CS/CNS/EE 253: Advanced Topics in Machine Learning
Topic: Dimensionality Reduction **Lecturer:** Andreas Krause
Scribe: Matt Faulkner **Date:** 25 January 2010

6.1 Dimensionality reduction

Previously in the course, we have discussed algorithms suited for a large number of data points. This lecture discusses when the dimensionality of the data points becomes large. We denote the data set as $x_1, x_2, \ldots, x_n \in \mathbb{R}^D$ for $D >> n$, and will consider dimensionality reductions $f : \mathbb{R}^D \to \mathbb{R}^d$ for $d << D$. We would like the function f to preserve some properties of the original data set, such as variance, correlation, distances, angles, or "clusters". For a concrete example, consider consider linear functions,

$$f(x) = Ax \ , \ A \in \mathbb{R}^{d \times D}$$

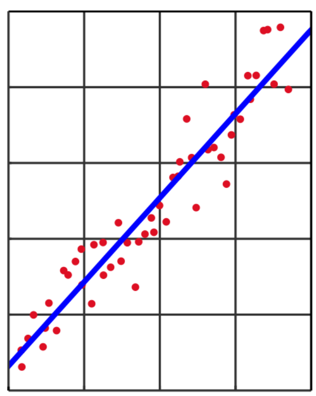

Figure 6.1.1: Reduce from $\mathbb{R}^2$ to $\mathbb{R}^1$

Figure 6.1.2 shows projection onto a line to preserve the variance of the data set. Specifically, when projecting from $\mathbb{R}^2$ to $\mathbb{R}^1$, we can take A to be a unit vector e, and consider the projection x' of a data point x onto e

$$\begin{aligned} x' &= \langle x, e \rangle \cdot e \\ c &= ||x||_2 \\ a &= \langle x, e \rangle \\ b &= \sqrt{c^2 - a^2} \end{aligned}$$

We want to choose e in order to maximize a, or equivalently, to minimize b. We will see that

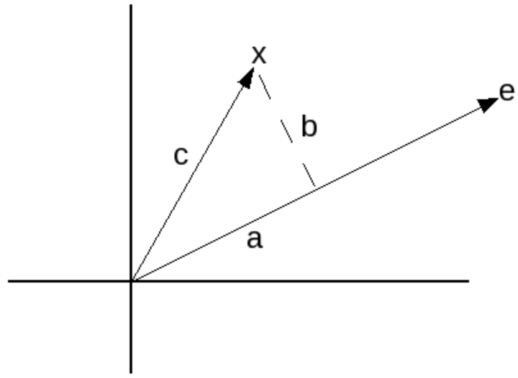

Figure 6.1.2: Choosing e to maximize the variance of the projection maximizes the a component over all data points; equivalently, choosing e to minimize reconstruction error minimizes b for all data points.

"Maximize variance" $\leftrightarrow$ "Minimize reconstruction error." Principal component analysis (PCA) [3] is one such technique for projecting inputs onto a lower dimensional space so as to maximize variance. The desired orthogonal projection matrix $A \in \mathbb{R}^{d\times D}$ can be expressed as

$$\begin{aligned} A^* &= \operatorname*{argmin}_{A=(e_1,\ldots,e_j)\in\mathbb{R}^{d\times D}} \quad \sum_i \left\| x_i - \sum_j \langle x_i, e_j\rangle \cdot e_j \right\|_2 \\ &= \operatorname*{argmax}_{A=(e_1,\ldots,e_j)\in\mathbb{R}^{d\times D}} \quad \sum_i \left\| \sum_j \langle x_i, e_j\rangle \cdot e_j \right\|_2 \\ &= \sum_j \sum_i \langle x_i, e_j\rangle^2 \end{aligned}$$

Thus we see that this optimization has a closed-form solution. Now, assume $d = 1$.

$$\max_{||e_x||\leq 1} \sum_i \langle x_i, e_1\rangle^2$$

$$\max_{||e_x||\leq 1} e_1^T \sum_i^n x_i \cdot x_i^T e_1$$

Noting that $\sum_i = 1^n x_i \cdot x_i^T = n \cdot \mathrm{Cov}(X)$ and letting C denote the covariance matrix, we get that the maximization can be expressed in terms of the Rayleigh quotient,

$$\max_{e_1} \frac{e_1^T C e_1}{e_1^T e_1}$$

This is maximized by selecting e_1 as the eigenvector corresponding to the largest eigenvalue.

