Analysis of High-dimensional Data Sets and Graphs

Mauro Maggioni

Mathematics, Computer Science, C.T.M.S.
Duke University

3/21/09
A little bit about myself

- Joined Duke Math 2.5 years ago
- Fields of expertise, within Applied Math/Analysis:
  - Harmonic Analysis (Fourier analysis, wavelets)
  - Signal Processing (analysis of sounds, images; algorithms for compression, denoising, classification of images) and approximation theory
  - Graph theory, in particular spectral graph theory, random walks, and applications to analysis of data sets, networks
  - Computational aspects of all of the above
- Fields of interest:
  - High-dimensional geometry and probability
  - Machine learning, Markov decision processes, statistical learning
  - Stochastic processes and probability in general
Structured data in high-dimensional spaces

A deluge of data...
Documents, web searching, customer databases, hyperspectral imagery (satellite, biomedical, etc...), social networks, gene arrays, proteomics data, neurobiological signals, sensor networks, financial transactions, traffic statistics (automobilistic, computer networks)...

...of higher and higher dimensionality
Instruments capture data with higher and higher “resolution” (e.g. more and more megapixels in a camera, more and more spectral bands in a mass- or Raman- spectrometer, more and more detailed user transaction data on web sites etc...). The number of samples (e.g. pictures, spectra, users being tracked etc...) also increases.
Many questions and problems

1. How to represent the data efficiently? How to store it?
2. How to search the data? How to organize it? How to extract information?
3. How to visualize high-dimensional data for human interaction?
4. How to use the data to make predictions? How to learn from data?

The high-dimensionality of the data may make many of these questions hard.
Many questions and problems

1. How to represent the data efficiently? How to store it?
2. How to search the data? How to organize it? How to extract information?
3. How to visualize high-dimensional data for human interaction?
4. How to use the data to make predictions? How to learn from data?

The high-dimensionality of the data may make many of these questions hard.
Many questions and problems

1. How to represent the data efficiently? How to store it?
2. How to search the data? How to organize it? How to extract information?
3. How to visualize high-dimensional data for human interaction?
4. How to use the data to make predictions? How to learn from data?

The high-dimensionality of the data may make many of these questions hard.
Many questions and problems

1. How to represent the data efficiently? How to store it?
2. How to search the data? How to organize it? How to extract information?
3. How to visualize high-dimensional data for human interaction?
4. How to use the data to make predictions? How to learn from data?

The high-dimensionality of the data may make many of these questions hard.
Many questions and problems

1. How to represent the data efficiently? How to store it?

2. How to search the data? How to organize it? How to extract information?

3. How to visualize high-dimensional data for human interaction?

4. How to use the data to make predictions? How to learn from data?

The high-dimensionality of the data may make many of these questions hard.
Machine learning problems

Framework for machine learning, or learning from examples. Probability distribution in $\mathbb{R}^d \times \mathbb{R}$, from which random samples $\{(x_i, y_i)\}_{i=1,...,n}$ are drawn. The $x_i$’s are the data points, and $y_i$’s are values of a function of interest. We think of

$$y_i = f(x_i) + \eta, \quad \eta \text{ a noise term}$$

Goal: return (learn) a function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ such that

$$\hat{f}(x) \sim f(x)$$

(e.g.: would like $\hat{f}(x) = \mathbb{E}[f(x)|x]$). $\hat{f}$ predicts $f$ for every data $x$. Typical gameplay: assume $f$ is regular in some sense, then estimate the relationship between $n$ and $||\hat{f} - f||$. 

Mauro Maggioni

Analysis of High-dimensional Data Sets and Graphs
The curse of dimensionality: dividing the unit cube in $d$ dimensions in $1/\epsilon$ pieces in each dimension, leads to $1/\epsilon^d$ cubes with side $1/\epsilon$ within the unit cube. Therefore to approximate a generic density of points in the unit cube, with precision $\epsilon$, we need $1/\epsilon^d$ points. E.g.: $\epsilon$ equal to a modest $1/10$, $d = 10$ (also modest), implies $10^{10}$ points needed!

