Machine Learning for Applied Physics and High Energy Physics

Lecture 19 Dimensional reduction and data visualisation

In this last module of the course we will deal with unsupervised learning. Remember that supervised learning consists in determining a mapping f such that $f: \vec{x}_n \longrightarrow \vec{y}_n$ where $\mathcal{D} = \{(\vec{x}_n, \vec{y}_n)\}$ $n = 1, \dots, N$ is a collection of pairs features - labels. In unsupervised ML, the data set is not labelled. Supervised ML has indeed some limitations.

1) Need of labelled data. Labelled data is hard to get.
2) Determining a mapping is very computationally intensive and data intensive. This becomes hard if the dataset is limited
3) The data must be homogeneous. It is harder to mix and match different data types.
4) Many physics problems are not about prediction. We want to learn something about the underlying distribution that generates the data.

Unsupervised learning is concerned with discovering structure in unlabelled data. In this lecture we start our journey into unsupervised ML by discussing dimensional reduction. The goal is to identify correlated or redundant features along with irrelevant features (noise). This can be done efficiently only if the data set is embedded or projected onto a lower dimensional space, called LATENT SPACE. Information loss must be limited to a minimum.

High dimensional data have some challenges.

a High-dimensional data lives near the edge of sample space. Example. Consider data distributed uniformly at random in a D-dimensional hypercube $\mathcal{C} = [-e/2, e/2]^D$ where e is the edge length. Consider also a D-dimensional hypersphere $\mathcal{S}$ of radius $e/2$ centered at the origin and contained in $\mathcal{C}$. The probability that a date point $\vec{x}$ drawn uniformly at random in $\mathcal{C}$ is contained in $\mathcal{S}$ is approximated by the ratio of the volume of $\mathcal{S}$ to $\mathcal{C}$:

$$p(\|\vec{x}\|_2 < e/2) \sim (1/2)^D$$

Therefore, as $D \to \infty$, $p \to 0$ (exponentially). Most of the data will concentrate outside the hypersphere, at the corners of the hypercube.

b Real-world data vs. uniform distribution. Real-world data is not random or uniformly distributed. Real data usually leave in a lower dimensional space than the original space in which the features are measured. This is sometimes called "blessing of non uniformity". A local variation of the data will not incur in a change of the target variable. The data can be described by low-dimensional "order parameters" or effective degrees of freedom (as a gas).

c Intrinsic dimensionality and the crowding problem. The objective is to preserve the relative pairwise distance between date points from the original space to the latent space. Nearby points remain close. Example: the Swiss Roll. Each point (in cylindrical coordinates) is

$$P = (x_1 \cos\vartheta;\ x_1 \operatorname{sen}\vartheta;\ x_2)$$

The distance between two points is

$$d_{PP'} = \sqrt{(x_1\cos\vartheta - x_1'\cos\vartheta)^2 + (x_1\sin\vartheta - x_1'\sin\vartheta)^2 + (x_2 - x_2')^2}$$

$$= \sqrt{(x_1 - x_1')^2\cos^2\vartheta + (x_1 - x_1')^2\sin^2\vartheta + (x_2 - x_2')^2}$$

$$= \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2}$$

The critical dimension is two (or intrinsic dimension).
If one attempts to represent the data in a space of dimensionality lower thant its intrinsic dimension, he incurs in an overcrowding problem.

13.1 Principal component analysis (PCA).

The goal of PCA is to perform an orthogonal transformation of the data in order to find high-variance directions. In many cases, the relevant information in a signal is contained in the direction of larger variance.

Let us consider N data points that live in a p-dimensional feature space

$$\mathcal{D} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\} \qquad \vec{x}_i \in \mathbb{R}^p \quad i = 1, \dots, N$$

Let's assume, without loss of generality, that the empirical mean of the dataset is zero:

$$\bar{\vec{x}} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i = \vec{0}$$

Let us denote the $N \times p$ design matrix $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N]^T$
(the rows are the data points; the colums are the features)
The $p \times p$ (symmetric) covariance matrix is

$$\Sigma(X) = \frac{1}{N-1} X^T \cdot X$$

The j-th diagonal entry of $\Sigma(X)$ corresponds to the variance of the j-th feature and $\Sigma_{ij}(X)$ is the matrix element that measures the covariance between the features i and j.