For $d-$dimensional projection ($d > 1$), if we let $e_1, \dots, e_D$ denote the eigenvectors of the covariance matrix C corresponding to eigenvalues $\lambda_1 \geq \dots \geq \lambda_D$, then the $d-$dimensional projection matrix is given by

$$A^* = (e_1, \dots, e_d)$$

6.2 Preserving distances

We would like to produce a faithful reduction, in that nearby inputs should be mapped to nearby outputs in lower dimensions. Motivated by this, we can formulate an optimization that seeks for each x_i a $\psi_i \in \mathbb{R}^d$ s.t.

$$\min_{\psi} \sum_{i,j} \left(||x_i - x_j||^2 - ||\psi_i - \psi_j||^2\right)$$

Intuitively, this minimizes the "stress" or the "distortion" of the dimension reduction. This optimization has a closed-form solution. Further, preserving distances turns out to be equivalent to preserving dot products:

$$\min_{\psi} \sum_{i,j} \left(\langle x_i - x_j\rangle - \langle \psi_i - \psi_j\rangle\right)^2$$

This is convenient because the reduction can be formulated for anything with a dot product, allowing the use of kernel tricks. This optimization, known as multidimensional scaling (MDS) [4] [2], also has a closed form solution. Define $S_{i,j} = ||x_i = x_j||^2$ to be the matrix of squared pairwise distances in the input space. We can then define the Gram matrix $G_{i,j} = \langle x_i, x_j\rangle$. This matrix can be derived from S as

$$G = \frac{1}{2}(I - uu^T)S(I - uu^T)$$

where u is the unit length vector

$$u = \frac{1}{\sqrt{n}}(1, \dots, 1)$$

Here, the terms $(I = uu^T)$ have the effect of subtracting off the means of the data poitns, matking G a covariance matrix. The optimal solution ψ^* is computable from the eigenvectors of the Gram matrix. Letting $(v_i, \dots, v_n)$ be the eigenvectors of G with corresponding eigenvalues $(\mu_1, \dots, \mu_n)$, the optimum ψ results from projecting each data point x_i onto the d (scaled) eigenvectors $\sqrt{\mu_1}v_1, \dots, \sqrt{\mu_d}v_d$.

$$\psi^*_{i,j} = \sqrt{\mu_j}v_j \cdot x_i$$

For the optimal solution ψ^* it holds that $\psi^* = A_{PCA}x_i$, so PCA and MDS give equivalent results. However, PCA uses $C = XX^T \in \mathbb{R}^{DxD}$, while DS uses $G = X^TX \in R^{nxn}$. These computational differences are summarized in Table 1. MDS also works whenever a distance metric exists between the data objects, so the objects are not required to be vectors.

6.3 Isomap: preserving distances along a manifold

Suppose that the data set possesses a more complex structure, such as for the "Swiss Roll" in Fig. 6.3.3. Linear projections would do poorly, but it is clear that the data live in *some* lower-dimensional space.

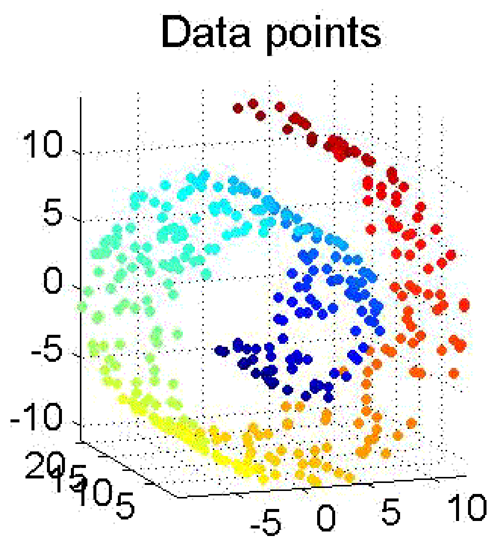

Figure 6.3.3: The Swiss Roll data set.

