Utilizing External Information in Social Networks

David J. Marchette

Naval Surface Warfare Center
david.marchette@navy.mil

August 4, 2009

Graphs vs Networks

- A graph is a pair (V, E) of vertices and edges.
  - V is a set (called vertices or actors).
  - E is a set of pairs of vertices (unordered, if a graph, ordered if a directed graph).
  - We'll assume the graph is simple: no self loops or multiple edges.
- A network is a graph with extra information:
  - Weights or covariates on the:
    - Edges.
    - Vertices.
    - There may be external information about the graph as a whole.
  - The usual definition of a network is a directed graph with weighted edges, but we will broaden this definition.

Relationships Matter

- Many (most?) decision making is performed in an environment in which relationships among the actors are critical.
- Encoding the relationships in a way that allows proper decision making is a challenge.
- Statisticians need to be able to:
  - Process (perform inference on) these types of relationship data.
  - Combine these data with other data relevant to the inference.
  - Report the results to decision makers in ways that they can be understood.
- Policy decisions that do not take into account what is known are likely to be misguided.

Examples

- Typical problem domains:
  - Nation states: the actions or stability of a nation depends not just on their internal state (covariates) but also on their relationships in the world community.
  - Computer networks: the security of a computer depends not only on its software suite, but also on what services it provides and to whom, as well as what services it requests, and from whom.
  - Criminal organizations: the extent of an organization's activity is a function of both the people in the organization and their interaction with other organizations and with society.
  - Sensor networks: a network is more efficient if processing takes network topology into account.

What Decision Makers Need

- Decisions must be made in complex situations in which both traditional information (measurements, covariates, sensor output) must be combined with relationship and other complex information.
- Thus, we need to have ways to fuse as much information together as possible, for problems where the types of data relevant to the problem are very diverse, and good models are often hard to come by.
- There is a premium on methods that are easily translated for the non-specialist.

Framework: Setup

- Here is a basic framework that is general enough to discuss the various issues.
- Given $D_t = (G_t, H_t, Z_t)$ where
  - $G_t = (V, E_t)$ is an unweighted (simple) graph, $|V| = n$.
  - $H_t = (V, W_t)$ is a weighted graph with (edge) weights W_t.
  - Z_t are (vertex) covariates, Z_t is an $n \times q$ matrix.
- Note that we assume the vertex sets are the same (and static) in both graphs, and the graphs are labeled.
- Various extensions are possible, but this is enough to illustrate various ideas.

Framework: Embedding

- We seek to embed D_t in some space in which we can perform inference.
- $h : \mathcal{D} \rightarrow \mathbb{R}^{n \times d}$. This is an embedding of the n vertices.
- We can embed each D_t independently, or we can utilize the time series nature of the data in the embedding. We will primarily look at ways to embed each observation separately.
- Ideally, we want the best such embedding for a given inference task, measured by β:

$$h^* = \arg\max_h \beta(h; \mathcal{D})$$

Framework: Embedding

- We can embed each piece of D_t separately:

$$\begin{aligned} G_t &\mapsto X_{G_t} \in \mathbb{R}^{n \times d_G} \\ H_t &\mapsto X_{H_t} \in \mathbb{R}^{n \times d_H} \\ Z_t &\mapsto X_{Z_t} \in \mathbb{R}^{n \times d_Z} \end{aligned}$$

- Then combine the embeddings:

$$(X_{G_t}, X_{H_t}, X_{Z_t}) \subset \mathbb{R}^{d_G + d_H + d_Z}.$$

- Note: We will be primarily interested in embedding the vertices of the graphs, rather than the graph as a whole, but the basic ideas are the same in either case.

Framework: Embedding

- Or we can embed the totality:

$$(G_t, H_t, Z_t) \mapsto \mathbb{R}^{n \times d}.$$

- In principle, this approach should be superior to separate embedding, in situations in which there is correlation, or similar information, in the different graphs and/or covariates.
- We seek methods which can be applied to the pieces separately or the combination.

Inference on Graphs

- Inference can take (at least) two distinct forms:
  - Inference about the entities in this particular graph.
  - Inference about the process that generated the graph.
- In either case, the graph may be modeled as random, there may be random covariates or edge weights, there may be a time series of graphs, etc.