The number of data points, while often growing, is not growing fast enough with the dimensionality of the data acquired.

But what IF the data, while measured in $\mathbb{R}^d$, actually lies on a lower-dimensional set?
The curse of dimensionality: dividing the unit cube in $d$ dimensions in $1/\epsilon$ pieces in each dimension, leads to $1/\epsilon^d$ cubes with side $1/\epsilon$ within the unit cube. Therefore to approximate a generic density of points in the unit cube, with precision $\epsilon$, we need $1/\epsilon^d$ points. E.g.: $\epsilon$ equal to a modest $1/10$, $d = 10$ (also modest), implies $10^{10}$ points needed!

The number of data points, while often growing, is not growing fast enough with the dimensionality of the data acquired.

But what **IF** the data, while measured in $\mathbb{R}^d$, actually lies on a lower-dimensional set?
Handwritten Digits

Data base of about 60,000 $28 \times 28$ gray-scale pictures of handwritten digits, collected by USPS. Point cloud in $\mathbb{R}^{28^2}$. Goal: automatic recognition.

Set of 10,000 picture (28 by 28 pixels) of 10 handwritten digits. Color represents the label (digit) of each point.
1000 Science News articles, from 8 different categories. We compute about 10000 coordinates, \( i \)-th coordinate of document \( d \) represents frequency in document \( d \) of the \( i \)-th word in a fixed dictionary.
An example from Molecular Dynamics, I

The dynamics of a small protein in a bath of water molecules is approximated by a Langevin system of stochastic equations

$$\dot{x} = -\nabla U(x) + \dot{w}.$$

The set of states of the protein is a noisy set of points in $\mathbb{R}^{36}$. 

---

Mauro Maggioni  
Analysis of High-dimensional Data Sets and Graphs
Application to Hyper-spectral Pathology

For each pixel in a hyper-spectral image we have a whole spectrum (e.g. a 128-dimensional vector). The ensemble of all spectra in a hyper-spectral image is a cloud in $\mathbb{R}^{128}$, induce a random walk on the point set and map the values of the top 3 eigenfunctions to RGB: they divide the spectra into different classes, which turn out to be biologically distinct and relevant. On the left, we have mapped the values of the top 3 eigenfunctions to RGB.
For each pixel in a hyper-spectral image we have a whole spectrum (e.g. a 128-dimensional vector). The ensemble of all spectra in a hyper-spectral image is a cloud in $\mathbb{R}^{128}$, induce a random walk on the point set and map the values of the top 3 eigenfunctions to RGB: they divide the spectra into different classes, which turn out to be biologically distinct and relevant. On the left, we have mapped the values of the top 3 eigenfunctions to RGB.
The samples are typically collected in “random” order. Can we order them? One way of interpreting this question is: find clusters, or parametrizations for these low-dimensional sets embedded in high-dimensional space.

**Study of the geometry of data sets**

Often one wants to make predictions on the sample: these are functions on the data sets. It is natural to use the geometry of the data to design schemes for approximating functions on them.

**Study of functions on data sets**
What can math do?

Design novel schemes for analysing, in a quantitative, numerically efficient, and principled fashion the geometry of data sets and functions on them.

Tools I use: generalized harmonic analysis (Fourier, multiscale analysis etc...); probability and Markov chains; approximation theory; geometric measure theory...
Random walks and heat kernels on the data

Assume the data \( X = \{ x_i \}_{i=1}^{N} \subset \mathbb{R}^d \). Typically \( d \) large (from 10 to \( 10^4 \)). Assume we can assign local similarities via a kernel function \( K(x_i, x_j) \geq 0 \).

Example: \( K_\sigma(x_i, x_j) = e^{-||x_i - x_j||^2/\sigma} \).

