Manifold Learning & Random Projections

Aman Dhesi

Special Interest Group in Machine Learning
IIT Kanpur

February 13, 2010

Outline

Why reduce dimensionality?

"The curse of dimensionality"

- Massive, high-dimensional data sets
- Many widely used machine learning techniques scale exponentially with data dimension
- Difficult to visualize anything above 3D, difficult to find meaningful representation
- Features may be correlated, redundant, causing data to have *low intrinsic dimension*

Why reduce dimensionality?

An example of superficially high-dimensional data - $64X64$ pixel images, each image is a point in 4096-dimensional space.

Each image can be described by only two variables, the up-down and left-right pose.

Why reduce dimensionality?

More examples

- Handwritten digit images, text document "bag of words" representation.
  (Data is sparse, some features are always 0)
- Human body motion capture - 100 markers attached to body, each marker measures position in 3 dimensions, 300 dimensional feature space.
  (Motion constrained by joints and angles in the human body)

What is a manifold?

Definition

(Manifold)
A d-dimensional manifold is a subset $M \subset \mathbb{R}^D$ such that for each $x \in M$, there is an open neighbourhood around x, $N(x)$ and a diffeomorphism $f : N(x) \to \mathbb{R}^d$.

- A line is a 1-dimensional manifold.
- A d-dimensional affine subspace is a d-dimensional manifold.
- An arbitrary curve is a 1-dimensional *non-linear* manifold.

The manifold assumption

- Assume that the data lies on a d-dimensional manifold, *isometrically* embedded in $\mathbb{R}^D$.
- In other words, the data comes from a probability distribution, whose support lies on or close to a low-dimensional manifold.
- The learner receives only a few samples from this distribution.

What is Manifold Learning?

Manifold Learning is the study of algorithms that infer properties of data sampled from a manifold.

Generally interested in two things:

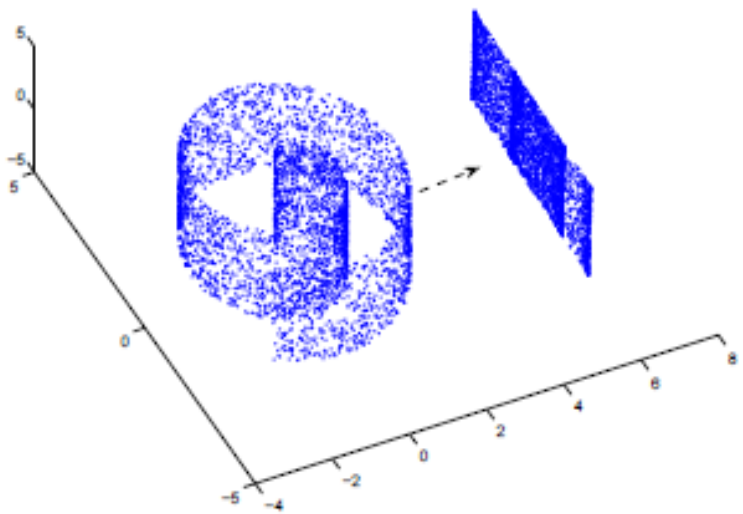
- Learning an explicit low dimensional representation of the data for visualization or as a preprocessing step for other algorithms.
- Exploiting this intrinsic low-dimensionality in data to speed up tasks like clustering and classification which suffer from a curse of dimensionality.

Principal Components Analysis

- Approximates the data by a low-dimensional affine subspace, such that the data projected onto the subspace has maximum variance.
- Algorithm:
  1. Compute sample covariance matrix S of centered data
  2. Compute eigenvectors of S corresponding to the d largest eigenvalues.
  3. Project data points onto the linear subspace spanned by these eigenvectors.
- If the dataset lies exactly on a d-dimensional subspace, only d eigenvalues will be non-zero.

Principal Components Analysis

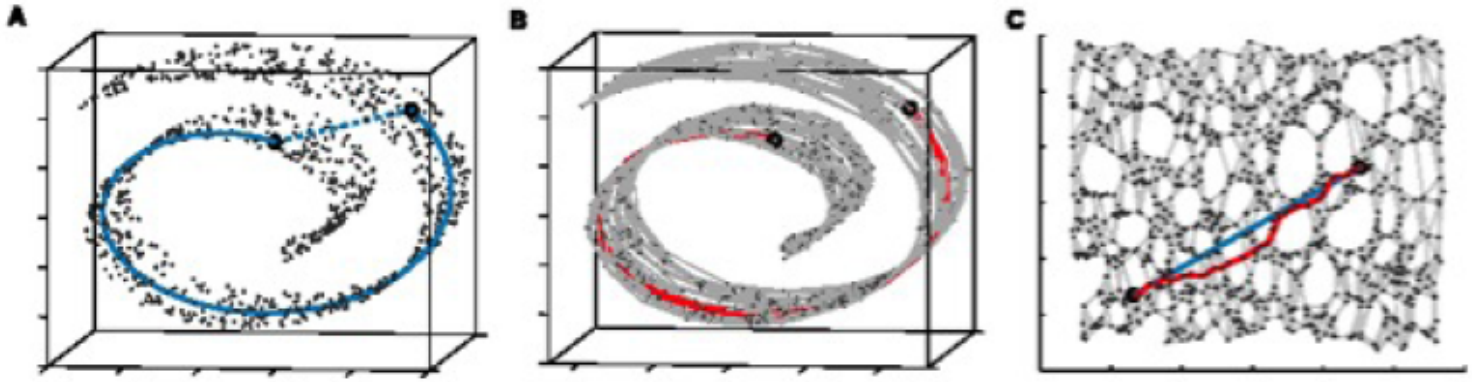
- Many machine learning tasks are based on inter-point distances or dissimilarities - clustering, quantization etc.
- Thus it is desirable that the mapping preserves inter-point distances.
- Easy to construct instances where PCA is unable to preserve distances -