Typical Inferences

- A vertex or group of vertices is “acting differently” than the others or than it has in the past.
- The graph was generated by this process with these parameters (or not).
- The random graph process has changed.
- Certain vertices (or edges) are “important” (central, bridges, sources/sinks, choke-points).
- Certain structure is/is not present in the graph.
- Information flows most efficiently along these paths, through these vertices.
- The graph does/does not predict the covariates (or vice-versa).

Three Views of Graph Inference

- There are three basic approaches to inference on graphs:
  - Feature extraction: compute invariants of the graph (features).
  - Model: posit a model of the random graph (process).
  - Project: embed the graph in a space.
- In some sense, the first two could be thought of as special cases of the third.
- All of these can be performed on the graph as a whole, or on the nodes in the graph.
- The idea is to move from “graph space” to something familiar where we can perform inference.

Time Series of Graphs

- A time series of graphs is a collection of graphs indexed by time. We will assume the graphs are labeled: each vertex has a unique identifier that is consistent across all graphs containing it.
- We will consider four methods for analyzing a time series of graphs:
  1. Extract an invariant, and model the time series of the invariant.
  2. Compute a change in the edges structure (Frobenius or related norm).
  3. Embed via a spectral method, and track the embedded points.
  4. Embed via a random embedding method, and track the embedded points.
- There are also various model based methods to consider.

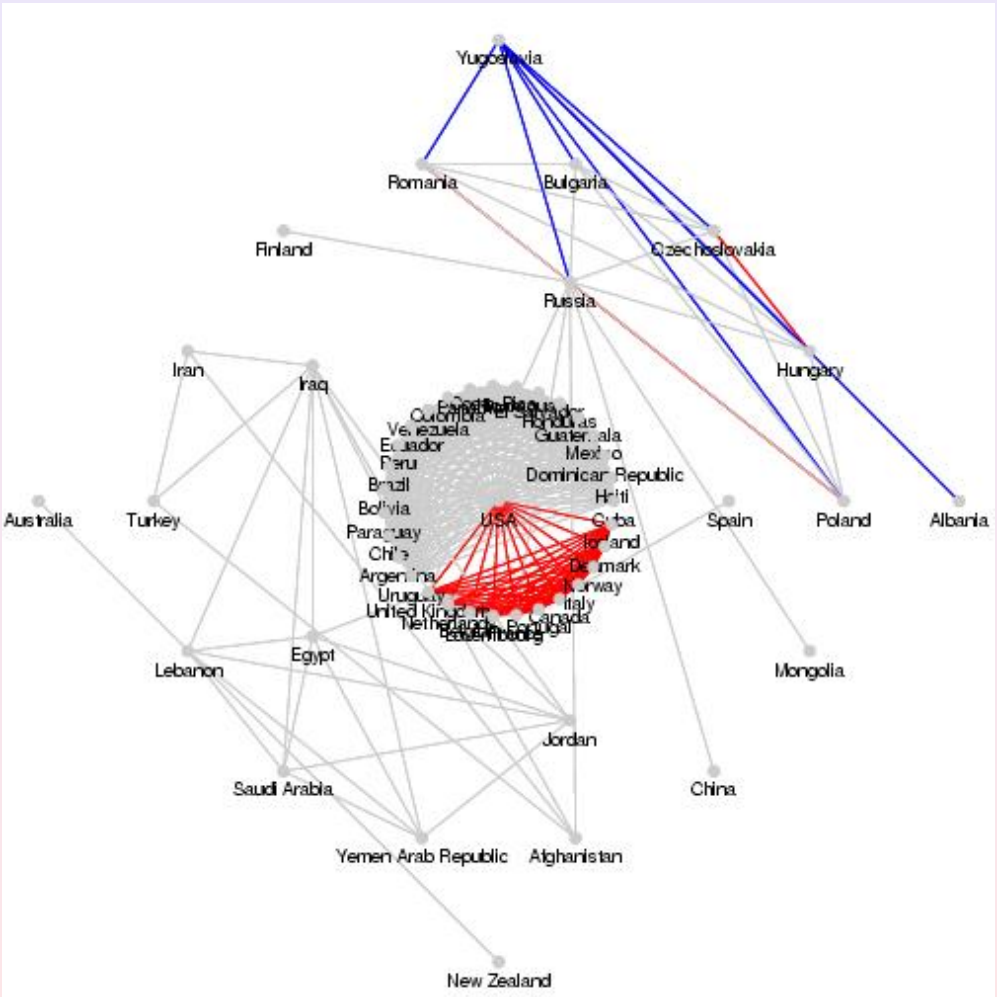
Graph Features

- Several standard approaches to graph invariants:
  - Centrality measures: how “central” are nodes in the graph.
  - Cliques, clubs, blockmodels, other measures of clustering of nodes.
  - Scan statistics.

