tag{10.3}$$

where λ_k is the k-th eigenvalue of $q(y|x)$, i.e.,

$$\int q(y|x)\psi_k(y)dy = \lambda_k \psi_k(x) s(x)$$

and $\lambda_0 = 1 \geq |\lambda_1| \geq |\lambda_2| \geq \cdots$. Thus, instead of computing the integral in equation (10.2), we can use equation (10.3) to compute the diffusion distance.

Finally, we represent the point $x \in \mathbb{R}^d$ using the coordinate

$$\Psi_m(x) = (\lambda_1^t \psi_1(x), \cdots, \lambda_m^t \psi_m(x))^T \in \mathbb{R}^m.$$

This is the reduced dimension version of x.

In the sample version, the diffusion distance between observations X_i and X_j is

$$D_t(i,j) = \frac{1}{\rho} \sum_{k=1}^{n} D_{kk}^{-1} \left([L^t_{\mathsf{un}}]_{ik} - [L^t_{\mathsf{un}}]_{jk} \right)^2, \tag{10.4}$$

where $\rho = \mathsf{Tr}(D)$. The quantity L^t_{un} is the matrix L_{un} raised to the power of t (t-step transition probability matrix). Note that ρ is analogous to the the normalizing constant c and the degree matrix D is a sample analogue of $s(x)$. And similar to equation (10.3), we can express the diffusion distance as

$$D_t(i,j) = \sqrt{\sum_{k=1}^{n} \lambda_k^{2t} (\psi_{k,i} - \psi_{k,j})^2}, \tag{10.5}$$

where (λ_k, ψ_k) is the k-th largest absolute eigenpair of L_{un} with $\psi_k \in \mathbb{R}^n$.

The embedded location of X_i is then

$$Y_i = (\lambda_1^t \psi_{1,i}, \cdots, \lambda_m^t \psi_{m,i})^T \in \mathbb{R}^m. \tag{10.6}$$

10.7 Random projection

Random projection (RP) is an elegant and powerful method for dimension reduction. A very impressive property of RP is that it can almost preserve pairwise distance between high dimensional observations with only a few dimensions.

Let $X_1, \cdots, X_n \in \mathbb{R}^d$ be the original observations. Note that here d is large. For any point $x \in \mathbb{R}^d$, we define a random projection onto $\mathbb{R}^m (m < d)$ using

$$L(x) = \frac{Sx}{\sqrt{m}},$$

where $S \in \mathbb{R}^{m \times d}$ is a projection matrices whose elements are IID from $N(0,1)$.

With a given projection L, we define the projected points

$$Y_1, \cdots, Y_n \in \mathbb{R}^m, \qquad Y_i = L(X_i).$$

So the new observations $Y_1, \cdots, Y_n$ are random projected version of $X_1, \cdots, X_n$ in the m dimensional space.

The following famous Johnson-Lindenstrauss Theorem shows that as long as m is not too small, we can preserve the pairwise distance with a high probability.

Theorem 10.1 (Johnson-Lindenstrauss) *Fixed ϵ. Let $L(x)$ be the above random projection. Suppose $m \geq 32 \frac{\log n}{\epsilon^2}$. Then with a probability of at least $1 - e^{-m\epsilon^2/16} \geq 1 - \frac{1}{n^2}$, we have*

$$(1-\epsilon)\|X_i - X_j\|^2 \leq \|Y_i - Y_j\|^2 \leq (1+\epsilon)\|X_i - X_j\|^2$$

uniformly for all i, j.

Note that the original dimension d plays no role in the above theorem!

Essentially, Theorem 10.1 shows that if use L to project a d-dimensional data onto m-dimensional subspace, we can almost preserve the pair-wise distance as long as we choose m to be of the order of $\log n$.

Proof:

Consider any pair i, j,

$$\begin{aligned}\frac{\|Y_i - Y_j\|^2}{\|X_i - X_j\|^2} - 1 &= \frac{\|S(X_i - X_j)\|^2}{m\|X_i - X_j\|^2} - 1 \\ &= \frac{1}{m}\sum_{\ell=1}^{m} \frac{\|S_\ell^T (X_i - X_j)\|^2}{\|X_i - X_j\|^2} - 1,\end{aligned}$$

where S_ℓ is the ℓ-th row of S. Because every element in S is from IID $N(0,1)$, $\frac{\|S_\ell^T (X_i - X_j)\|^2}{\|X_i - X_j\|^2} = Z_\ell^2$ and $Z_1, \cdots, Z_m$ are IID $N(0,1)$. Thus,

$$\frac{\|Y_i - Y_j\|^2}{\|X_i - X_j\|^2} - 1 = \frac{1}{m}\sum_{\ell=1}^{m} Z_\ell^2 - 1. \tag{10.7}$$

In what follows, we will use the conventional approach of deriving a large deviation bound in this case.

