Estimating Edge Weights in Dynamic Graphs Based on Events

Pascal Held **Christian Braune** **Rudolf Kruse**

Otto-von-Guericke University of Magdeburg
Faculty of Computer Science
Department of Knowledge Processing and Language Engineering
Universitätsplatz 2, D-39106 Magdeburg
Tel.: +49 391 67 58718
Fax: +49 391 67 12018
E-Mail: {pheld,cbraune,kruse}@iws.cs.uni-magdeburg.de

Abstract

Dynamic graphs are ubiquitous in real world applications. They can be found, e.g. in biology, neuroscience, computer science, medicine, social networks, the World Wide Web. There is a great necessity and interest in analyzing these dynamic graphs efficiently. Typically, analysis methods from classical data mining and network theory have been studied separately in different fields of research. For dealing with complex networks in real world applications there is a need to perform interdisciplinary research by combining techniques of different fields. In this paper, we analyze dynamic graphs from the social science. For the representation of edge weights in a social network graph we propose a method to efficiently represent the strength of a relation between two entities based on events involving both entities. The Butterworth filter is used to describe the continuous relation that can otherwise only be represented by a series of discrete events.

1 Introduction

Complex dynamic networks are ubiquitous. They can be found, e.g. in biology [1], neuroscience [2], computer science [3], medicine [4], social networks [5], and the World Wide Web [6]. There is a great necessity and interest in analyzing these dynamic graphs efficiently as patterns inside of these structures might reveal knowledge about the underlying system. Classically, analysis methods from both network theory and knowledge discovery in databases have been studied separately in different fields of research. The analysis of complex networks as they occur in real world applications can be supported by combining techniques of these two fields [7, 8]. In this paper, we present a real-world problem of dynamic graphs from the point of social sciences. We propose a method to efficiently represent the strength of a relation between two entities based on events involving both entities.

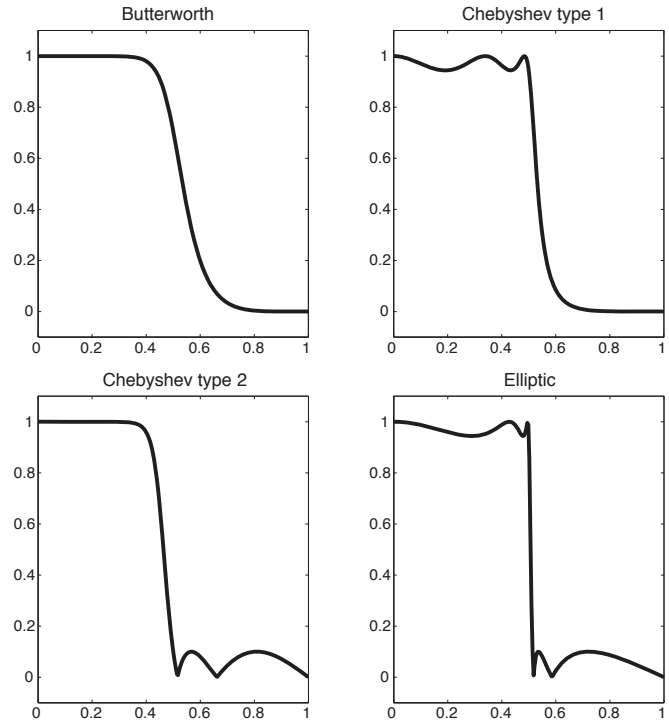

Figure 1: Rippling effects for different filter types

1.1 Butterworth Filtering

Representing the structure of a social network not only by the *friendship* relation (i.e. nodes represent persons, edges represent the relation), which results in a more or less static description of the graph, but also by adding weights to such edges where the weight reflects the amount of activity between the two corresponding nodes, requires a way to describe this activity. Event-based weighting of edges in a social graph could be accomplished by simply storing all the timestamps at which events between two nodes occurred. Obviously this approach would become unfeasible very soon due to the amount of memory required for such a procedure. An additional disadvantage of such an approach would be that, while we can make statements about the point in time when an event occurred. If possible at all, we can roughly estimate the current weight that should be assigned to an edge at a given point in time. Operations like a sliding average would be able to adapt to such a problem with the major drawback, that only a small time frame can be used to determine the current average due to memory restrictions — no further information about the past is available if only such a value is used.