An Invariant: Scan Statistics

- A locality statistic is a statistic computed on a subgraph.
  - For each vertex v, consider the closed k-neighborhood $N_k[v]$.
  - Compute an invariant on the induced subgraph $\Omega(N_k[v])$.
- This can be standardized across recent time (accounting for “loud” vertices).
- A scan statistic is the maximum of the locality statistics.
- This is designed to detect local changes in the graph.

Formation of NATO



Fusing Graphs

- Spectral method (assumes the graphs are labeled and share the vertex set):
  - Form the combined $n \times 2n$ matrix: $A = (A_1, A_2)$. Here A_i may be the adjacency matrix (possibly augmented), the Laplacian, or the incidence matrix.
  - Project the vertices via singular value decomposition (spectral embedding).
  - If the A_i is the incidence matrix, this method extends directly to hypergraphs.
- Note that covariates can be incorporated by direct product as in the projection method, or by appending the covariates onto the matrix A.

Measuring Change

- Determining if there has been a change in a time series of graphs can be done in several ways:
  - Measure the number of changes (difference in the adjacencies). This is a global measure.
  - Measure the change in features (invariants) extracted from the graph (also global).
  - Measure the change in projected points (can be local: each vertex contributes).
- Note that since we are embedding the vertices, the spectral method can detect changes in a subset of the vertices.

Change in Adjacency

- Compare the pairs of adjacency matrices:
  - Given two graphs represented by their adjacency matrices: A_{t-1}, A_t
  - Compute $\sum |A_{t-1} - A_t|$, where the sum is computed over vertices containing at least one edge in one of the graphs.
- Look for large values.

Spectral Embedding

- There are various spectral embedding approaches.
- We will investigate one of the simplest, from the random dot product model:
  - Given two graphs represented by their adjacency matrices: A_G, A_H, augment them with the row sums:

$$\begin{aligned} D_G &= \text{diag}(\text{deg}_i/(n_t - 1)) \\ D_H &= \text{diag}(\text{rowSum}_i/(n_t - 1)) \\ B_G &= D_G + A_G \\ B_H &= D_H + A_H \\ A &= (B_G, B_H). \end{aligned}$$

  - Decompose $A = XX'$ via the singular value decomposition.
- This is related to the random dot product model of random graphs.

Random Embedding

- Assign to each vertex v a vector $U_v \in \mathbb{R}^d$.
- Associate to each edge $e = vw$ the vector $g(U_v, U_w) \equiv g(e)$ (for example $g(U_v, U_w) = U_v * U_w$ where multiplication is term-by-term). If the edge is weighted, g operates on the weight as well (for example by multiplication).
- Let $s(v)$ be a subgraph containing v (for example, edges incident on v), and $E_{s(v)}$ its edge set.
- Associate to each vertex v the vector

$$X_v = \frac{1}{|E_{s(v)}|} \sum_{e \in E_{s(v)}} g(e).$$

- So, each vertex gets mapped to the centroid of its "associated" edges.

Random Embedding

- U_V can be random, but fixed for all time, or it can correspond to (a function of) the covariates, and thus vary in time.
- Note that the method extends to the general case. If

