1 Dimensionality Reduction

There are many issues with working in high dimensions. As the dimension d grows, machine learning algorithms can become more computationally intensive. It also becomes more difficult to visualize our data - humans are notoriously bad at visualizing beyond 3 dimensions. Additionally, redundant features can add more noise than signal. There is a widely held belief that most natural data in high dimensions (for example, data used in genomics) can be represented in a lower dimensional space.

Dimensionality reduction is an *unsupervised* method - unlike supervised learning, there are no labels that we need to match, no classes to predict. As such, defining problems becomes more subjective and heuristic. One approach to dimensionality reduction is feature selection, in which we remove features that we deem to be irrelevant based on some criteria. For example, the LASSO provides this feature selection using L^1 regularization.

Another approach to dimensionality reduction is learning latent features. This approach seeks to find new latent features that are transformations of our given features that represent the data well.

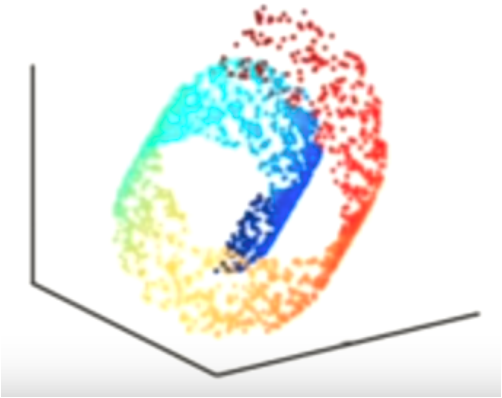

Figure 1: The Swiss Roll dataset is a 2-dimensional manifold embedded in 3 dimensional space. We should be able to represent it with a 2-dimensional feature space.

1.1 Principal Components Analysis

PCA can be used for dimensionality reduction. Recall that finding the PCA decomposition of $X \in \mathbb{R}^{n \times D}$ amounts to finding the SVD

$$X = USV^T$$

Suppose $d << D$, and let $\tilde{U}, \tilde{S}, \tilde{V}$ contain the first d columns of U, S, V respectively. Then $\tilde{V}^T x$ projects data x into a d-dimensional latent space. This projection into a lower-dimensional space will typically cause some loss of information, but if the data really does live in a d-dimensional subspace, then we should be able to reconstruct x via $\tilde{V}\tilde{V}^T x$ and obtain something that is close to the original x. PCA can be seen as finding a basis for a subspace that minimizes this reconstruction error (low-rank approximation property).

Sometimes, it does not make sense to find orthogonal directions that capture the maximum variance, as PCA does. Independent Components Analysis instead seeks directions that are statistically independent, and is more suitable for some applications.

1.2 Nonnegative Matrix Factorization

NMF takes a non-negative matrix X, where each column is a data entry, and approximately factors it into $X = BH$, where B is skinny, H is fat, and both these matrices are non-negative. The k-th column of X is the sum of the columns of B, weighted according to the entries in the k-th column of H. In this regard, each column of B can be seen as a feature that contributes to the data in a meaningful way - the number of columns in B is the number of features we are allowed to reconstruct the data with. If each column of X is a vectorized image, then each column of B will be a vectorized image. NMF learns a part-based representation of the data, since the reconstruction is a non-negative linear combination of features of the same dimension as the data. It turns out that NMF will tend to produce sparser features than PCA. NMF has applications in computer vision and recommender systems, among other fields.

1.3 Multidimensional Scaling

MDS seeks to learn a low-dimensional representation such that pairwise distances between points are exactly preserved in the latent representation. Specifically, if $X_i \in \mathbb{R}^D$ and $W_{ij} = \|X_i - X_j\|^2$, then MDS finds Y_i such that $\|Y_i - Y_j\|^2 \approx W_{ij}$. More generally, one can use any dissimilarity measure $d(X_i, X_j)$ in place of the norm. If the dissimilarity measure is a metric, then MDS is equivalent to PCA. MDS is typically used to visualize high dimensional data.

Note that while MDS captures interpoint distances in its low-dimensional embedding, it is incapable of capturing local geometric structure - for example, in the Swiss roll dataset shown above, points in the red region are dissimilar from points in the blue region, but are relatively close to these points in distance, so MDS will provide a representation where red and blue points are not separated.

1.4 Nonlinear Methods

Linear methods such as PCA are not well-suited to capture intrinsic geometric structure in data. There are several approaches to solving this problem:

- Kernel methods: it is possible to derive a kernelized version of PCA, for example.

- Manifold learning: an n-manifold[1] is a surface that locally resembles n-dimensional Euclidean space. For example, the Swiss roll is a 2-manifold embedded in 3-dimensional space. In manifold learning, we learn a mapping from data to a low-dimensional manifold - for example, a mapping from Swiss roll to 2-dimensional plane.

1.5 Isometric Feature Mapping (IsoMap)

IsoMap performs MDS on the geodesic distances between points, as follows:

(1) Construct a local neighborhood graph by connecting each point with its k nearest neighbors.

(2) Compute all-pairs shortest path distances (these are the geodesic distances).

(3) Apply MDS on geodesic distances.

If we apply IsoMap to the Swiss roll, we see that while the Euclidean distance between the red and blue regions is low, the geodesic distance is high - the geodesic distance between points represents how far we would have to go if we were walking on the Swiss roll manifold in 2 dimensions.

IsoMap has been used effectively for dimensionality reduction in facial data.

Next time: Laplacian Eigenmaps, t-SNE.

[1] For the mathematically inclined, a manifold is a second-countable Hausdorff topological space such that each point has a neighborhood homeomorphic to a neighborhood in Euclidean space. Most machine learning researchers do not care for these definitions and will call almost anything a manifold.