Obviously using all the bins to calculate an interpolating polynom would suffice to give a continuous time representation of the data along with a

very high precision. Yet is memory efficiency of such an approach vastly high if such information needs to be stored for each and every each in a (possibly even fully connected) graph. Beside this challenge, interpolating between each timebin is very sensitive to outliers. From electronic signal processing the Butterworth filter is a well-known variant of an infinite impulse response filter that produces an output signal as response to its input signal without causing the rippling effects in the number of frequencies removed from the signal (see Figure 1 other filters suffer from. The resulting signal that such a filter calculates can in some terms be interpreted as an approximation of the interpolating polynom, enveloping the originally discrete signal.

In general such a filter is defined by two sets of coefficients B and A. These sets depending on the selected passband frequency f. The filter's response y for a signal x at the bin n can be obtained by computing

$$y_n = \sum_{i=1}^{n_b} (b_i \cdot x_{n-(i-1)}) - \sum_{j=2}^{n_a} (a_j \cdot y_{n-(j-1)}), \text{ where}$$

$$\{b_1, \ldots, b_{n_b}\} = B \text{ and } \{a_1 = 1, a_2, \ldots, a_{n_a}\} = A.$$

This recursive representation makes it possible to avoid enumerating all signal values from negative to positive infinity.

Other parameters that either influence the shape of the resulting curve or the set of possible edges that are considered are:

- Step width: Length of time bin
- Grade: The grade of the filter determines the two sets B, A of coefficients responsible for the shape of the resulting filter response. The number of coefficients depends directly on the grade and describes how many past signal (and response) values are considered for the calculation.

2 Related Work

2.1 Social Network Analysis

Social network analysis has already been popular long before websites like Facebook, XING or Google+ — now commonly understood/known as social networks — were launched. In [9] a comprehensive approach of modeling social network data as (un)directed graphs has been proposed and has

been widely accepted. Over the years a lot of research has been performed on e.g. cohesiveness of groups of members in social graphs [10] or segmentation of social networks [5]. In [11] or [12] web communities were targeted by the research and graph-based algorithms were used to distinguish between different groups. In [13] the authors analyzed mobile phone communication and used the sum of calls from each subscriber as weights in a graph representation. Social networks like Facebook have been the subject of analysis in [14] where a snapshot of the friendship relation for five American universities was analyzed by means of graph analysis tools. All these methods have in common that they use a static representation of the social graph underlying the respective social network.

Attempts have been made to infer information from dynamic graphs (e.g. in [15]) but they either restrict themselves to fairly simple questions like connectivity or to path finding problems in order to cope with the changing structure of the graph. Every binning leads to a loss of information, namely the exact time when an event has happened. Such approaches do not take into account the frequency with which events occur but rather lists their absolute number.

2.2 Butterworth Filter

The Butterworth filter [16] is one of the best-known infinite impulse response filters. One of its most interesting features is its flat frequency response, i.e. it does not generate rippling effects, when the signal strength changes. Interpreting the binned events of a social graph as a time- and strength-discretized signal the filter response of such a Butterworth filter should have the desired properties that events (dirac pulses) can be binned while keeping some information on the frequency.

In [17] the authors describe how the Butterworth filter can be used to reduce the computation time in online electroenzephalograms (EEG) while in [18] the Butterworth filter is reduced to describe trends in oscillating oceanographic data. This is particularly interesting because the number of messages sent in a social graph w.r.t. their time bins can also be seen as a oscillating (or at least fluctuating) signal that we want to represent by the filter output.