Model the data as a weighted graph \((G, E, W)\): vertices represent data points, edges connect \( x_i, x_j \) with weight \( W_{ij} := K(x_i, x_j) \), when positive. Let \( D_{ii} = \sum_j W_{ij} \) and

\[
\begin{align*}
P &= D^{-1} W, \\
T &= D^{-\frac{1}{2}} WD^{-\frac{1}{2}}, \\
H &= e^{-t(I-T)}
\end{align*}
\]

random walk \hspace{1cm} symm. “random walk” \hspace{1cm} Heat kernel

Note 1: \( K \) typically depends on the type of data.

Note 2: \( K \) should be “local”, i.e. close to 0 for points not sufficiently close.
Random walks and heat kernels on the data

Assume the data $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^d$. Typically $d$ large (from 10 to $10^4$). Assume we can assign local similarities via a kernel function $K(x_i, x_j) \geq 0$.

Example: $K_\sigma(x_i, x_j) = e^{-\|x_i - x_j\|^2/\sigma}$.

Model the data as a weighted graph $(G, E, W)$: vertices represent data points, edges connect $x_i, x_j$ with weight $W_{ij} := K(x_i, x_j)$, when positive. Let $D_{ii} = \sum_j W_{ij}$ and

$$P = D^{-1}W, \quad T = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \quad H = e^{-t(I-T)}$$

random walk symm. “random walk” Heat kernel

Note 1: $K$ typically depends on the type of data.

Note 2: $K$ should be “local”, i.e. close to 0 for points not sufficiently close.
Random walks and heat kernels on the data

Assume the data $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^d$. Typically $d$ large (from 10 to $10^4$). Assume we can assign local similarities via a kernel function $K(x_i, x_j) \geq 0$.

Example: $K_\sigma(x_i, x_j) = e^{-||x_i-x_j||^2/\sigma}$.

Model the data as a weighted graph $(G, E, W)$: vertices represent data points, edges connect $x_i, x_j$ with weight $W_{ij} := K(x_i, x_j)$, when positive. Let $D_{ii} = \sum_j W_{ij}$ and

\[
P = D^{-1}W, \quad T = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \quad H = e^{-t(I-T)}
\]

random walk symm. "random walk" Heat kernel

Note 1: $K$ typically depends on the type of data.

Note 2: $K$ should be “local”, i.e. close to 0 for points not sufficiently close.
Does the above sound strange?

Ever wondered how Google sorts the results of a search? It constructs a random walk on the set of (all!) Web pages ($8 \cdot 10^9$ of them...), let’s call that $P$, and computes the eigenvector corresponding to the largest eigenvalue (which is in fact 1): $\phi P = \phi$. One can show that $\phi(x) \geq 0$, and PageRank sorts the results, i.e. web-pages $x$, in decreasing order of the value of $\phi(x)$. 
Connections with harmonic analysis, probability....

\[ N = \#\text{pts} \to +\infty: \text{these operators tend to natural operators on manifolds (e.g. Laplace-Beltrami, heat kernel), if the data is sampled from a manifold.} \]

Connections with:

1. Laplacian on manifolds, heat kernels;
2. Random walks on graphs
3. Eigenfunctions of the Laplacian, Fourier analysis, wavelet analysis, signal processing
4. Function approximation
5. High-dimensional geometry, with stochastic effects (high-dimensional noise)
6. ...
One can try to use the operator $T$ as above, which is intrinsic to
the data, to construct parametrizations of the data.
Popular choice of embedding/parametrization $\mathcal{M} \rightarrow \mathbb{R}^m$ is:

$$x \mapsto (\varphi_1(x), \ldots, \varphi_m(x)),$$

where $\varphi_i$ is the $i$-th eigenfunction of the Laplacian $L$.
Used in many fields for decades, almost nothing known about
the properties of this map. Used in all the examples above.
Recently [jointly with P.W. Jones and R.Schul] we showed that
a wise choice of eigenfunctions leads to optimal bi-Lipschitz
embeddings of large portions of $\mathcal{M}$. 
Some recent/ongoing projects

