Out Of Sample Extensions

Mauricio Gonzalez Soto

9 de febrero de 2016

Introduction

- Many unsupervised learning algorithms based on eigendecompositions provide an embedding or a clustering only for given training points.
- How do we deal with out-of-sample examples without recomputing eigenvectors?

Basic idea.

- We will provide a unified framework based on seeing the algorithms as learning eigenfunctions of a data-dependent kernel.
- With this framework, we will be able to extend:
  - Multidimensional Scaling
  - Local Linear Embedding.
  - Isomap.
  - Laplacian Eigenmaps.
  - Spectral Clustering.

The Algorithms

Quick overview of the five algorithms

Multidimensional Scaling

- We start with observations $x_1, ..., x_n \in \mathbb{R}^p$.
- Let d_{ij} distances between observations i, j.
- We seek $z_1, ...z_n \in \mathbb{R}^k$ such that

$$S_M(z_1, ..., z_n) = \sum_{i \neq j} (d_{ij} - ||z_i - z_j||)^2$$

- We are finding a lower-dimensional representation of the data that preserves pairwise distances as well as possible.

Non-linear dimension reduction

- Several methods have been recently proposed for non-linear dimension reduction.
- The idea is that the data lie close to an intrinsically low-dimensional non-linear manifold embedded in a high-dimensional space.
- We are somehow flattening the manifold.
- Two algorithms:
  - Isomap.
  - LLe.

Isomap

- Generalizes MDS to non-linear manifolds. It is based on replacing Euclidean distance by an approximation of the geodesic distance on the manifold.
- Constructs a graph to approximate the geodesic distance between points along the manifold.
- Specifically, for each data point we find its neighbors, then we construct a graph with an edge between any two neighbors.
- We approximate the geodesic distance by the shortest path in the graph.

LLE

- Looks for and embedding that preserves local geometry in the neighborhood on each point.
- Each data point is approximated by a linear combination of neighboring points.

Spectral Clustering

- Traditional clustering methods like K-means use a spherical or elliptical metric to group. They do not work well when clusters are non-convex.
- The starting point is an $n \times n$ matrix of pairwise similarities between all observations.
- We represent the observations in an undirected graph.
- The vertices represent the observations and the edges the similarities.
- Clustering is now a graph-partition problem, where we identify connected components with clusters.
- We wish to partition the graph such that edges between groups have low weights and edges within groups high weights.
- We use as similarity matrix the radial-kernel gram matrix.

Common Framework

- The five algorithms that we are considering can be casted in the same framework; they're all based on the computing an embedding for the training points.
- This embedding is obtained from the eigenvectors of a symmetric matrix

Generic Algorithm

- We start from a data set $D = \{x_1, ..., x_n\}$.
- Construct an $n \times n$ similarity Matrix M, and let $K_D(\cdot, \cdot)$ the data-dependent function which produces M; i.e. $M_{ij} = K_D(x_i, x_j)$.
- (Optionally) Transform M to a (somehow) "normalized" $\tilde{M}$. Obviously, $\tilde{K}_D$ is what you are thinking it is.
- Compute the m largest eigenvalues λ_k and eigenvectors v_k of $\tilde{M}$.
- The embedding for each example x_i is the vector y_i with y_{ik} is the i-th element of the k-th principal eigenvector v_k of $\tilde{M}$.
- For MDS and Isomap, the embedding is given by e_i where

$$e_{ik} = \sqrt{\lambda_k} y_{ik}$$

If the first m eigenvalues are positive, then $e_i \cdot e_j$ is the best approximation of $\tilde{M}_{ij}$ in the squared-error sense using only m coordinates.

Particular Cases

- In the following, we will consider the particular cases of the Generic Algorithm for the previously mentioned learning algorithms.
- Let S_i be the i-th row sum of the matrix M:

$$S_i = \sum_j M_{ij}$$

- We say that two points a, b are $k - nearest - neighboors$ of each other if a is among the k nearest neighboors of b in $D \cup \{a\}$ or vice-versa.

Once again, MDS

- We take

$$\tilde{M}_{ij} = -\frac{1}{2}\left(M_{ij} - \frac{1}{n}S_i - \frac{1}{n}S_j + \frac{1}{n^2}\sum_k S_k\right)$$

- The embedding is given by

$$e_{ik} = \sqrt{\lambda_k} v_{ki}$$

Spectral Clustering.