2.3 Enron Data Set

For the analysis and validation of our method we used the Enron data set[1]. The Enron mail corpus is a collection of email boxes from 150 employees of the Enron company. It contains the mail communication of these employees in a timeframe of about one year [19]. Like in every group of people there are subgroups (clusters) of people which are communicating together more often then with other employees. We removed both external contacts from the data (Enron employees sending mails to non-Enron employees) and all mail contacts with mailing lists. Duplicates (*firstname.lastname* vs. *firstnamelastname*) have been reduced to one single node and mails that were sent to several users at once were treated as separate events (such that a mail sent from A to B and C was considered as two identical mails that were sent from A to B and from A to C).

3 Methodology

As the Butterworth filter produces a continuous signal we want the filter response to be $^{\#msgs}/_{l}$ for a time step of length l to give a better generalization. This restriction and the fact that it produces an equal sum of values over a continuous time span directly lead to two adversing goals in finding an optimal frequency to describe the filter:

1. Minimize the difference between the discretely binned signal and the filter response.
2. Find a frequency $f \in (0, 1)$ that produces a continuous, smooth and locally linear approximation of the signal (i.e. has only a few local extrema).

While the number of extrema can be reduced by lowering the passband frequency (which at some point will result in a nearly constant response), the error can be reduced by increasing it. This interrelation is illustrated by Fig. 2, which shows the filter response for three different passband frequencies when applied to event data from the Enron data set (for simplicity, the data were treated as if belonging to a single edge). Of course the data should be split up into the real edges for any further analysis.

To evaluate the total complexity of the resulting model depending on the frequency we adapted Akaike and Bayesian Information Criterion (AIC /

[1] Obtained from `http://www.cs.cmu.edu/~enron/`.

BIC) [20, p. 110] to include the parameters we want to optimize on. For any given frequency f we can compute the mean squared error (MSE) for the resulting signal and count the number of extrema. The number n_e of extrema can be used as a measure for the complexity of the resulting curve by assuming we have to store this curve as a polynomial with a degree of $n_e + 1$.

Thus, the objective functions we need to minimize are

$$AIC(f) = 2k - 2 \cdot \ln(L) \quad \text{and} \quad BIC(f) = k \cdot \ln(n) - 2 \cdot \ln(L),$$

respectively, where k is the number of parameters and L is the likelihood of the model. Assuming the error in the model is standard normally distributed both functions can be simplified to [20, p. 110]

$$AIC(f) = 2k + n \cdot \ln(MSE) \quad \text{and} \quad BIC(f) = k \cdot \ln(n) + n \cdot \ln(MSE).$$

The MSE for a frequency f can be computed from the signal X with bins $x_1, \ldots, x_t, t = \|X\|$ and the produced (w.r.t. the frequency f) filter response $Y_f = (y_1, \ldots, y_t)$ as

$$MSE_f(X, Y_f) = \frac{1}{t} \sum_{i=1}^{t} (x_i - y_i)^2,$$

the number of parameters equals $n_e + 2$.

These functions can then be optimized using standard optimization techniques like simulated annealing [21] or gradient descent [22] techniques. According to the resulting shape of the curves for both objective functions (see Fig. 3) the optimal passband frequency for the depicted example data set lies near $f = 0.0075$, when limited to $(0, 0.2]$. Higher frequencies result in a filter response that does never fulfill the smoothness requirement although they might lead to lower values of the objective functions (see also Sect. 4).

For the purpose of storing event information in a coherent way across multiple edges in the interaction graph it is useful to only use one global frequency to apply the same filter on all edges. This removes the need to store the individual filter parameters, and results in only storing the last few signal and filter response values to be able to calculate the new filter response with the given, global parameter set.

Analyzing the distribution of optimal frequencies yields the curve from Figure 4. The optimal frequency found by the optimization algorithm employed is marked as dotted line and almost coincides with the lower marker.

