TKK Reports in Information and Computer Science
Espoo 2009 TKK-ICS-R20

GRAPH VISUALIZATION WITH LATENT VARIABLE MODELS

Kristian Nybo, Juuso Parkkinen and Samuel Kaski

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

TKK Reports in Information and Computer Science
Espoo 2009 TKK-ICS-R20

GRAPH VISUALIZATION WITH LATENT VARIABLE MODELS

Kristian Nybo, Juuso Parkkinen and Samuel Kaski

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O.Box 5400
FI-02015 TKK
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 451 1
Fax +358 9 451 3369
E-mail: series@ics.tkk.fi

ISBN 978-952-248-095-8 (Print)
ISBN 978-952-248-095-5 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)
URL: http://lib.tkk.fi/Reports/2009/isbn9789522480955.pdf

TKK ICS
Espoo 2009

ABSTRACT: Large graph layout design by choosing locations for the vertices on the plane, such that the drawn set of edges is understandable, is a tough problem. The goal is ill-defined and usually both optimization and evaluation criteria are only very indirectly related to the goal. We suggest a new and surprisingly effective visualization principle: Position nodes such that nearby nodes have similar link distributions. Since their edges are similar by definition, the edges will become visually bundled and do not interfere. For the definition of similarity we use latent variable models which incorporate the user's assumption of what is important in the graph, and given the similarity construct the visualization with a suitable nonlinear projection method capable of maximizing the precision of the display. We finally show that the method outperforms alternative graph visualization methods empirically, and that at least in the special case of clustered data the method is able to properly abstract and visualize the links.

KEYWORDS: Graph clustering, graph visualization, latent variable model

CONTENTS

1 INTRODUCTION

The graph layout problem has been studied for decades, and still dozens of papers are written on it each year. The two traditional main algorithm families are so-called force-based methods and spectral methods. The force-based methods (e.g. [11]) rely on a spring analogy; each edge is assigned a spring which tries to keep the edges at a fixed length. When the spring contracts or extends, a force is applied at its endpoints. The energy function to be optimized balances the forces. Spectral methods, on the other hand, are generally based on computing the eigenvalues of a matrix related to the graph, such as the Laplacian. They have the advantage of being extremely fast and scaling to huge graphs. The disadvantage is that they often produce clearly worse layouts than the more computationally demanding force-based methods [3].

Recently, principled graph layout methods for specific tasks or assumptions have emerged. A method called LinLog has been shown to produce visualizations that reveal community structures [5]. In general, however, the principles underlying graph visualization are not clearly connected to visualization or graph properties.

We introduce a principle for visualization which says that nodes nearby on the display should have similar link distributions. Then their edges become naturally grouped and highlight structural features. We back this surprisingly simple principle by empirical comparisons and an analysis in the special case of clustered nodes. Similarity needs to be defined, and in the definition it is possible to incorporate our assumptions on what properties of the graphs are important and should be visualized well.

In latent variable modeling we generally build our assumptions about what is important into the modeling assumptions, such that the latent space captures what is important in the samples. Hence, given such a latent variable model, we compute the similarities using the model. After we have a similarity measure that we believe in, the remaining problem is to reduce the dimensionality into two such that close-by nodes on the display are similar. This can be done with a recent method NeRV [10] which allows controlling the tradeoff between precision and recall of the visualization. Visualizations maximizing precision would fulfill our principle that nearby nodes on the display should be similar in terms of their link distributions.

We will show that at least under simplifying assumptions the method successfully reveals the link structure of the graph, in particular when the data have cluster structure. We empirically verify the performance, and further show that our method outperforms alternative graph visualization methods.

2 MODEL-BASED GRAPH VISUALIZATION: LDA-NERV

We claim that if we visualize a graph such that nearby nodes in the visualization have similar link distributions, the visualization will be good. Next we introduce the steps needed in practice, then discuss why the visualization should be good, and in the next section show the performance experimentally.

2.1 Method

Generative models for graphs

We assume that the links have been generated from a latent variable model, where the latent variables capture what is central in the graph, and the rest is noise. We should, as usual in modeling, build our assumptions about the data into the model, and those assumptions will then determine what kinds of properties of the graphs will be visualized well.