- The affinity Matrix is formed using a kernel such as the Gaussian.
- Several normalizations, the most succesful is:
- Take

$$\tilde{M}_{ij} = \frac{M_{ij}}{\sqrt{S_i S_j}}$$

- To obtain m clusters, the first m principal eigenvectors of $\tilde{M}$ are computed and $y_{ik} = v_{ji}$ ## Isomap

LLE

- First, a sparse matrix of local predictive weights W_{ij} is computed such that

$$(\sum_j W_{ij} x_j - x_i)^2$$

  is minimized.
- Then, the matrix

$$M = (I - W)'(I - W)$$

  is formed.

Laplacian Eigenmaps

- Solving generalized eigenproblem

$$(S - M)v_j = \lambda_j S v_j$$

- I really hope my classmates explain this one.

From eigenvectors to eigenfunctions.

- We start from data D, obtain and embedding, and add more data.
- The embedding for the points in D will converge.
- Each eigenvector converges to an eigenfunction.
- Wtf?
- It converges in the sense that the i-th element of the k-th eigenvector converges to the application of the k-th eigenfunction to x_i.
- Still. . . wtf?

Hoping this one make things clear.

Proposition 1 Let $\tilde{K}(a, b)$ be a Kernel function, not necessarily positive semi-definite that gives rise to a symmetric matrix $\tilde{M}$ with entries $\tilde{M}_{ij} = \tilde{K}(x_i, x_j)$ upon a dataset D. Let (v_k, λ_k) a pair that satisfies $\tilde{M}v_k = \lambda_k v_k$. Let (f_k, λ'_k) be a pair that satisfies

$$(\tilde{K}_p f_k)(x) = \lambda'_k f_k(x)$$

for any x and p the empirical distribution over D. Let $e_k(x) = y_k(x)\sqrt{\lambda_k}$ or $y_k(x)$ denote de embedding associated with a new point x. Then,

- $\lambda'_k = \frac{1}{n}\lambda_k$
- $f_k(x) = \frac{\sqrt{n}}{\lambda_k}\sum_i v_{ki}\tilde{K}(x, x_i)$
- $f_k(x_i) = \sqrt{n}v_{ki}$
- $y_k(x) = \frac{f_k(x)}{\sqrt{n}} = \frac{1}{\lambda_k}\sum_i v_{ki}\tilde{K}(x, x_i)$
- $y_k(x_i) = y_{ik}$
- $e_k(x_i) = e_{ik}$

Proposition 2 If the data-dependent kernel $\tilde{K}_D$ is positive semi-definite, then

$$f_k(x) = \sqrt{\frac{n}{\lambda_k}}\pi_k(x)$$

where $\pi_k(x)$ is the

Extending to new points.

- Using proposition 1, one obtains a natural extension of all unsupervised learning algorithms mapped to the Generic Algorithm, provided we can write a kernel function $\tilde{K}$ that gives rise to the matrix $\tilde{M}$.

Extending MDS

- Take

$$\tilde{K}(a,b) = -\frac{1}{2}(d^2(a,b) - \mathbb{E}[d^2(x,b)] - \mathbb{E}[d^2(a,x')] + \mathbb{E}[d^2(x,x')])$$

Extending Spectral Clustering

- Take,

$$\tilde{K}(a,b) = \frac{1}{n} \frac{K(a,b)}{\sqrt{\mathbb{E}[K(a,x)]\mathbb{E}[K(b,x')]}}$$

Extending Isomap

- We do not use the new point to compute geodesic distances.
- Apply double centering just as in MDS.
- A formula has been proposed,

$$e'_k(x) = 1/2\sqrt{\lambda_k}\sum_i v_{ki}(E[\tilde{D}^2(x', x_i)] - \tilde{D}^2(x_i, x))$$

Extendind LLE

- LLE is complicated because it doesn't fit as well the framework of the Generic Algorithm. The Matrix M doesn't have a clear interpretation as a distance matrix.
- Sauk and Roweis proposed a method, where the embedding of a new point is given by,

$$y_k(x) = \sum_i y_k(x) w(x, x_i)$$

where $w(x, x_i)$ is the weight of x_i in the reconstruction of x by its nearest k-neighbors in the training set D

References

- "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps and Spectral Clustering". Yoshua Bengio, Jean-Francois Paiement, Pascal Vincent, Olivier Dellaleu, Nicolas LeRoux and Marie Ouimet.
- "Geometría Riemanniana". Hector Sanchez Morgado, Óscar A. Palmas Velasco.