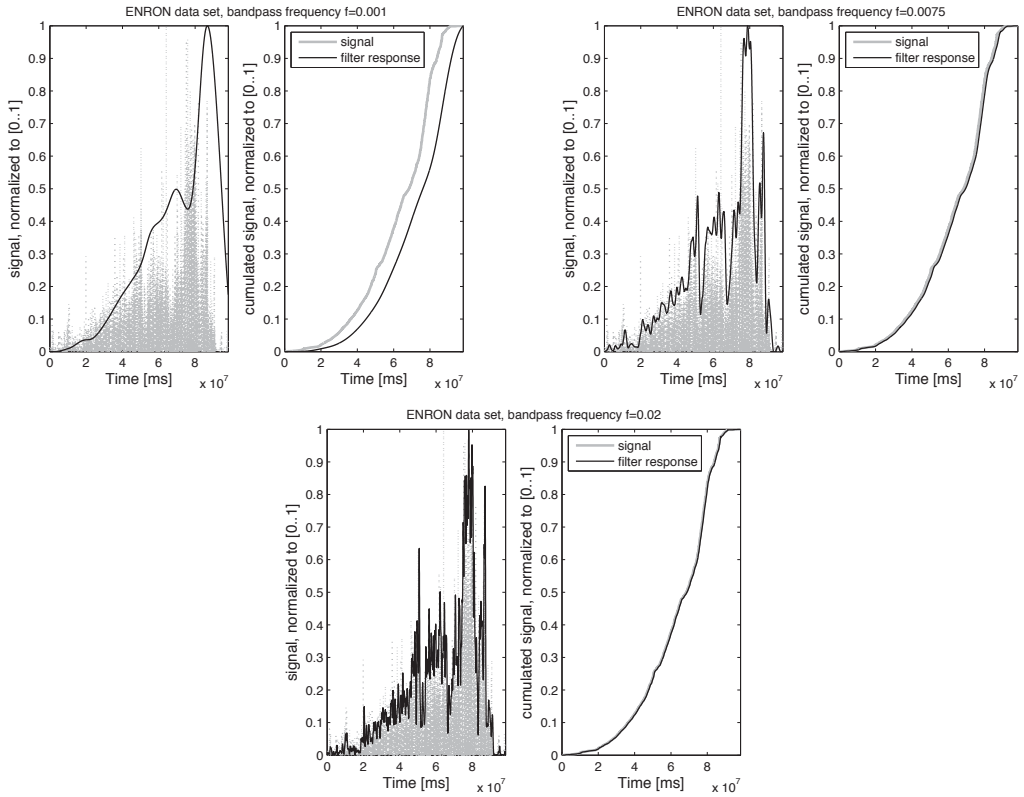

Figure 2: Filter response for different passband frequencies for Enron data set, time binning: 10'000 $^{ms}/_{bin}$.

The central marker gives the median of the distribution while the two outer markers indicate the points where the the curvature of the distribution goes toward zero. The frequencies falling between these outer markes roughly correspond to the frequencies which produce the minimal values for the AIC measure (see Figure 3) which makes this interval particularly interesting.

4 Evaluation/Results

4.1 Experimental Results

Naturally, when compared to a moving average filter, the MSE of our approach as compared to the original signal will be significantly higher (see Tbl. 1). As we never aimed at solely minimizing the error but also the complexity of the response signal, our method outperforms the moving average when using the BIC as optimization criterion. Though it may seem that the moving average performs better when considering AIC this is owed to this measure being biased toward models with very high complexity (i.e. very