A convenient, flexible choice is SSN-LDA (Simple Social Network Latent Dirichlet Allocation) [12], a generative topic-type model for graphs. In SSN-LDA each node is associated with a membership vector over a set of latent components. Each component is in turn associated with a distribution over the nodes in the graph. Edges are generated by first drawing a component for the starting node, and then drawing the receiving node from the component-specific distribution. The assumption behind this generative process is that the graph can be decomposed into overlapping latent components, that is, groups of nodes with similar edge distributions. Hence we assume that the components are more important for graph visualization than are details in link patterns.

For SSN-LDA the latent space takes the form of component probabilities given the node, and thus the distances between link distributions should be evaluated in terms of those probabilities. We will use the quickly computable Hellinger distance which has been proven useful for topic models earlier [1],

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2}, \tag{1}$$

where p and q are the probability distributions over the components. The distances could alternatively be computed in the link space, using information geometric formulations.

Nonlinear dimensionality reduction

The final step is to position the nodes on the display so that nearby nodes will have similar link distributions. In other words, we want to place similar nodes, *and only similar nodes*, close to each other on the display.

There are many nonlinear dimensionality reduction methods that operate on a distance or similarity matrix of data. As far as we know, however, there is only one which allows choosing between the two kinds of errors: placing dissimilar nodes close to each other, and placing similar nodes far apart. The method called NeRV (Neighbor Retrieval Visualizer) [10] formulates visualization as visual information retrieval, where the two kinds of errors translate to precision and recall of retrieving relevant (=similar) points based on the display. Having only similar points nearby equates high precision.

The tradeoff between precision and recall can be controlled by a single parameter, λ. Although $\lambda = 0.0$ maximizes precision, we perturb it slightly away from zero, to $\lambda = 0.1$, to regularize the solution.

2.2 Why should this work?

An obvious consequence of placing nodes with similar edge distributions near each other is that edges will tend to bundle; the density of a bundle will increase with the number of nodes with a similar distribution (more nodes in the same place, sending out edges in the same directions). This feature will draw attention to interesting structures in the latent space: nodes that are well described by a single component will form clusters with dense edge bundles, resulting in a nice-looking and interpretable visualization.

If the graph contains a community, that is, a set of nodes with many edges within the set and relatively few edges to the outside, the nodes in the community will be clustered together on the display. This is because the nodes in the community will have similar edge distributions (they mainly link to each other). The community will also most likely be clearly separated from other nodes in the layout because the other nodes will have different edge distributions. In other words, LDA-NeRV can be expected to reveal any community structure that a graph may contain.

More generally, if the nodes form clusters that are not communities, that is, groups of nodes that link to other groups in the same way, then they will be grouped together in the visualization for the same reason, and the edges between clusters will form clearly visible bundles. Our experiments show that this is indeed the case (see Figure. 2).

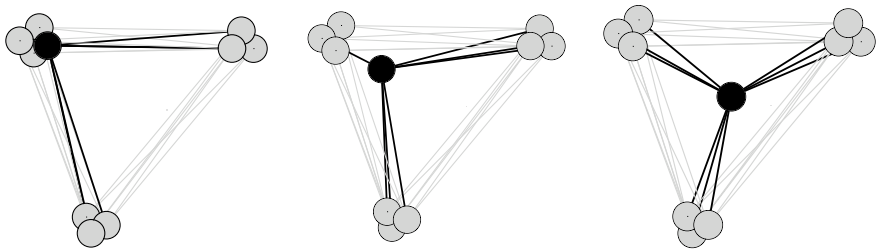

Figure 1: Interpolation between clusters. This artificial example shows three simple graphs, where 9 nodes (gray) form three clear clusters or components, each having links to the other two clusters. On the left, the black node belongs to one of the clusters based on its linking behavior. As the linking behavior changes towards right, it is best described as an additive mixture of the components, and the position of the node becomes correspondingly interpolated between the clusters.

In the ideal case of well-separable clusters described above our method intuitively works well. Next we discuss what happens if the cluster structure is less clear, which is often the case in real-world graphs.