A key insight is that within a small neighborhood of points, a linear method works well. We would like these local results to be consistent. One algorithm which formalizes this idea is Isomap [6]. Roughly, the algorithm performs three computations:

1. Construct a graph hby connecting k nearest neighbors, as in Fig. 6.3.4.
2. Define a metric $d(x_i, x_j)$ = length of shortest path on graph from x_i to x_j. This is the "geodesic distance".
3. Plug this distance metric into MDS.

The result is that the distance between two points A and B on the roll respects the "path" through the manifold of data.

6.4 Maximal Variance unfolding

Maximal variance unfolding (MVU) [7] is another dimensionality reduction technique that "pulls apart" the input data while trying to preserve distances and angles between nearby data points. Formally, it computes

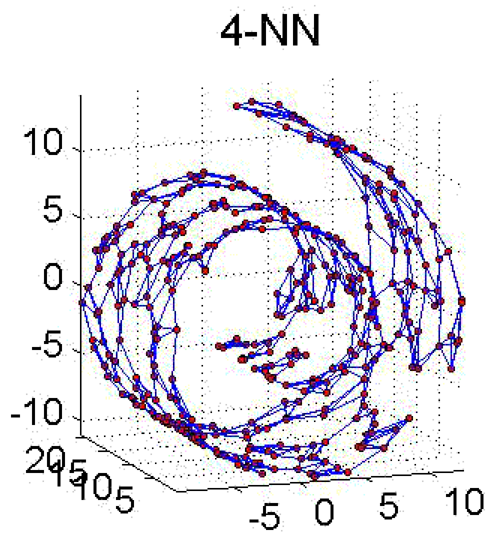

Figure 6.3.4: Graph of four nearest neighbors on the Swiss Roll.

Algorithm	Primary computation	Complexity
PCA	$C = XX^T \in \mathbb{R}^{DxD}$	nD^2
MDS	$G = X^T X \in \mathbb{R}^{nxn}$	$n^2 D$
ISOMAP	geodesics and MDS	$n^2 \log n + n^2 D$
MVU	semidefinite programming	n^6

Table 1: Computational efficiency of dimensionality reduction algorithms.

$$\begin{aligned} \max_X \quad & \sum_i ||\psi_i||_2^2 \\ \text{subject to} \quad & ||\psi_i||_2^2 = ||x_i - x_j||^2 \\ & \sum_i \psi_i = 0, \; \forall\, i, j \end{aligned}$$

A closed-form solution for this problem does not exist, but it can be optimally solved using semidefinite programming.

6.5 Computational Summary

Table 1 summarizes the computational complexity of the dimensionality reduction algorithms considered so far. Notably, the choice between PCA and MDS may depend on the relative size of the input n and its dimensionality D.

6.6 Random projections

For comparison, what if A is picked at random? For linear dimension reduction only, consider choosing the entries of the matrix A as

$$A_{i,j} \sim N(0,1)$$

or

$$A_{i,j} = \begin{cases} +1 \text{ with prob } \frac{1}{2} \\ -1 \text{ with prob } \frac{1}{2} \end{cases}$$

Somewhat surprisingly, this A can work well.

Theorem 6.6.1 (Johnson & Lindenstrauss) *Given n data points, for any $\epsilon > 0$ and $d = \Theta(\epsilon^{-2} \log n)$, with high probability*

$$(1-\epsilon)||x_i - x_j||^2 \leq ||Ax_i - Ax_j|| \leq ||x_i - x_j||(1+\epsilon)$$

6.7 Further reading

A few good surveys on dimensionality reduction exist, such as those by Saul [5], and Burges [1].

References

[1] Christopher J. C. Burges. Geometric methods for feature extraction and dimensional reduction. In *In L. Rokach and O. Maimon (Eds.), Data*. Kluwer Academic Publishers, 2005.

[2] TF Cox and MAA Cox. Multidimensional Scaling. Number 59 in Monographs on statistics and applied probability. *Chapman & Hall. Pages*, 30:31, 1994.

[3] IT Jolliffe. *Principal component analysis*. Springer verlag, 2002.

[4] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.

[5] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee. Spectral methods for dimensionality reduction. *Semisupervised Learning. MIT Press, Cambridge, MA*, 2006.

[6] J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[7] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.