Isomap : A non-linear approach

- Instead of preserving interpoint Euclidean distances, preserve *Geodesic* distances.
  Given points $x_1, x_2 ... x_n$ from a d-dimensional manifold, find points $y_1, y_2 ... y_n$ in $\mathbb{R}^d$ s.t. $||y_i - y_j|| = \rho_M(x_i, x_j)$
- Estimate geodesic distances between sampled points by a "Nearest-Neighbours Graph".
- Use metric multidimensional scaling to find a set of points in Euclidean space that closely approximate the interpoint geodesic distances.

Isomap : A non-linear approach



Algorithm:

1. Construct graph G with each sample point connected to its k nearest neighbours with edge weight $=$ euclidean distance.
2. Calculate shortest path distance between each pair of points.
3. Construct geodesic distance matrix D with d_{ij} equal to the shortest path distance between nodes i and j in G.
4. Embed D using MDS.

Questions

- How to choose k - the number of nearest neighbours ?
- How to efficiently extend to out-of-sample points?
- Need a notion of intrinsic dimensionality that generalizes manifold dimension.
- When do manifold learning algorithms find good embeddings close to the intrinsic dimension?

  In other words, when is it possible to find an embedding $f : \mathbb{R}^D \to \mathbb{R}^d$ s.t. $||f(x) - f(y)||$ closely approximates the ambient or geodesic distances $\rho(x, y)$ for all $x, y \in M$?

Characterizing the manifold model

Question: Under what data model should the algorithms be analyzed?
Possible Answer: The bounded curvature condition

- Suppose M is a d-dimensional submanifold of $\mathbb{R}^D$
- Medial axis: set of points in $\mathbb{R}^D$ with more than one nearest neighbour in M
- ber: Find the largest $\tau > 0$ s.t. every point on the manifold has distance $\geq \tau$ to the medial axis. Then, condition number $= 1/\tau$.

The condition number is useful because it upper-bounds the maximum curvature of the manifold, among other things.

Intrinsic dimension

Question: Can we find a broad notion of intrinsic dimensionality that generalizes manifold dimension, is empirically verifiable and facilitates analysis of algorithms?

1. Covering dimension: a set $S \subset \mathbb{R}^D$ has covering dimension d if there is a constant c s.t. for any ϵ, S has an ϵ-cover of size $c(1/\epsilon)^d$.
2. Doubling dimension: a set $S \subset \mathbb{R}^D$ has doubling dimension d if for every ball B, $S \cap B$ can be covered by 2^d balls of half the radius.

Random Projections...

... for an arbitrary finite point set.

- Given a set of points in $\mathbb{R}^D$, project them onto a *random subspace* of dimension d

Theorem

(Johnson-Lindenstrauss Flattening Lemma)
For any $0 < \epsilon < 1$ *and set of* n *points* U *in* $\mathbb{R}^D$, *let* d *be a positive integer s.t.* $d = \Omega(\frac{\log n}{\epsilon^2})$. *Then there is a linear map* $\Phi : \mathbb{R}^D \to \mathbb{R}^d$ *s.t. for all* $x, y \in U$:

$$(1-\epsilon) \leq \frac{||\Phi x - \Phi y||^2}{||x-y||^2} \leq (1+\epsilon)$$

- A projection onto a random subspace satisfies this with high probability.

Random Projections...

... for a restricted finite point set

- The JL-Lemma applies to arbitrary point sets. What if the set has low intrinsic dimension - for example, if the set is confied to a low dimensional affine subspace, or to a low dimensional manifold ?
- According to the JL-Lemma, as number of sampled points n increases, embedding dimension increases as $\log n$. This seems counter-intuitive, can we do better?

  Yes ! The JL Lemma can be used to show that random projections preserve entire subspaces. This can be extended to show that they preserve manifolds.

Random Projections...