In topic models the link distribution of each node is explained by a mixture of links from the components. Each node belongs to each component to a degree, and given that most nodes are clustered the degrees of a node determine interpolation between the clusters (Fig. 1). The deviation of a node from the clusters can be measured in terms of the entropy of the node's component distribution: the lower the entropy, the more clearly the node belongs to certain clusters.

As a practical remark, if there are very high-entropy nodes because their

membership to the components is very uncertain, they will have small Hellinger distances from each other. If we have a graph where some of the nodes have very low entropy, and the rest have very high entropy, the high entropy nodes will by symmetry have nearly the same, large Hellinger distance to any of the low entropy nodes. As a consequence, the high entropy nodes will tend to cluster in the center of the display, whereas the low entropy nodes form their clusters on the outskirts of the display (as in Fig. 1).

When visualizing graphs with a notable number of high-entropy nodes, as we will do in Figure 3, we can clarify the visualization by making the size of the nodes inversely proportional to the entropy. This makes sense since for high-entropy nodes the component membership profile is very uninformative and hence the locations are as well.

3 EXPERIMENTS

We compare LDA-NeRV with two graph layout methods (see below) on three very different graphs.

The Football graph consists of 115 nodes and 613 edges. Each node represents a team, and an edge between two teams implies that the teams played each other during one season [2]. Each team is known to belong to one of 12 conferences. The teams in each conference played heavily against each other, and there is some structure in games between the conferences which is not as obvious.

In the Cora graph, each node is a scientific publication and an edge between two nodes indicates that one of the papers cites the other [7]. There are 2485 nodes and 5068 edges in the graph, and the papers belong to seven predefined classes, such as Machine Learning or Artifical Intelligence. Although there is definitely structure in this data, it is much more varied and complicated than the communities of the football graph.

In the Adjective–noun graph [4] each of the 112 nodes represents a word, and nodes are connected by an edge (424 in total) if the words have appeared next to each other in running text. There are two kinds of words: adjectives and nouns. As would be expected, a noun appearing next to a noun and an adjective appearing next to an adjective are much rarer occurrences than a noun appearing next to an adjective, so the graph is almost bipartite. We chose this dataset as an example of data that has definite structure but no communities.

3.1 Comparison methods

We compare LDA-NeRV against two representative graph layout methods. The first algorithm is from the dominant method family, force-based methods, a variant of Walshaw's multilevel force-based algorithm [11] implemented as a Cytoscape [8] plugin [6]. Like any force-based algorithm, Walshaw's algorithm treats the edges as springs with uniform natural lengths. When a spring extends or contracts beyond its natural length, it applies a corresponding attractive or repulsive force to its endnodes. The graph layout is produced iteratively by giving the vertices initial positions and letting the system find

an equilibrium. Walshaw's algorithm greatly speeds up the convergence by creating a good initial guess for the final layout based on layouts for coarser approximations of the actual graph.

The second method is a recent principled algoritm for revealing and visualizing community structure in graphs, Andreas Noack's edge-repulsion LinLog [5]. The layouts are produced by minimizing an energy function, and it has been shown that minimizing this energy function also minimizes the ratio of the mean distance between connected nodes to the mean distance between all nodes. This suggests that LinLog layouts should indeed reveal communities.

3.2 The football graph

The layouts of the football data with the three methods, displayed in Figure 2, show that LinLog and LDA-NeRV give very similar results and both find the structure as expected. LDA turned out to be very robust against the choice of the number of components. The figure shows results for 24 components, double the number of conferences, suggesting that simply choosing a large number is a feasible strategy; alternatively of course standard model complexity criteria could be applied, or Dirichlet Processes. Walshaw's algorithm has also placed the conferences in almost disjoint areas of the display, but the structure in the linking between them is completely hidden.

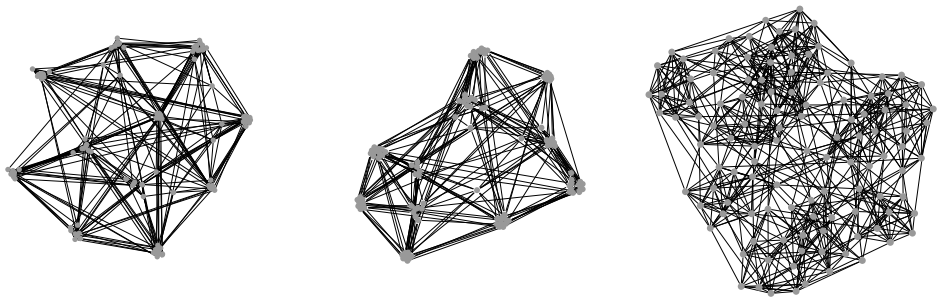

Figure 2: Football data set visualized with LDA-NeRV (left), LinLog (center) and Walshaw's algorithm (right).

