I couldn't attend all classes this semester and I was very busy with my thesis. I'm thus only able to read through the papers now. Here are my answers to the questions / discussions from the presentations.

1 Airport transportation network

The authors chose to use degree (how many edges has each node) and betweenness (how many shortest path lengths pass through this node). For both the degree- as well as the betweenness-distribution, they observed a truncated power law. For a scale free network, we would expect a (non-truncated) power law. By comparing the two distributions, we see that there are a lot of cities with a high betweenness and low degree. The authors attribute this to the existence of communities in the network. E.g. there are some airports in Alaska and all of them are connected to the world through Anchorage. This results in Anchorage having a high betweenness.

One could question the validity of these results: The authors assume an unweighted network structure. A flight with 2 people, from let's say Anchorage to a neighbouring town, is as important in the analysis as a flight with 500 people from New York to Paris. Furthermore, the reliability of the results can be questioned: the data only holds flights from US to international airports, flights from international airports to the US airports and flights from other airports to airports that are connected to the US. We will thus never find a flight from Antwerp to Ajaccio as none of those fly to the US directly. Applying the same methods on a different data-set will thus yield different results.

2 Visualisation techniques

For each of the following: What is novel in the proposed technique? What are the criteria used to define a good network drawing?

Force Atlas 2: Jacomy et al. (2014)

Network: Scale free, 10-10.000 nodes (max 100.000 nodes).

New: Implementation of various techniques such as Barnes Hut simulation, degree-dependent repulsive force, and local and global adaptive temperatures. Focus on user experience. Force Atlas 2 provides live spacialization, meaning that the layout stops at the user's request. This design decision allows the user to play with the network in an interactive way and get more intuition about what is happening.

Technique: Force-directed layout

OpenOrd (Martin et al. 2011)

Network: large scale, undirected graphs

New: The algorithm is the only one available which incorporates all three of the following ideas: a multilevel approach, node clustering, and a parallel implementation. This makes it specialized for drawing large-scale real-world graphs. In essence, the algorithm first coarsens the graph by clustering the nodes in multiple steps (coarser and coarser). Then it calculates the force layout for this graph. In a next step, it refines the graph. Nodes that were previously merged, are now placed on the position where the merged node was located. By refining more and more, we get a force directed visualisation of the large-scale graph at a fine level.

Technique: Edge-cutting, a multi-level approach, average-link clustering, and a parallel implementation. At each level, vertices are grouped using force-directed layout and average-link clustering. The clustered vertices are then re-drawn and the process is repeated. When a suitable drawing of the coarsened graph is obtained, the algorithm is reversed to obtain a drawing of the original graph. This approach results in layouts of large graphs which incorporate both local and global structure.

VOS: van Eck et al. (2010)

Network: Medium to large data sets (for sets with <100 items, the advantage of VOS is less apparent)

New: A new dimensional reduction method that reduces artefacts compared to MDS. These artifacts consist of: 1) the tendency to locate the most important items in the centre of a map and less important items in the periphery. 2) the tendency to locate items in a circular structure. VOS shows better, more compact, clustering than other techniques.

Technique: The idea of VOS is to minimize a weighted sum of the squared distances between all pairs of items. The squared distance between a pair of items is weighted by the similarity between the items. To avoid trivial solutions in which all items have the same location, the constraint is imposed that the average distance between two items must be equal to one. The similarities are typically calculated using the association strength. This association strength is the co-occurrence of element i and j divided by the multiplication of the occurrences of both elements.

3 Traversals

Newman: A measure of betweenness centrality based on random walks

Problems addressed with flow betweenness: The conventional way of calculating betweenness consists out of counting the total number of shortest paths that pass through a given node. This way of thinking assumes that information only spreads using these shortest (geodesic) paths. In real life applications, this is often not the case: e.g. we can have two large groups of nodes which are connected by two nodes (A and B). These two nodes on their turn are connected to a third node (with degree = 2). In the conventional calculation of betweenness, c would get a betweenness score of 0, as all shortest paths pass through A and B. In real life, such a third party often occurs and could play an important role in the information flow.

Implications of only using the net number of passes through a node: Imagine that a random walk keeps jumping back and forth between a node and its neighbours. Then this node will build up a high random walk betweenness. By cancelling out the attribution to the betweenness if a random walk arrives in the same node again, we get the net number of passes.

Think of an example of why we want to calculate betweenness: As being mentioned earlier in this document, nodes with high betweenness and low degree could surface communities in the network. E.g. I have my group of friends, a friend from primary school has his group of friends. Me and him are connected in the graph and serve as a bridge for paths from his group of friends to my group of friends.

4 Graph Databases

Graph Analysis: Do we have to reinvent the weel?

What is the main argument in this paper?
Graph databases are over hyped and, although being beneficial for a small number of tasks, everything they can do can be done by using conventional databases (relational).

How does this relate to this lecture? + What would a Neo4J sales person answer to this paper?
In this lecture, we saw the pros and cons of graph databases. One of the major benefits of this type of databases is the intuitive querying and visualisation capability. In the article, they make use of SQL queries, which tend to become very complicated very quickly. Furthermore, the article only tackles the shortest path problem. This problem only takes into account the nodes and edges of one type (e.g. users and friendships). The true power of graph databases shows itself in the capability of linking multiple types of nodes and by giving attributes to the nodes and edges. E.g. the Facebook graph data base is an excellent example thereof. Lastly (this is a more subjective reason), I think that Facebook choosing to build its infrastructure on top of a graph data base proves a lot about the value of this kind of databases.