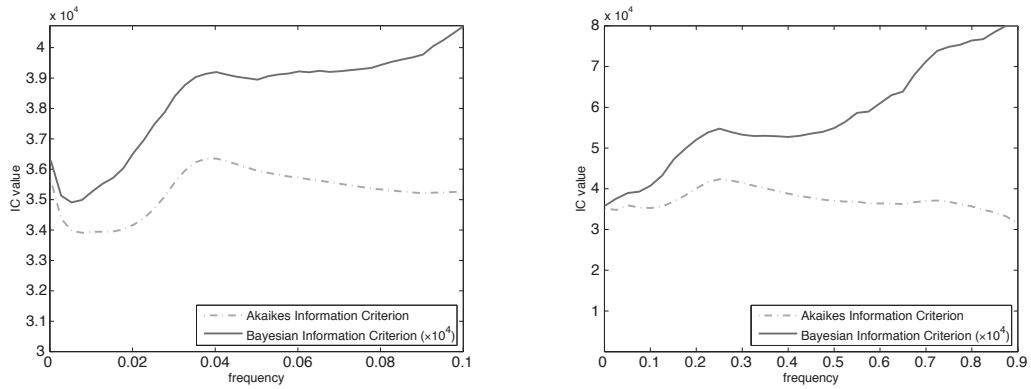

Figure 3: Left: AIC and BIC for the Enron data set, plotted against different passband frequencies (filter grade = 4). Right: Development of the two information criteria used over a larger interval. The interval $(0.9, 1]$ has been left out as the MSE heads towards zero in these cases which in turn leads to a term in AIC becoming negative.

low error). This effect can be seen on the right-hand side in Fig. 3. Here can be seen clearly, that the AIC has its true minimum for a frequency above 0.8. Hence we restricted optimization already to find an optimum only in the interval $(0, 0.2]$ where desirable (in terms of number of local extrema) results are achieved.

The comparison with other filters such as the Chebychev [23, p 36ff] or elliptical [23, p 44ff] may seem of interest here as well. Other filters tend to let frequencies pass that the Butterworth filter would not allow to pass. The optimal frequency derived for other filters will certainly differ from the one found by our method as the description because the set of parameters is langer for any of these filters while not changing the error measures very much. One of our goals was to increase the memory efficiency when storing edge weights so that the increased storage need for other filters is not justified by the at best slightly increased quality.

	Moving Average	Butterworth Filter
MSE	19.3163	33.3691
AIC	31068	33445
BIC	38814	34367

Table 1: Different evaluation measures to compare the moving average with the Butterworth Filter for the Enron data set.

As already described above, we observed during the evaluation of the error measures, that depending on the passband frequency the filter response shows some offset (see Fig. 5), which decreases with increasing frequency. We tried to find a best offset which could be applied to the filter response in order to reduce the overall error that occurs simply due to the offset.

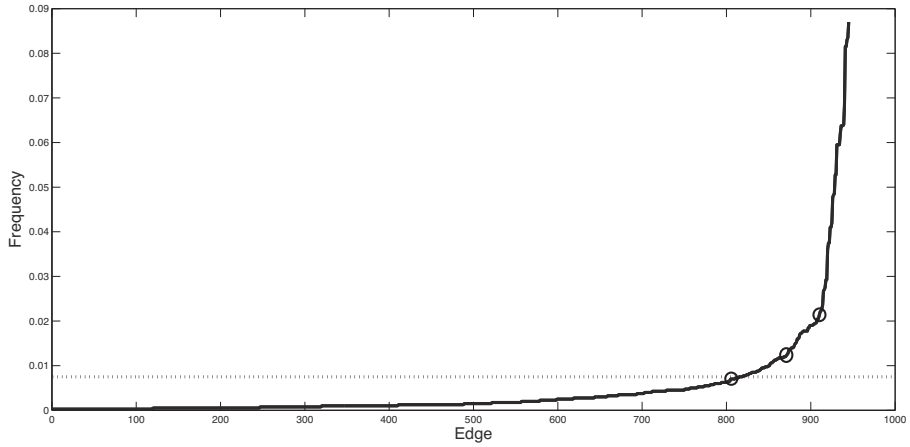

Figure 4: Distribution of optimal cutoff frequencies. Frequencies > 0.1 were dropped. The curvature of this distribution tends toward zero in the outer parts (outside the markers). The central marker gives the median of the distribution.