We are now interested in finding a new basis for the data that emphasises highly variable directions while reducing redundancy between basis vectors. In particular we look for a linear transformation that reduces the covariance between different features. To do so, we perform singular value decomposition on the design matrix X

$$X = USV^T$$

where S is a diagonal matrix of singular values s_i, U is an orthogonal matrix that contains (as columns) the left singular vectors of X and V contains the right singular vectors. One can the rewrite the covariance matrix as

$$\Sigma(X) = \frac{1}{N-1} VS\underbrace{U^T U}_{\mathbb{1} \text{ orthogonal}} SV^T$$

$$= \frac{1}{N-1} V S^2 V^T = V \Lambda V^T \quad \text{with } \Lambda = \frac{S^2}{N-1}$$

Λ is a diagonal matrix with eigenvalues λ_i in decreasing order. The right singular vectors of X (i.e. the colums of V) are principal directions of $\Sigma(X)$ and the singular values of X are related to the eigenvalues of $\Sigma(X)$ as $\lambda_i = \frac{s_i^2}{N-1}$.

To reduce the dimensionality of data from p to $\tilde{p} < p$ we first construct the projection matrix $\tilde{V}_{p'}$ by selecting the singular components with the $\tilde{p}$ largest singular values. The projection of the data from p to $\tilde{p}$ dimensions is

$$\tilde{Y} = X \tilde{V}_{p'}$$

The singular vector with the largest singular value is referred to as the first principal component, and so on. An important quantity is the ratio

$$\frac{\lambda_i}{\sum_{i=1}^{r} \lambda_i}$$

which is referred to as the percentage of the explained variance contained in a principal component

Only the first, more relevant, principal components are used to approximate X into $\hat{Y}$.

13.2 Multidimensional scaling (MDS)

Multidimensional scaling is a non-linear dimensional reduction technique which preserves the distance (or dissimilarity) d_{ij} between data points. There are two types of MDS: metric and non-metric. In metric MDS, the distance is computed under a pre-defined metric and the latent coordinates $\hat{Y}$ are obtained by minimizing the distance measured in the original space $d_{ij}(X)$ and in the latent space $d_{ij}(Y)$

$$\hat{Y} = \underset{Y}{\operatorname{argmin}} \sum_{i<j} w_{ij} \left| d_{ij}(X) - d_{ij}(Y) \right|$$

where $w_{ij} \geq 0$ are weight values. The weight matrix is a set of free parameters w_{ij} that specify the level of confidence (or precision) in the value of $d_{ij}(X)$. If the Euclidean metric is used, PCA is recovered. Therefore metric MDS is considered as a generalisation of PCA.

In non-metric MDS, d_{ij} can be any distance matrix. The objective function is then to preserve the ordination in the data, i.e. if $d_{12}(X) < d_{13}(X)$ in the original space, then in the latent space we should have $d_{12}(Y) < d_{13}(Y)$.

<u>Observations</u>

1) We have defined the PCA based on the covariance matrix $\Sigma'(X)$. Sometimes it is better to work with the correlation matrix ρ, which is uniform w.r.t. the scale.
2) The choice of latent dimensions (or principal components with which one can approximate the original space) typically entails some arbitrariness. A common way is to look at the scree plots (where the eigenvectors of the covariance matrix are displayed in decreasing order).
3) PCA scales as $O(Np^2 + p^3)$; Np^2 is due to the decomposition of the covariance matrix; p^3 stems from eigenvalue decomposition.

   MDS scales as $O(N^3)$, therefore it is limited to applications to small data sets.