1. Estimate intrinsic dimensionality of data sets with noise (with A. Little and Y.-M. Jung - Math)
2. Multiscale geometric analysis of data sets (with A. Little and Y.-M. Jung - Math)
3. Analysis of multiscale graphs and their dynamics (with K. Balachandran - Math)
4. Visualizing and navigating graphs in a multiscale fashion (with E. Monson and R. Brady - CS)
5. Statistical analysis on manifolds, with applications to analysis of gene microarrays (with J. Guinney and S. Mukherjee - Stat & Comp.Bio.)
6. Signal processing on manifolds, with generalized Fourier and wavelet analysis (with R.R. Coifman)
7. Parametrizations of manifolds with heat kernels and eigenfunctions (with P.W. Jones and R. Schul)
8. Analysis of social networks (with R. Rajae (EE) and W. Willinger (AT&T))
Thanks!

I’ll be happy to talk with any you later today - and/or answer questions by e-mail after today.

For more info/presentations/tutorials you can check out my web page at

www.math.duke.edu/~mauro

Enjoy your visit at Duke!
Example: Multiscale text document organization

Multiscale directory structure

Scaling functions at different scales represented on the set embedded in $\mathbb{R}^3$ via $(\xi_3(x), \xi_4(x), \xi_5(x))$. $\phi_{3,4}$ is about Mathematics, but in particular applications to networks, encryption and number theory; $\phi_{3,10}$ is about Astronomy, but in particular papers in X-ray cosmology, black holes, galaxies; $\phi_{3,15}$ is about Earth Sciences, but in particular earthquakes; $\phi_{3,5}$ is about Biology and Anthropology, but in particular about dinosaurs; $\phi_{3,2}$ is about Science and talent awards, inventions and science competitions.
## Doc/Word multiscales

<table>
<thead>
<tr>
<th>Scaling Fcn</th>
<th>Document Titles</th>
<th>Words</th>
</tr>
</thead>
</table>
| $\varphi_{2,3}$ | Acid rain and agricultural pollution  
Nitrogen's Increasing Impact in agriculture | nitrogen, plant, ecologist, carbon, global |
| $\varphi_{3,3}$ | Racing the Waves Seismologists catch quakes  
Tsunami! At Lake Tahoe?  
How a middling quake made a giant tsunami  
Waves of Death  
Seabed slide blamed for deadly tsunami  
Earthquakes: The deadly side of geometry | earthquake, wave, fault, quake, tsunami |
| $\varphi_{3,5}$ | Hunting Prehistoric Hurricanes  
Extreme weather: Massive hurricanes  
Clearing the Air About Turbulence  
New map defines nation's twister risk  
Southern twisters  
Oklahoma Tornado Sets Wind Record | tornado, storm, wind, tornadoe, speed |

Some example of scaling functions on the documents, with some of the documents in their support, and some of the words most frequent in the documents.
Given:

- $X$: all the data points
- $(\tilde{X}, \{\chi_i(x)\}_{x \in \tilde{X}, i=1,\ldots,I})$: a *small* subset of $X$, with labels: $\chi_i(x) = 1$ if $x$ is in class $i$, 0 otherwise.

Objective:

- guess $\chi_i(x)$ for $x \in X \setminus \tilde{X}$.

Motivation:

- data can be cheaply acquired ($X$ large), but it is expensive to label ($\tilde{X}$ small). If data has useful geometry, then it is a good idea to use $X$ to learn the geometry, and then perform regression by using dictionaries on the data, adapted to its geometry.
Algorithm:

- use the geometry of $X$ to design a smoothing kernel (e.g. heat kernel), and apply such smoothing to the $\chi_i$'s, to obtain $\tilde{\chi}_i$, soft class assignments on all of $X$. This is already pretty good.

- The key to success is to repeat: incorporate the $\tilde{\chi}_i$'s into the geometry graph, and design a new smoothing kernel $\tilde{K}$ that takes into account the new geometry. Use $\tilde{K}$ to smooth the initial label, to obtain final classification.

Experiments on standard data sets show this technique is state-of-the-art.