$$A = (B_G, B_H),$$

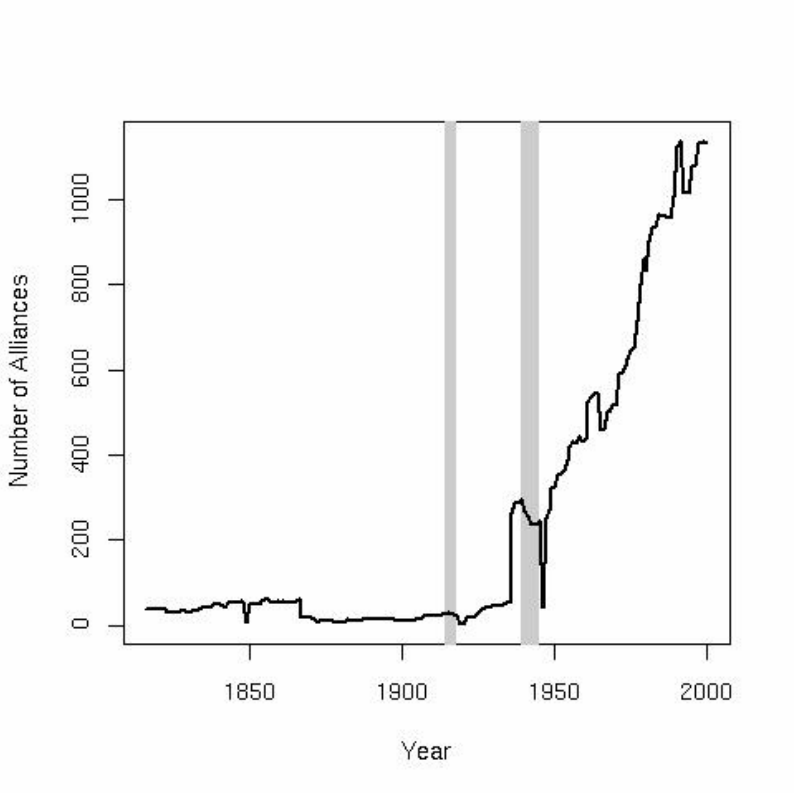
  each column of A is considered an edge, each edge is projected according to g, and the vertices are projected to the centroid of their edges.
- Note that for some applications, the ability to embed both vertices and edges in the same space may be important.

The Alliance Data

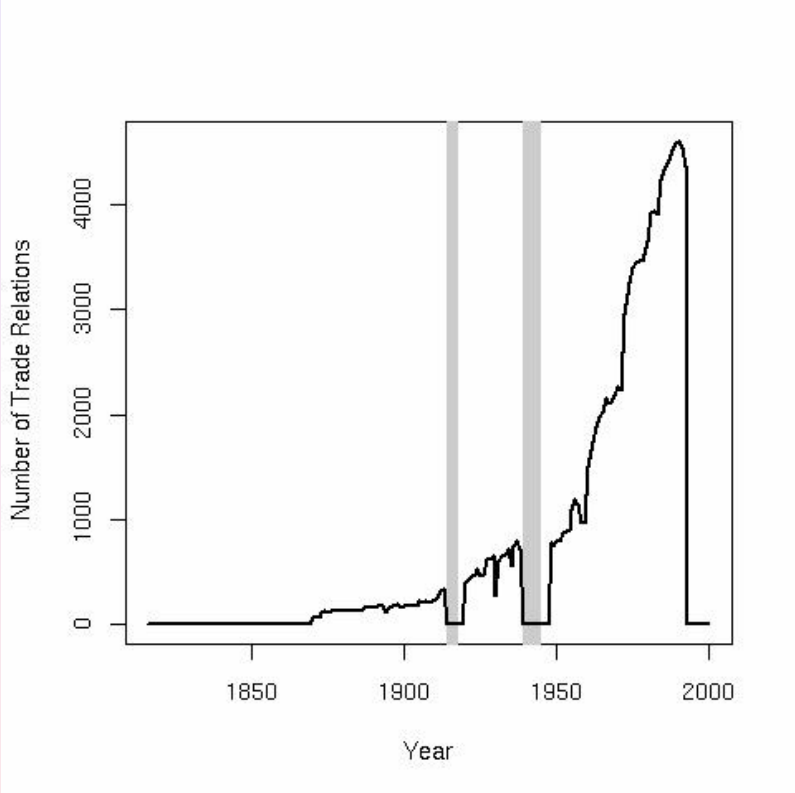
- Obtained from the Correlates of War (COW) web site (`www.correlatesofwar.org`).
- Corresponds to nations from 1816 to 2000.
- Consists of two graphs:
  - Alliances between countries.
  - Total trade between countries (only available from 1871 on).
- Each country has several covariates measured each year:
  - Military Expenditures and Personnel
  - Energy, and Iron and Steel production
  - Total Population and Urban Population
- This is a subset of the data available at COW.