Since $Z_\ell^2 \sim \chi_1^2$, $\mathbb{E}(Z_\ell^2) = 1$ and the moment generating function of Z_ℓ^2 is $\phi_{Z_\ell^2}(t) = \frac{1}{\sqrt{1-2t}}$ for $t < \frac{1}{2}$ (property of a χ_1^2 distribution). Moreover, for any small t, we have

$$\mathbb{E}(e^{t(Z_\ell^2 - 1)}) = \frac{e^{-t}}{\sqrt{1-2t}} \leq e^{2t^2}$$

Using the fact that $Z_1, \cdots, Z_m$ are IID, we conclude that

$$\mathbb{E}(e^{t\sum_{\ell=1}^m (Z_\ell^2-1)}) = \prod_{\ell=1}^m \mathbb{E}(e^{t(Z_\ell^2-1)}) \le e^{2mt^2}.$$

Thus, by the Markov inequality,

$$\begin{aligned} P\left(\frac{1}{m}\sum_{\ell=1}^m Z_\ell^2 - 1 \ge \epsilon\right) &= P\left(\sum_{\ell=1}^m Z_\ell^2 - 1 \ge m\epsilon\right) \\ &= P\left(\exp\left(\sum_{\ell=1}^m t(Z_\ell^2-1)\right) \ge e^{tm\epsilon}\right) \\ &\le \mathbb{E}\left(\exp\left(\sum_{\ell=1}^m t(Z_\ell^2-1)\right)\right) e^{-tm\epsilon} \\ &\le e^{2mt^2 - tm\epsilon} \end{aligned}$$

for any t. So we solve for the optimal $t = t^* = \frac{\epsilon}{4}$, which leads to the bound

$$P\left(\frac{1}{m}\sum_{\ell=1}^m Z_\ell^2 - 1 \ge \epsilon\right) \le e^{-m\epsilon^2/8}.$$

A similar bound can be derived for

$$P\left(\frac{1}{m}\sum_{\ell=1}^m Z_\ell^2 - 1 \le -\epsilon\right) \le e^{-m\epsilon^2/8}.$$

As a result, equation (10.7) leads to

$$\begin{aligned} P\left(\left|\frac{\|Y_i - Y_j\|^2}{\|X_i - X_j\|^2} - 1\right| > \epsilon\right) &= P\left(\left|\frac{1}{m}\sum_{\ell=1}^m Z_\ell^2 - 1\right| > \epsilon\right) \\ &\le 2e^{-m\epsilon^2/8}. \end{aligned}$$

Using the union bound trick, we can easily generalize the bound to

$$P\left(\max_{i\neq j}\left|\frac{\|Y_i - Y_j\|^2}{\|X_i - X_j\|^2} - 1\right| > \epsilon\right) \le 2n^2 e^{-m\epsilon^2/8} \le e^{-m\epsilon^2/16}$$

when $m \ge 32\log n/\epsilon^2$.

■

10.8 t-SNE

t-distributed stochastic neighbor embedding (t-SNE) is a very popular dimension reduction and visualization technique that is widely used in many applied science. It was proposed in the following paper:

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

It often leads to elegant visualization of a high dimensional data with clustering structure, so the choice $m = 2$.

Note: it is designed mainly for visualization, which is only part of the objective of dimension reduction. Making analysis based on the reduced dimensional data from t-SNE may not be better than other methods we have mentioned.

The t-SNE starts with defining a similarity between pairs of observations using Gaussian kernels and transition. Similar to the random walk Laplacian, we define the transition probability from $X_i \in \mathbb{R}^d$ to $X_j \in \mathbb{R}^d$ as

$$p(i \to j) = \frac{\exp(\frac{\|X_i - X_j\|^2}{2\sigma^2})}{\sum_{k \neq i} \exp(\frac{\|X_i - X_k\|^2}{2\sigma^2})}$$

and set $p(i \to i) = 0$ We symmetrize this quantity to obtain a similarity measure

$$p_{ij} = \frac{1}{2n}(p(i \to j) + p(j \to i)).$$

Note that this similarity measure has the property that

$$\sum_{i,j} p_{ij} = 1.$$

We then consider finding 2-dimensional representation points $Y_1, \cdots, Y_n \in \mathbb{R}^2$ but we measure their transition using t-distribution kernel with degree of freedom 1 (Cauchy distribution), i.e.,

$$q_{ij} = \frac{(1 + \|Y_i - Y_j\|^2)^{-1}}{\sum_{k \neq \ell}(1 + \|Y_k - Y_\ell\|^2)^{-1}}$$

and set $q_{ii} = 0$. Note that the denominator of $q(i \to j)$ is summation over all pairs so q_{ij} itself is already symmetric.

Similar to p_{ij}, we have $\sum_{i,j} q_{ij} = 1$. Thus, both $p \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^{n \times n}$ can be viewed as a distribution. We then measure their difference using the KL-divergence, i.e.,

$$L(Y_1, \cdots, Y_n) = \sum_{i,j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right). \tag{10.8}$$

We then search for $Y_1, \cdots, Y_n$ that minimizes the above loss function. The minimization of $L(Y_1, \cdots, Y_n)$ is done by a random initialization of $Y_1, \cdots, Y_n$ and then applying gradient descent of each component.

The main motivation of t-SNE is from the crowding problem. For a point $X_i \in \mathbb{R}^d$, consider the ball $B(X_i, r) \subset \mathbb{R}^d$. The number of observation within this ball is at the order of r^d. Thus, there will be around $O(r^d)$ observations with a value of $p_{ij} \sim \exp(r^2/\sigma^2)$. If we want to use the same normal kernel to reduce the dimension in $m = 2$, we cannot squeeze into $O(r^d)$ observations into the area $B(Y_i, r) \subset \mathbb{R}^2$ because of the low dimensional nature. However, if we replace the normal kernel with a heavier tail kernel (such as Cauchy tail/t-distribution tail), we can use a larger radius to put in a similar amount of observation because the heavy-tail kernel function decays slower than the Gaussian.