What properties of a social network are relevant for this comparison?
I would add the calculation of graph properties (local and global) to the comparison. Furthermore, I would try to make visualisations using both databases and compare runtimes. Also, very complex queries would form an interesting challenge to write in SQL whereas they can 'easily' be written in e.g. Cypher. Furthermore, this article shows the performance / strength of Neo4J on complex queries:
https://www.researchgate.net/publication/308969172_Running_Complex_Queries_on_a_Graph_Database_A_Performance_Evaluation_of_Neo4j

Would you use Dijkstra for a routing problem? Why?
No, Dijkstra only takes into account the cost/weight of the edges. This could lead to first exploring a lot of nodes (with lower weight) before starting to 'move' into the right direction. An algorithm like A* alleviates this problem by introducing a heuristic. However, A* is only applicable if we can define a relevant heuristic function.

5 Parallel & distributed graph processing

Investigate the implementation of a graph algorithm within a DGP-framework (Thijs, 2017)

This paper uses a messaging procedure and calculation functions. Which programming model is applied?
The vertex centric approach is used: At each super step, we first update the vertex value based on some user defined function and previous value (and incoming message if it's not the first super step), propagate this value over the edges (sending message) and collect the value (incoming message) at the destination vertices.

What can you say about memory consumption of this algorithm?
In this approach, memory consumption can be unbalanced over different worker nodes in the distributed network: High degree papers will take longer time to compute + request more memory. This problem can be alleviated when using the GAS model.

GraphX: A Resilient Distributed Graph System on Spark

Does their Pregel implementation follow the genuine Pregel model?
Not completely. They substitute the 'messaging component' of the vertex centric approach by a message generating function. From the paper: *"This function takes an edge containing the source and target attributes and returns a message or void indicating the absence of a message. By lifting the message construction out of the vertex-program, we are able to achieve a more efficient execution than the original Pregel framework and leverage the vertex-cut representation. Moreover, message construction for a single vertex can be distributed over the cluster, moved to the receiving machine, and executed in an efficient order."*

How many functions are required for the implementation of the Pregel function.
3 functions are used: *"vprogf function which takes a vertex and a message and returns a new attribute value for that vertex, a sendMsgf function which computes the new message along each edge, and a combinef function which is used to combine messages to the same vertex"*

How do they achieve a fair distribution of task executions?
They make use of random vertex cuts instead of random edge cuts. Through a simple analysis it can be shown that for the power-law degree distributions found in real-world graphs, random vertex-cuts can be orders of magnitude more efficient than random edge-cuts. The vertex cuts ensure equal distribution of edges across workers.

6 Random networks

Evolution of the Social Network of Scientific Collaboration

Degree
The degree distribution follows a power law; new authors collaborate with well-known authors. The degree of these well-known authors thus only increases. Furthermore, the degree increases over time. This can be attributed to the fact that no authors leave the system.

Average path length
In the article, this measure is called the average separation of the network. The calculation can be time consuming for large networks, therefor sampling methods are used. For the considered network, the average separation d decreases over time (and thus increasing network size). This is in contrast with theoretical findings that predict an increase of d for bigger networks. This finding can be attributed to 1) papers written by authors that were previously part of the database increase the interconnectivity thus decreasing the diameter, and 2) the authors don't have access to the full database, but only starting from year 1991. Such incomplete dataset could result in an apparently decreasing separation even if otherwise for the full system the separation increases

Clustering coefficient & connectedness & connected component
In simple terms the clustering coefficient of a node in the co-authorship network tells us how much a node's collaborators are willing to collaborate with each other, and it represents the probability that two of its collaborators wrote a paper together. The authors find that the clustering coefficient decreases over time. This finding is in line with the decreasing average path length => higher interconnectedness. Furthermore, they found that the relative size of the clusters increases with one big component appearing. This can be explained as follows: *In most research fields, apart from a very small fraction of authors that do not collaborate, all authors belong to a single giant cluster*

from the very early stages of the field. That is, the system is almost fully connected from the very first moment. The only reason why the giant cluster in our case grows so dramatically in the first several years is that we are missing the information on the network topology before 1991. Again, this can be attributed to the fact that the authors miss information from before 1991.

Overall, connectedness only increases because over time connections are only added, not deleted.

7 Clustering

Louvain Method

Agglomerative or divisive?
Louvain method is agglomerative

How are cuts/merged introduced?
1) It starts by assigning each node to its own cluster. In a next step, we iterate over each node, assign it to a cluster of one of its neighbours and calculate the change in modularity. This calculation is done for every neighbour of the node. If all changes in modularity are negative, the node stays in its current cluster. If one ore more changes are positive, the node moves to the cluster with the highest change in modularity. This process is repeated until no more changes happen.
2) A new, coarse-grained, network is built from the clustering obtained in 1). The newly discovered clusters thus form the nodes.

By repeating step 1) and 2), a clustering on different levels is obtained. The algorithm stops when no more modularity-optimising changes happen.

How is link calculated?
As mentioned before, we deal with an iterative process. The result of each iteration yields a clustering at a specific coarseness level. A node can thus be linked to different clusters in these different levels.

Time or Space Complexity?
Time complexity: O(nlog(n)) with n the number of nodes

What about Resolution Limit?
For larger networks, the Louvain method doesn't stop with the "intuitive" communities. Instead, there's a second pass through the community modification and coarse-graining stages, in which several of the intuitive communities are merged together. This is a general problem with modularity optimization algorithms; they have trouble detecting small communities in large networks. It's a virtue of the Louvain method that something close to the intuitive community structure is available as an intermediate step in the process. We can thus always go back to this intermediate step.