Plotting these offsets against the frequency they correspond to leads to Fig. 6.

Simple curve fitting yields that the optimal offset $o(f)$ can be calculated directly from the passband frequency used by the filter with the following formula: $o(f) = \left\lceil \frac{a}{f} + b \right\rceil$, where $a = 0.8347\ [0.8341, 0.8352]$ and $b = 0.3388\ [0.2085, 0.4691]$, (in brackets 95% confidence bounds). Actually the exponent for the factor f is not -1 but it is so close that we fixed it at -1 for simplicity. As we only have discrete bins, such a simplification seems reasonable as the following discretization of the result will obliterate most imprecisions. All of our experiments show that this formula seems to be independent from the given data set. That led us to the assumption to introduce this as a correction term into the objective function. This may be an important step for scenarios where the behavior of a user abruptly changes (increases or decreases). The filter will only adapt to this change after a certain amount of time. During adaption it will naturally deviate from the current process.

5 Conclusions

Applying a Butterworth filter can be used to describe event frequencies in event-based graphs as continuous signal as opposed to the inherent discrete nature of the signal. The resulting curve is continuous, smooth and without overfitting it gives a generally good approximation of the original

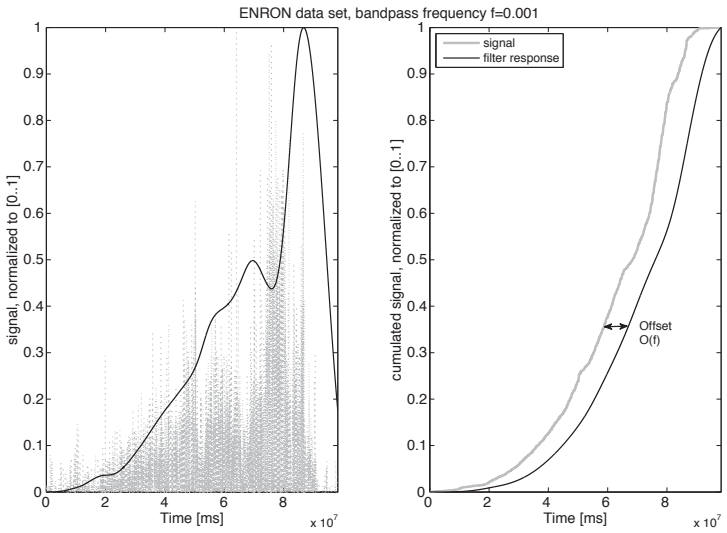

Figure 5: Offset between the true (left) and the response signal in the cumulated signal function (right).

signal. The filter itself can be described only by its coefficients and a few historical entries for each edge based on the grade of the filter, leading to an overall efficient memory usage. Still this approach leaves enough space for adjustments, e.g. by weighting the extrema or the MSE differently in the objective functions and thus leading to curves being either smoother or closer to the discrete signal. When changing the grade of the filter an even better approximation of the original curve is possible, decreasing the overall error at the cost of memory efficiency.

We investigated a complex network problem demanding hybrid analysis methods from both intelligent data analysis and network theory. We dealt with the analysis of dynamic graphs from social science. Firstly, we proposed a method to efficiently represent the strength of a relation between two entities based on events involving both entities. Using the Butterworth filter we were able to establish a continuous series of edge weights and thus graphs. Based on this data, the elements within the graph could be clustered. In Figure 7 such an clustering is proposed.