Some Statistics: Size

Alliances



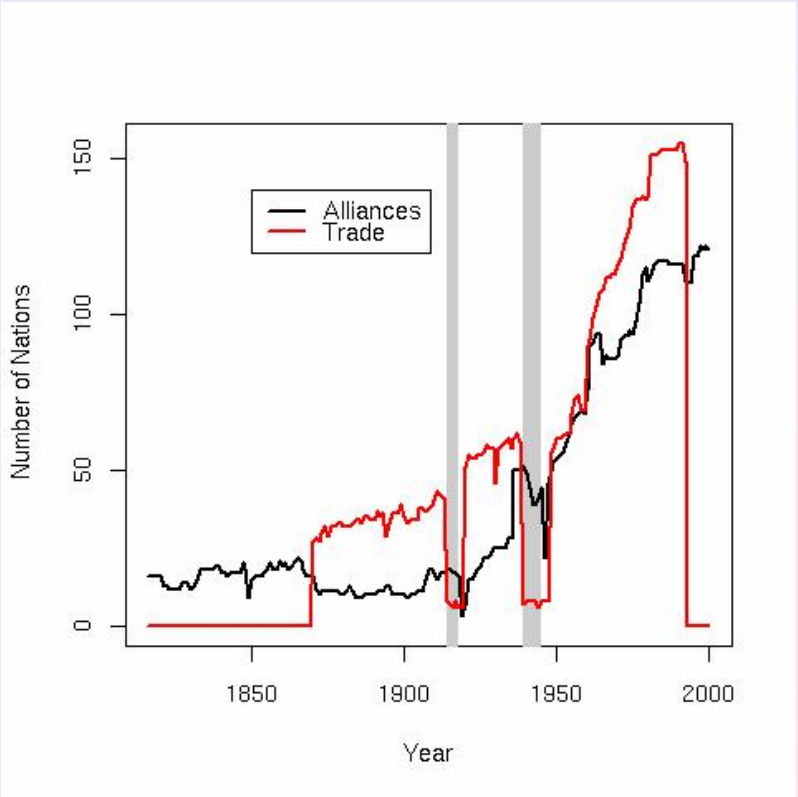
Trade



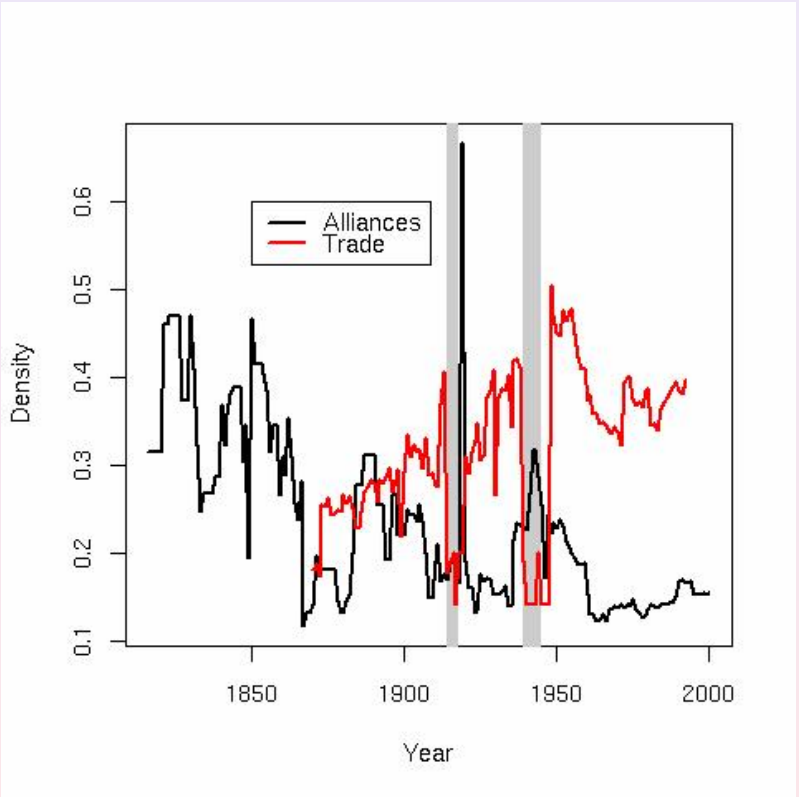
In all plots, the gray regions are the two world wars.

