Multi-Domain Data Analysis

Jason M. Kinser, D.Sc.

Bioinformatics and Comp. Biol., GMU
&
Computational Sciences,
Lawrence Berkeley National Lab
Sept., 2008

Introduction

Real world problems create multiple types of data:

Biology:
- DNA
- Proteins
- Pathways
- Expression Arrays
- Microscopic images

Forensics:
- DNA
- Fingerprints
- Shoe prints
- Faces
- Voice

Text Mining:
- Text
- Authors
- Citations
- Images
- Graphs
- Keywords
**Justification**

WHY?

Why do you want to sit here for the next hour?

Should I be emailing a friend instead?

**Real World Events**

Events in the real world creating multiple domains of data:

A publication can be considered as an event.

We can certainly compare two documents based on their word frequencies.

Should this comparison also be sensitive to the authors? (Is the word frequency due to similarity in topics – or word preferences of authors?)

Should searches in one domain be sensitive to information provided by other domains?
Common Approach

Common approaches uses a single database for each domain.

When you perform a BLAST search do you get to use micro-array information? NO, you just use the sequence.

1. Fuse First
2. Fuse Last

Neither is a terrific idea – BECAUSE ... (insert cliff hanger here)

Definitions

INTRA-DOMAIN comparisons. The engine that compares two pieces of data from the same domain. Examples: Dynamic programming alignment, BLAST, fingerprint search.

INTER-DOMAIN comparisons. The engine that compares two pieces of data from different domains. It could be a very simple Boolean term.

NOTE: In some applications it may desired to have multiple intra-domain comparison engines for a single domain.
Problems with Current Approach

We wouldn’t be here if there wasn’t problems with the current approach.

• It does not use inter-domain comparison engines.
• A fuse-afterwards approach (sort of) uses inter-domain comparison engines but they are decouple from intra-domain engines.
• The intra-domain engines are decoupled.

Solution to the Problem

Create a single data space that contains data from all domains.

Distances in this new space are linearly proportional to similarity measures in the input space.

That means:
- The distance between two points is proportional to the intra-domain engine.
- The distance between two points is proportional to the inter-domain engine.

Must be searchable.
Professional Systems

Professional Systems
Generally use one database for each domain.

AFIS is the fingerprint database system and it searches for fingerprints using fingerprint information.

Representing Data

We can represent data on a theoretic graph $G(V,E)$.

The nodes represent a data sample. The edges represent a measure of similarity. (For now we can only one edge – but that is just a temporary setback.)
Problems with Graph

What should be the distance between the two marked points?

We have to follow along the edges within the graph.
The shortest distance along the edges between two points is called the geodesic distance.

What is the geodesic distance between all possible pairs of data points?

Problem #2

Consider the problem of using the G(V,E) as an associative memory.

The new query data point needs to be located somewhere.
It probably won’t be on an edge or node.

Do we need to connect it to the graph? How? Where should it be located??

Why do we have more questions than answers?
Solution

The solution is to use a *distance preserving* transform such that:

- The distances along the edges are converted to Euclidean distances via a linear transform.
- All of the new space is defined as a Euclidean space.

Swiss Roll

Recent publications in SCIENCE demonstrate this idea. Geodesic distances become linear distances using MDS.

---

There were two consecutive papers in SCIENCE – one from Tenenbaum et al. and one from Roweis et al.
Why Swiss Roll

K-means stinks on this problem.

MDS

1. Create G(V,E) – distance map.
2. Compute geodesic distances
3. Center
4. Compute Eigen-stuff
5. Map Elements of the eigenvectors (scaled according to the eigenvalues)
Measuring the Performance

499,500 possible pairs of data points. Graph the distance in Swiss space versus distance in output space.

Perfect Linearity

If we could achieve perfect linearity then the blue and red lines would be the same.
Multiple Comparison Engines

Multiple intra-domain comparison engines for a single domain. Multiple edges between two nodes.

A little binf

In a previous study [5, Kanaya M, Kimouchi, T, Abi, Y, Kudo, Y, Yamada, T, Nishi, H, Mori, T, Ikemura, 2001] 29 different bacterial species were used that produced more than 59,000 genes each with at least 100 codons. The study showed that species had signature codon frequencies
The World is not Well Behaved

Comparing Codon Freq Vectors

However, when applied to a real world problem the linearity begins to fade a little.

A little? It was drawn up and quartered.

New Research

Thanks be given to Dr. James Sethian and Dr. Daniela Ushizima (LBNL).

http://vis.lbl.gov/~daniela/

http://math.berkeley.edu/~sethian/

http://en.wikipedia.org/wiki/James_Sethian
Breast Data

Data not available for release.

Results
GO Ontology

Different Method
Problems with this method

Increase the Dimensionality

It is possible to improve the linearity by increasing the dimensionality of the output space. HOWEVER, there is a limit to what this accomplishes.

The new space in 3D. It doesn’t lose the 2D shape – it just bends it.
Bridge

Slight Improvement in Linearity

OLD

NEW
Increase in Dimensionality

An increase in dimensionality = lowering the complexity of the data. Is there a limit to the number of dimensions? Will it haunt us later?

An increase in dimensionality will NOT cause problems in the applications.

More Justification (wake up)

Is this just another happy algorithm ... or can we really use it for something?

Can we do something that no one has ever done before??
The condition of linearity is critical.

If the condition of linearity holds then...
ALL Euclidean distances in the output space are linearly proportional to intra- and inter-domain comparison engines.
All space is defined as a weighted linear combination of these comparison engines.

This is a multi-domain clustering system.
Recall that the points in this space represent different domains of data.
An associative memory receives an input query and recalls similar items.

But you say, “We already have associative memories (neural networks).”

This is a multi-domain associative memory. The query can be one data type and the recall can be other data types.

The query is placed in the output space by using a few intra-domain comparisons. Once it is located then we make use of: the Euclidean distance is proportional to all comparison engines.

This task is not impaired by a large dimensionality in the output space.

Time based events leave a predictable trace through the output space. We can predict the next point in time (Kalman filtering).
Multi-Domain Prediction

These traces go through a multi-domain space.

Text Mining