References

[1] Fischhoff, I. R.; Sundaresan, S. R.; Cordingley, J.; Larkin, H. M.; Sellier, M.; Rubenstein, D. I.: Social relationships and reproductive

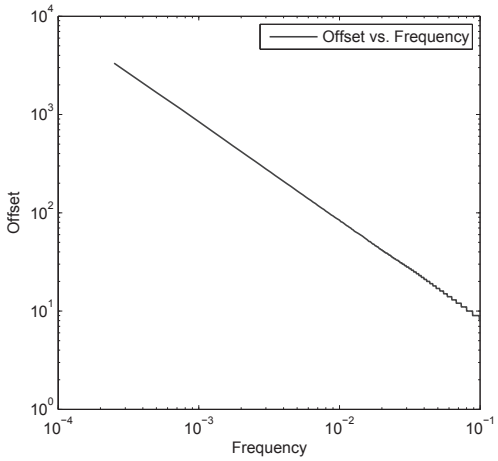

Figure 6: Optimal offset vs. passband frequency of Butterworth filter obtained by all experiments.

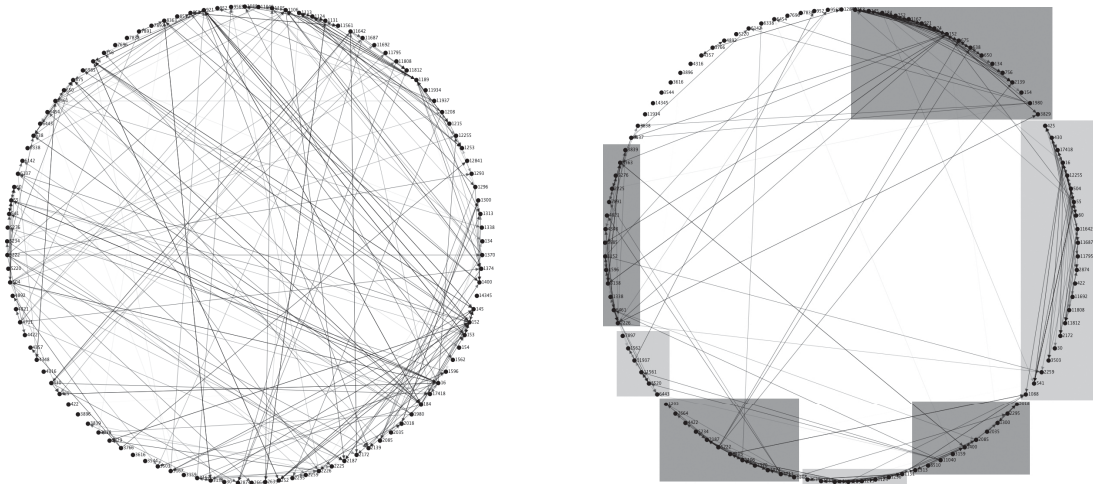

Figure 7: Snapshot of the Enron Data Set; left: unsorted; right: clustered and nodes sorted by clusters found

state influence leadership roles in movements of plains zebra, Equus burchellii. *Anim Behav* 73 (2007) 5, S. 825–831.

[2] Sporns, O.: *Networks of the Brain*. Cambridge, MA, USA: MIT Press. ISBN 978-0-262-01469-4. 2010.

[3] Faloutsos, M.; Faloutsos, P.; Faloutsos, C.: On power-law relationships of the Internet topology. In: *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '99, S. 251–262. New York, NY, USA: ACM. ISBN 1-58113-135-6. 1999.

[4] Pereira-Leal, J. B.; Enright, A. J.; Ouzounis, C. A.: Detection of functional modules from protein interaction networks. *Proteins: Struct Funct Bioinf* 54 (2004) 1, S. 49–57.

[5] Kumar, R.; Novak, J.; Tomkins, A.: Structure and evolution of online social networks. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, S. 611–617. New York, NY, USA: ACM. ISBN 1-59593-339-5. 2006.

[6] Kleinberg, J. M.; Kumar, R.; Raghavan, P.; Rajagopalan, S.; Tomkins, A. S.: The web as a graph: measurements, models, and methods. In: *Proceedings of the 5th annual international conference on Computing and combinatorics*, COCOON'99, S. 1–17. Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-66200-6. 1999.