3.3 The Cora graph

In layouts of both LDA-NeRV and LinLog (Figure 3), the content classes of the documents are clearly visible. Walshaw resulted in a large cloud without apparent structure (not shown). What is striking in the LinLog visualization is that it places a significant proportion of the nodes in very tight clusters, which are not really visible in the image since the nodes are so overlapping. As a result the visualization seems to have far fewer nodes. In the LDA-NeRV visualization there is a large bundle of nodes in the center of display; the nodes have a high cluster membership entropy and their salience has been decreased by making the size of the nodes proportional to the entropy of cluster membership as discussed in Section 2.2.

Here the resolution of the LDA-NeRV result can be selected by selecting the number of components; the quantitative quality of the results has turned out to be robust to the number, given that it is not very low of course. This visualization has 70 components; with 35 components the quantitative measures (introduced below) were still clearly better than with LinLog.

3.4 The Adjective–noun graph

On the Adjective–noun graph (Fig. 4) both Walshaw and LinLog failed miserably to find structure, whereas LDA-NeRV separated adjectives and nouns, and visualizes the predominantly bipartite structure. It is not surprising that LinLog fails, since the data does not contain communities, mutually connected clusters.

We further tested what would happen if the LDA in LDA-NeRV was replaced by another generative model, a model that assumes community structure as LinLog does. The Interaction Component Model (ICM) [9] a topic model and very similar to LDA, the main difference being that it assumes communities. The resulting ICM-NeRV visualizer fails as badly as LinLog (Fig. 4), emphasizing the importance of correct modeling assumptions.

3.5 Quantitative measures

In summary, it is clear from the visualizations that LDA-NeRV is able to visualize linking between clusters and components. LinLog does that as well given that the links are between clear and strong communities, as in the Football graph of Figure 2. In the Cora graph there apparently is no such connection; LinLog only finds communities, and cannot find link bundles (Fig. 3). Walshaw, on the other hand, does not assume communities but does not emphasize bundles either, which is visible in all images.

Next we will quantify this visual finding. We will not use the traditional measure of graph layout quality, number of edge crossings, because we claim that it is a very misleading figure. For instance, in the graph layouts of the Football data set (Fig. 2) there are lots of edge crossings but since the edges have become bundled, the crossings hardly disturb the interpretation of the graph at all.

Instead, we measure the compromise the display makes in visualizations, but computing precision and recall. The NeRV paper [10] recommends smoothed precision and smoothed recall, which are more sensitive variants of standard precision and recall. The measures may bias the result towards NeRV, but since they are the best available measures for our task, we simply use them to quantify the visual impression.

The measures for each method and graph, collected in Table 1, verify that the visual impression of visibility of the bundles seems to hold quantitatively, since LDA-NeRV has the highest precision and recall.