... preserve subspaces

Theorem

(Subspace Preservation Lemma)
Given an n-dimensional affine subspace of V of $\mathbb{R}^D$*, and 0* $\epsilon, \delta < 1$*, let d be a positive integer s.t.* $d = \Omega(\frac{n}{\epsilon^2}\log\frac{1}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\delta})$*. If* Φ *is a random projection to d dimensions, then with probability* $> 1 - \delta$*, for every* $x \in V$*, the following holds :*

$$(1-\epsilon)\sqrt{d/D} \leq \frac{||\Phi x||^2}{||x||^2} \leq (1+\epsilon)\sqrt{d/D}$$

Random Projections of Manifolds

Basic outline :

- At any point $x \in M$, the tangent space T_x is an affine subspace. In a small enough neighbourhood around x, distances between points on the manifold can be approximated by distances between their projections on the tangent space.
- Since the tangent space is preserved under a random projection, inter-point distances in small neighbourhoods are also preserved.
- By taking an ϵ-net of suitable resolution on the manifold, faraway points are also preserved.

Proof relies on bounded curvature (finite condition number)

Random Projections of Manifolds

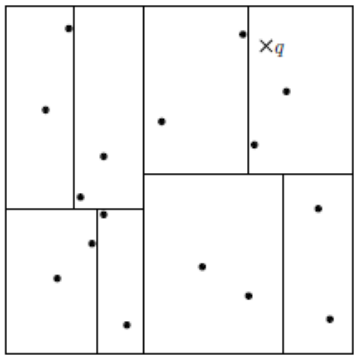
Theorem

(Manifold Preservation)
Given an n-dimensional manifold M in $\mathbb{R}^D$ with condition number $1/\tau$, suppose that for all $\epsilon > 0$, M has an ϵ-cover of size $\leq N_0(\frac{1}{\epsilon})^n$. Pick any $0 < \epsilon, \delta < 1$ and let d be a positive integer s.t. $d = \Omega(\frac{n}{\epsilon^2}\log\frac{D}{\epsilon\tau} + \frac{1}{\epsilon^2}\log\frac{N_0}{\delta})$. If Φ is a random projection to d dimensions, then with probability $> 1-\delta$, for every $x, y \in M$, the following holds :

$$(1-\epsilon)\sqrt{d/D} \leq \frac{||\Phi x||^2}{||x||^2} \leq (1+\epsilon)\sqrt{d/D}$$

Spatial data structures - kd trees

- Recursively partitions $\mathbb{R}^D$ into hyperrectangular cells.
- Used widely in machine learning & statistics.



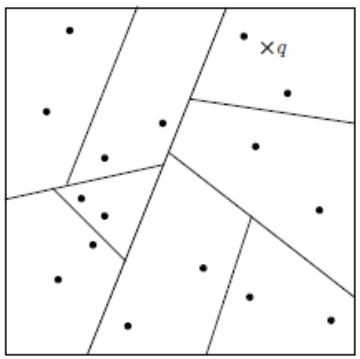
- Effectiveness depends on the rate at which the diameter of individual cells decreases down the tree.
- kd-trees can take D levels to reduce the diameter to half, this requires 2^D points.

Random-Projection trees

What if the data has low intrinsic dimension $d << D$? Do kd trees adapt to intrinsic dimensionality ? NO !

Enter Random-Projection trees:

- Instead of splitting along coordinate directions, split along a random direction in S^{D-1}. Instead of splitting along the median, add some random 'jitter'
- Claim: The random-projection tree adapts to doubling dimension



Random-Projection trees

RP-Tree Split Rule:
- choose a random unit direction $v \in \mathbb{R}^D$
- pick any $x \in S$, let $y \in S$ be the farthest point from it - choose δ uniformly randomly in $[-1, 1].6||x - y||/\sqrt{D}$
- rule(p):= $p.v \leq median(\{q.v : q \in S\} + \delta)$

Theorem

(RP-tree adapts to doubling dimension)
Suppose an RP-tree is built using data set $S \subset \mathbb{R}^D$. Pick any cell C in the RP-tree, suppose that $S \cap C$ has doubling dimension d. Then $\exists c_1$ s.t. with probability $\geq \frac{1}{2}$, for every descendant C' which is more than $c_1 d \log d$ levels below C, we have: radius$(C') \leq$ radius$(C)/2$

Random-Projection trees

Sketch of proof:

- Suppose $S \cap C$ lies in a ball of radius Δ. Cover $S \cap C$ with balls $B_1, B_2 \ldots B_N$ of radius $\Delta/\sqrt{d}$
- Since the doubling dimension of S is d, $n = O(d^{d/2})$ suffices.
- If two balls B_i and B_j are more than $(\Delta/2) - (\Delta/\sqrt{d})$ apart, then a single split has a constant probability of seperating them
- Since there n^2 such pairs (i, j), after $\Theta(d \log d)$ splits, each pair will have been split.
- Thus, after $\Theta(d \log d)$ levels, each cell will contain points from balls that are within distance $(\Delta/2) - (\Delta/\sqrt{d})$

References

- Random projection trees and low dimensional manifolds
  Dasgupta & Freund (STOC 2008)
- Random projections of smooth manifolds
  Baraniuk & Wakin (FoCM 2009)
- Mathematical Advances in Manifold Learning
  Nakul Verma (UCSD TR 2008)