[7] Zhang, J.: A Survey on Streaming Algorithms for Massive Graphs. In: *Managing and Mining Graph Data* (Aggrawal, C. C.; Wang, H., Hg.), Bd. 40 von *Advances in Database Systems*, S. 393–420. New York, NY, USA: Springer Science+Business Media, LLC. ISBN 978-1-4419-6044-3. 2010.

[8] Lahiri, M.; Berger-Wolf, T. Y.: Periodic subgraph mining in dynamic networks. *Knowl Inf Syst* 24 (2010), S. 467–497.

[9] Wassermann, S.; Faust, K.: *Social Network Analysis: Methods and Applications*, Bd. 8 von *Stuctural Analysis in the Social Sciences*. Cambridge, UK: Cambridge University Press. ISBN 0-521-38707-8. 1997.

[10] White, D. R.; Harary, F.: The Cohesiveness of Blocks In Social Networks: Node Connectivity and Conditional Density. *Sociol Methodol* 31 (2001) 1, S. 305–359.

[11] Dourisboure, Y.; Geraci, F.; Pellegrini, M.: Extraction and classification of dense communities in the web. In: *Proceedings of the 16th international conference on World Wide Web*, WWW '07, S. 461–470. New York, NY, USA: ACM. ISBN 978-1-59593-654-7. URL http://doi.acm.org/10.1145/1242572.1242635. 2007.

[12] Flake, G.; Lawrence, S.; Giles, C.; Coetzee, F.: Self-organization and identification of Web communities. *Computer* 35 (2002) 3, S. 66 –70.

[13] Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008 (2008) 10, S. P10008. URL http://stacks.iop.org/1742-5468/2008/i=10/a=P10008.

[14] Traud, A.; Kelsic, E.; Mucha, P.; Porter, M.: Comparing community structure to characteristics in online collegiate social networks. *Arxiv preprint ArXiv:0809.0690* (2008).

[15] Alberts, D.; Cattaneo, G.; Italiano, G. F.: An empirical study of dynamic graph algorithms. *J Exp Algoritm* 2 (1997).

[16] Butterworth, S.: On the Theory of Filter Amplifiers. *Wireless Engineer* 7 (1930), S. 536–541.

[17] Alarcon, G.; Guy, C.; Binnie, C.: A simple algorithm for a digital three-pole Butterworth filter of arbitrary cut-off frequency: application to digital electroencephalography. *Journal of neuroscience methods* 104 (2000) 1, S. 35–44.

[18] Roberts, J.; Roberts, T.: Use of the Butterworth low-pass filter for oceanographic data. *Journal of Geophysical Research* 83 (1978) C11, S. 5510–5514.

[19] Klimt, B.; Yang, Y.: The Enron Corpus: A New Dataset for Email Classification Research. In: *Machine Learning: ECML 2004* (Boulicaut, J.-F.; Esposito, F.; Giannotti, F.; Pedreschi, D., Hg.), Bd. 3201 von *Lecture Notes in Computer Science*, S. 217–226. Springer Berlin / Heidelberg. ISBN 978-3-540-23105-9. 2004.

[20] Berthold, M. R.; Borgelt, C.; Höppner, F.; Klawonn, F.: *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data*. Texts in Computer Science. Berlin: Springer-Verlag. 2010.

[21] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P.: Optimization by Simulated Annealing. *Science* 220 (1983) 4598, S. 671 –680.

[22] Snyman, J. A.: *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, Bd. 97 von *Applied Optimization*. New York, NY, USA: Springer Science+Business Media, Inc. ISBN 978-0-387-24348-1. 2005.

[23] Wangenheim, L.: *Aktive Filter und Oszillatoren: Entwurf und Schaltungstechnik mit integrierten Bausteinen; mit 26 Tabellen*. Springer. ISBN 9783540717393. 2008.