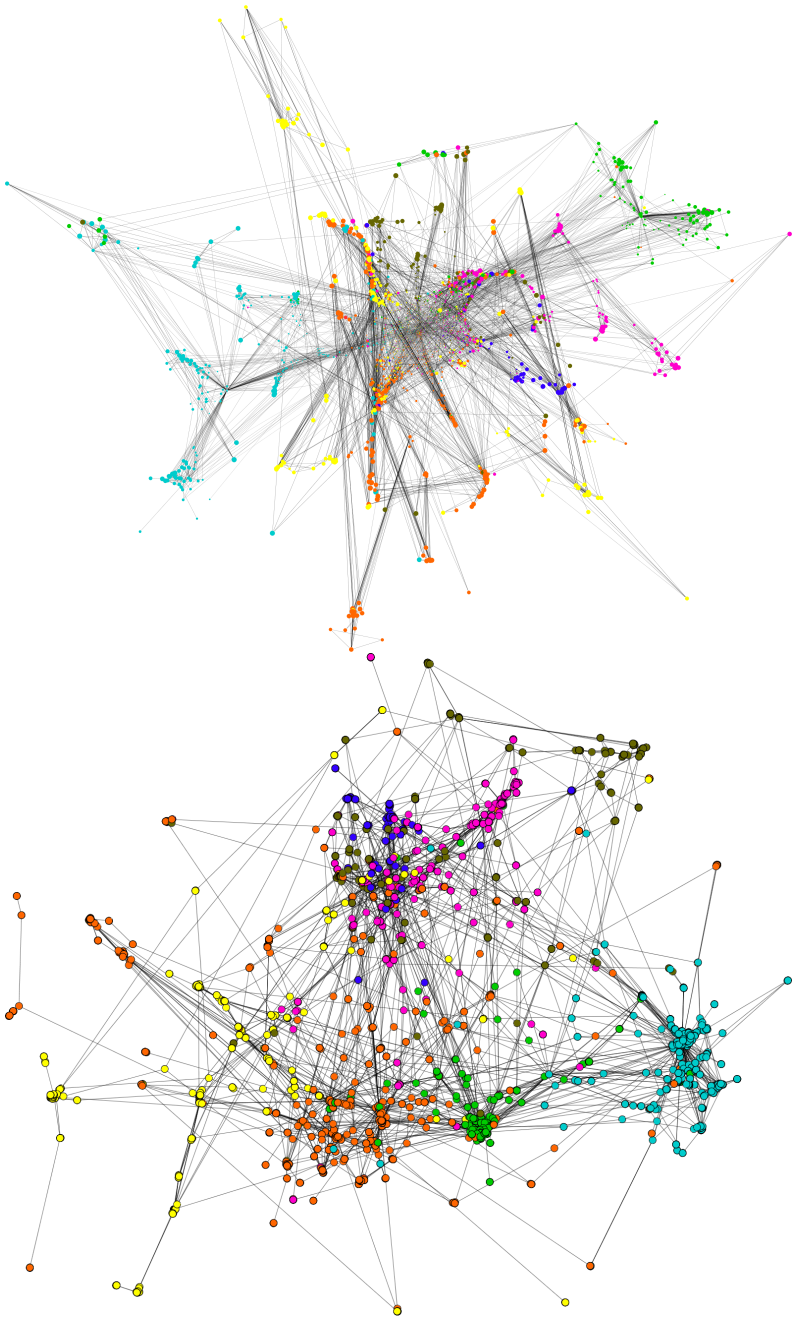

Figure 3: Cora data set visualized with LDA-NeRV (top) and LinLog (bottom). Colors indicate topic classes of the documents.

	SmP			SmR		
	Football	Cora	adjnoun	Football	Cora	adjnoun
Walshaw	370	140,000	1300	580	180,000	1000
LinLog	370	140,000	1300	600	180,000	1100
LDA-NeRV	**61**	**23,000**	**170**	**77**	**34,000**	**80**

Table 1: Quantitative quality of the different layouts. The best result for each graph–measure-pair has been bolded. SmP: smoothed precision; SmR: smoothed recall.

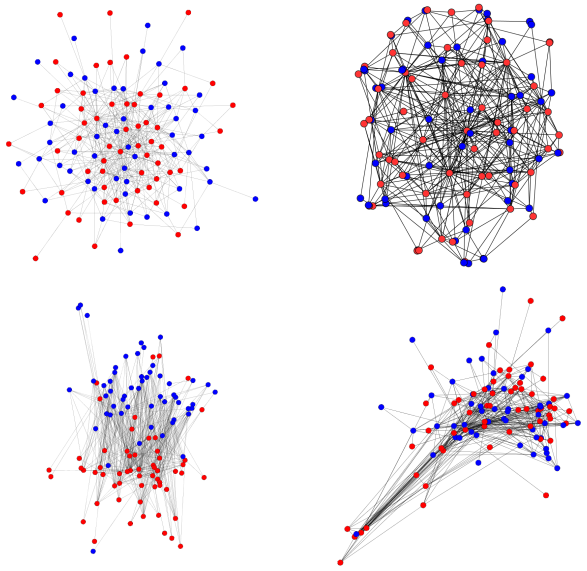

Figure 4: The Adjective–noun dataset visualized with Walshaw's algorithm (top left), LinLog (top right), LDA-NeRV (bottom left) and ICM-NeRV (bottom right). Blue: adjectives, red: nouns.