Some Statistics: Order and Density

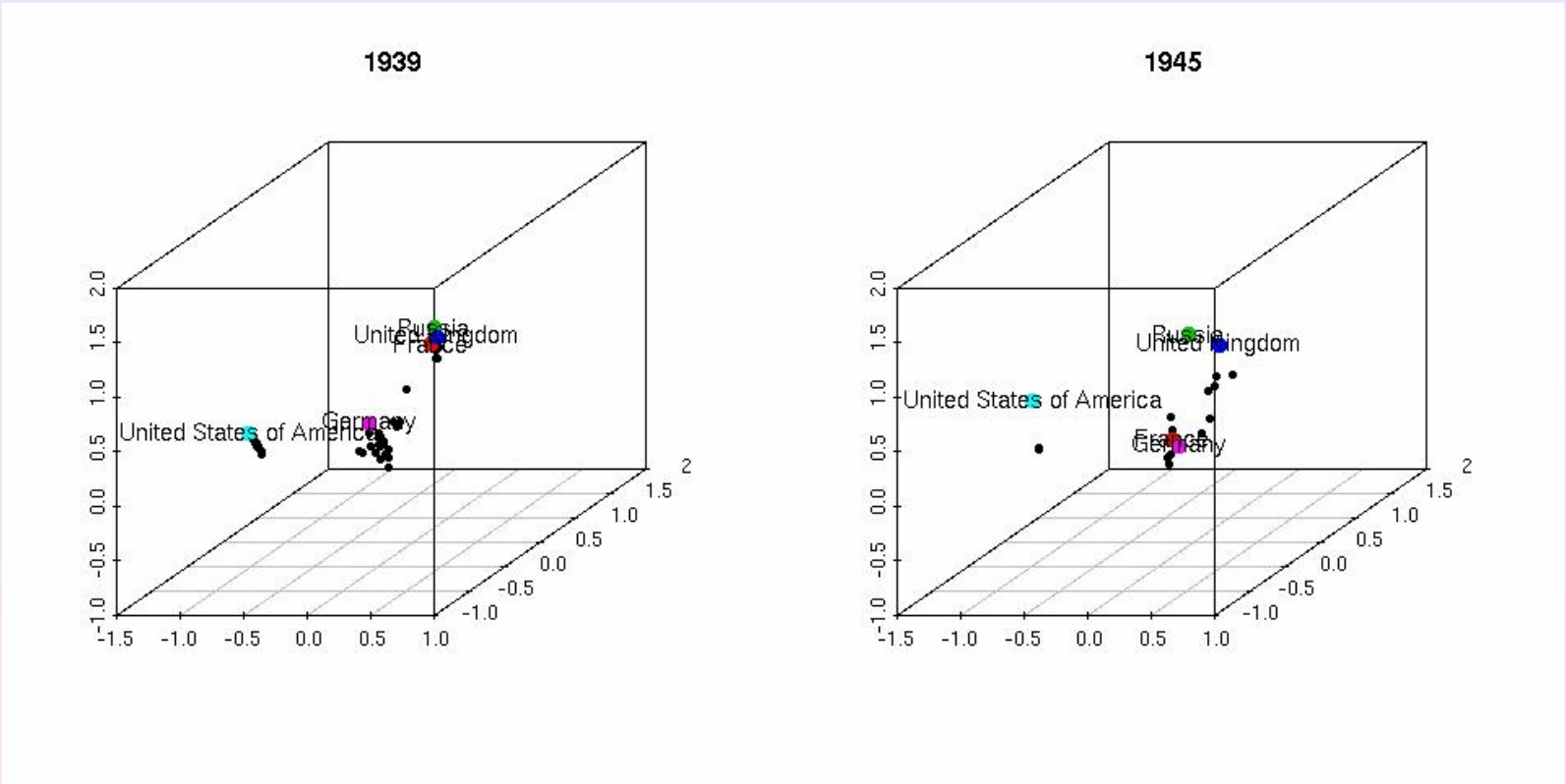
Order



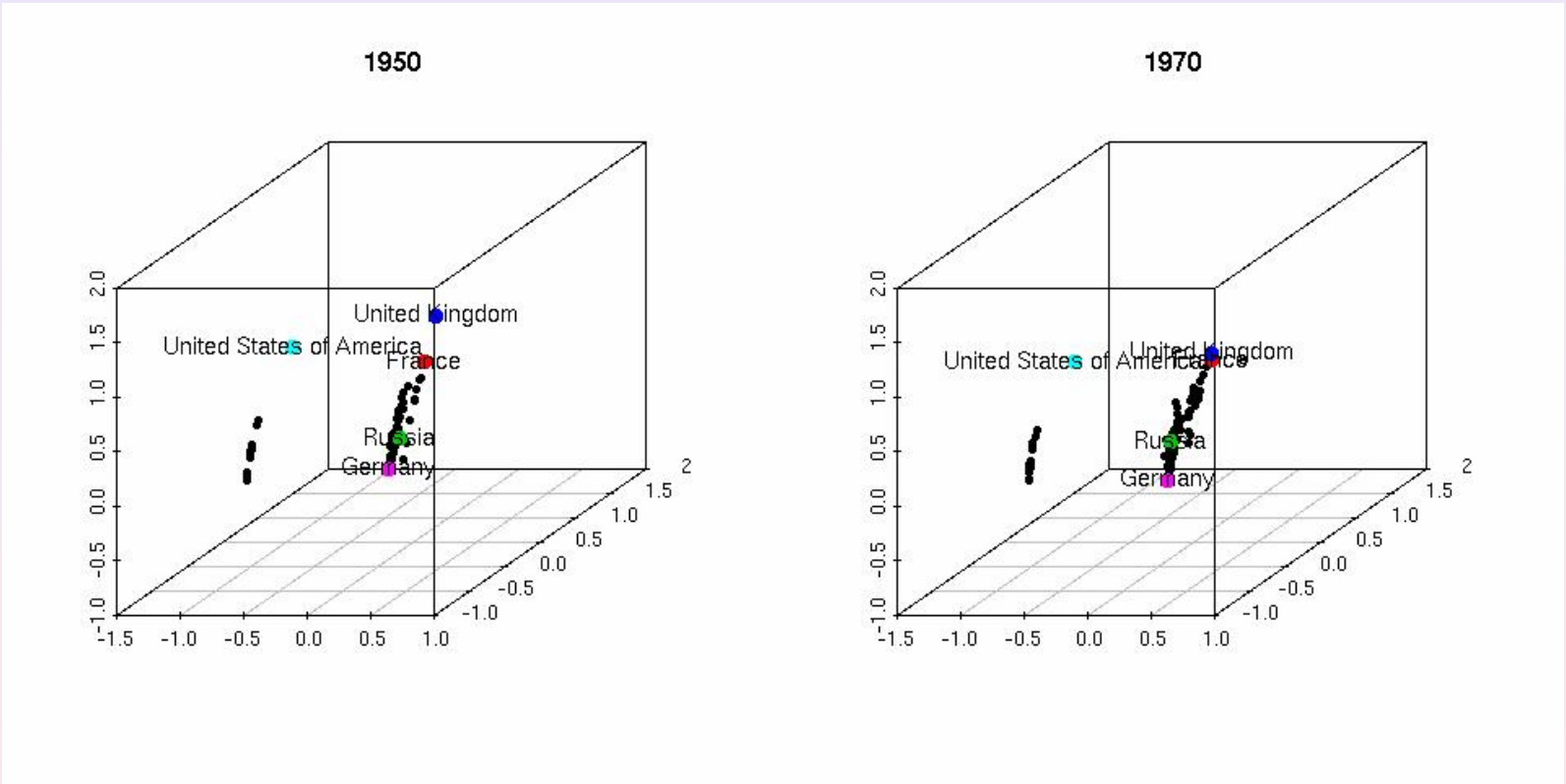
Density



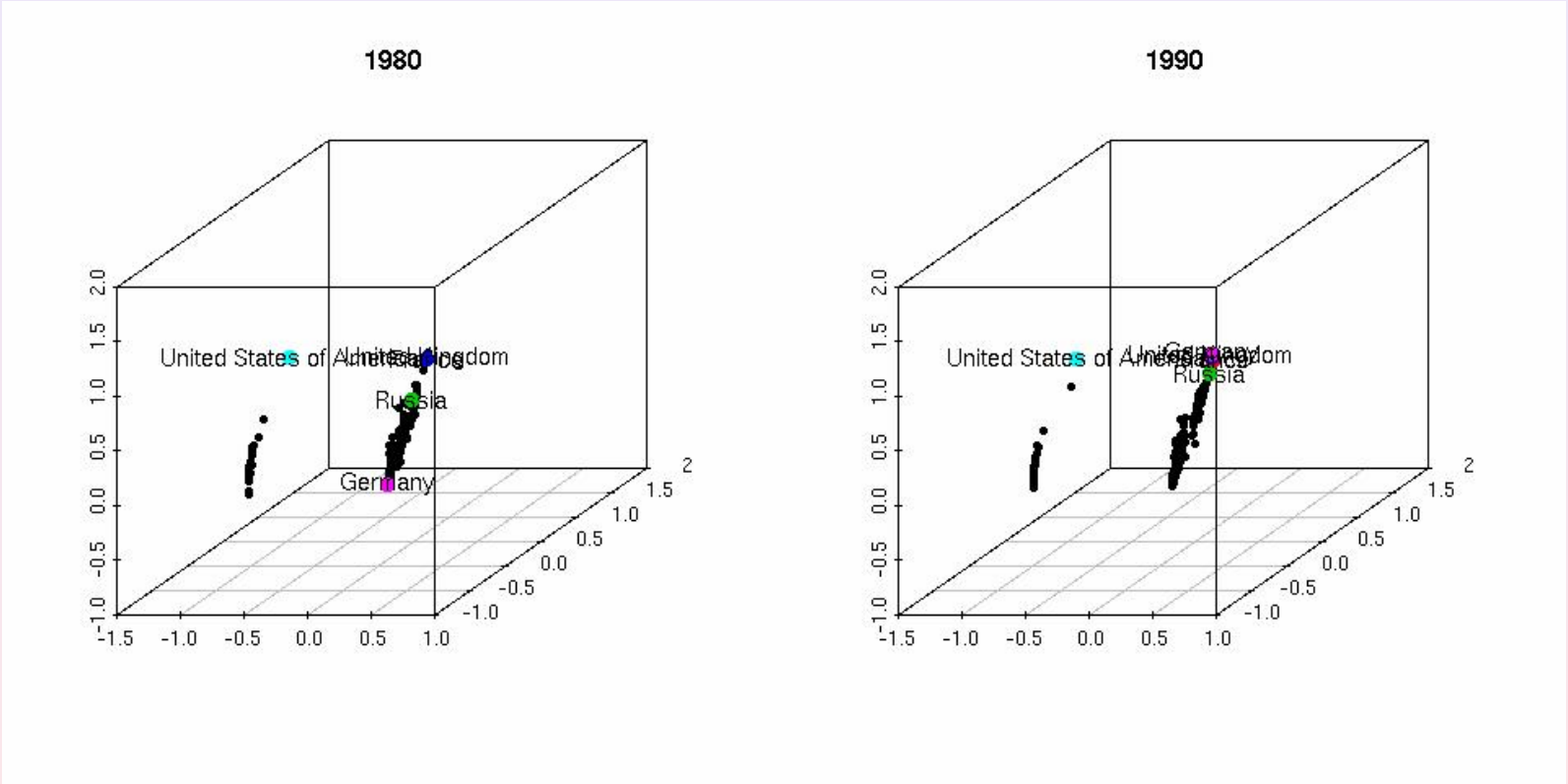
World War II


1939
United States of America
Germany
1945
United States of America

Post World War II



Post Cold War



Discussion

- We've seen several methods for embedding (the vertices of) multiple graphs, and covariates, into a space for performing inference.
- Similar methods are appropriate for embedding the graphs as entities.
- We have not discussed issues of weighting the different types of information.
- For specific problems, investigating when fusion is superior to marginals is of considerable interest.
- This is not the only approach: model based approaches are often possible for specific applications. Spectral approaches are to some degree "nonparametric".

Data Challenge

- Next year the Section on Statistics in Defense and National Security will hold a data challenge.
- Data collected from the Correlates of War web site (`www.correlatesofwar.org`).