4 DISCUSSION

We have introduced a new simple principle for graph visualization, and shown how it can be taken into use in a method which outperforms existing graph visualization algorithms, in the hard task of visualizing large (and small) graphs. What is particularly attractive is that the method is model-based. A generative model of the graph is assumed, and the visualization focuses on those properties in the link distribution the generative model models well, that is, considers important. The visualization can be made to focus on different properties of the graph by changing the model. In effect, the principle turns graph visualization, for which only mostly heuristic solutions have existed so far, into a generative modeling problem.

The obvious disadvantage is longer running time, but the computation of the graphs in this paper only took some tens of minutes on a standard PC for all of the algorithms.

REFERENCES

[1] David Blei and John Lafferty. A correlated topic model of Science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[2] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA*, 99(12):7821–7826, 2002.

[3] Stefan Hachul and Michael Juenger. Large-graph layout algorithms at work: An experimental study. *Journal of Graph Algorithms and Applications*, 11(2):345–369, 2007.

[4] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[5] Andreas Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007.

[6] P. Salmela, O. S. Nevalainen, and T. Aittokallio. A multilevel graph layout algorithm for cytoscape bioinformatics software platform. Technical Report 861, Turku Centre for Computer Science, 2008.

[7] Prithviraj Sen and Lise Getoor. Link-based classification. Technical Report CS-TR-4858, University of Maryland, College Park, USA, 2007.

[8] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, November 2003.

[9] Janne Sinkkonen, Janne Aukia, and Samuel Kaski. Inferring vertex properties from topology in large networks. In *Working Notes of the 5th International Workshop on Mining and Learning with Graphs (MLG'07)*, Florence, Italy, 2007. Universita degli Studi di Firenze.

[10] Jarkko Venna and Samuel Kaski. Nonlinear dimensionality reduction as information retrieval. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS*07), San Juan, Puerto Rico, March 21-24*, 2007.

[11] Chris Walshaw. A multilevel algorithm for force-directed graph drawing. *Journal of Graph Algorithms and Applications*, 7(3):253–285, 2003.

[12] Haizheng Zhang, Baojun Qiu, C. Lee Giles, Henry C. Foley, and John Yen. An LDA-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics (ISI) 2007*, pages 200–207. IEEE, 2007.

TKK REPORTS IN INFORMATION AND COMPUTER SCIENCE

TKK-ICS-R10 He Zhang, Markus Koskela, Jorma Laaksonen
Report on forms of enriched relevance feedback. November 2008.

TKK-ICS-R11 Ville Viitaniemi, Jorma Laaksonen
Evaluation of pointer click relevance feedback in PicSOM. November 2008.

TKK-ICS-R12 Markus Koskela, Jorma Laaksonen
Specification of information interfaces in PinView. November 2008.

TKK-ICS-R13 Jorma Laaksonen
Definition of enriched relevance feedback in PicSOM. November 2008.

TKK-ICS-R14 Jori Dubrovin
Checking Bounded Reachability in Asynchronous Systems by Symbolic Event Tracing. April 2009.

TKK-ICS-R15 Eerika Savia, Kai Puolamäki, Samuel Kaski
On Two-Way Grouping by One-Way Topic Models. May 2009.

TKK-ICS-R16 Antti E. J. Hyvärinen
Approaches to Grid-Based SAT Solving. June 2009.

TKK-ICS-R17 Tuomas Launiainen
Model checking PSL safety properties. August 2009.

TKK-ICS-R18 Roland Kindermann
Testing a Java Card applet using the LIME Interface Test Bench: A case study. September 2009.

TKK-ICS-R19 Kalle J. Palomäki, Ulpu Remes, Mikko Kurimo (Eds.)
Studies on Noise Robust Automatic Speech Recognition. September 2009.

ISBN 978-952-248-095-8 (Print)
ISBN 978-952-248-095-5 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)