ECML PKDD 2011

EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
PRINCIPLES AND PRACTICE OF KNOWLEDGE DISCOVERY IN DATABASES

The Second International Workshop on Mining Ubiquitous and Social Environments

MUSE'11

September 5, 2011

Athens, Greece

Editors:
Martin Atzmueller
Knowledge and Data Engineering Group, University of Kassel
Andreas Hotho
Data Mining and Information Retrieval Group, University of Wuerzburg

Table of Contents

Preface

The 2^{nd} International Workshop on Mining Ubiquitous and Social Environments (MUSE 2011) was held in Athens, Greece, on September 5^{th} 2011 in conjunction with the The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2011).

Mining ubiquitous and social environments is an emerging area of research, with the focus on advanced systems for data mining in distributed and network-organized systems. The environments usually consist of small, heterogeneous, and distributed devices, potentially integrated with the social semantic web. Such contexts create a wide range of new interactions and challenges. Concerning these, the characteristics of ubiquitous and social mining are in general rather different from common mainstream machine learning and data mining approaches. The data emerges from potentially hundreds to millions of different sources, in a collective way. Since these sources are usually not coordinated, they can overlap or diverge in any possible way. Therefore, the analysis of the collected data, the adaptation of existing data mining and machine learning approaches, and the engineering and development of novel algorithms are challenging and exciting steps into this new arena. The goal of this workshop is to promote an interdisciplinary forum for researchers working in the fields of ubiquitous computing, social semantic web, Web 2.0, and social networks which are interested in utilizing data mining in an ubiquitous setting. The workshop focuses on contributions applying state-of-the-art mining algorithms on ubiquitous and social data. Papers considering the intersection of the two fields are especially welcome.

This proceedings volume comprises the papers of the MUSE 2011 workshop. In total, we received nine submissions, from which we were able to accept six submissions, three full papers and three short papers, based on a rigorous reviewing process. Additionally, the scientific program also featured an invited talk on the convergence of social and ubiquitous data by Kristian Kersting (Fraunhofer IAIS & University of Bonn, Germany).

Based on the set of accepted papers, and the invited talk, we set up three sessions. The first session discusses the foundations of spatial, sensor, and streaming data. The work *Dealing with Collinearity in Learning Regression Models from Geographically Distributed Data* by Annalisa Appice, Michelangelo Ceci, Donato Malerba and Antonietta Lanza discusses the usage of learning regression models in the context of ubiquitous and distributed data. Next, the paper *Spatio-Temporal Reconstruction of Un-Sampled Data in a Sensor Network* by Pietro Guccione, Anna Ciampi, Annalisa Appice, Donato Malerba and Angelo Muolo describes a resource-aware approach for handling spatial-temporal sensor data. Finally, *A Data Generator for Multi-Stream Data*, Zaigham Faraz Siddiqui, Myra Spiliopoulou, Panagiotis Symeonidis and Eleftherios Tiakas present a data generator for interrelated streaming data, especially concerning temporal data.

The second session is concerned with ubiquitous and social applications, bridging the two general topics of the workshop. Thus, the session directly leads to the topic of the invited talk. In *The Generation of User Interest Profiles from Semantic Quiz Games*, Magnus Knuth, Nadine Ludwig, Lina Wolf and Harald Sack provide a unified approach to generate user interest profiles without direct user interaction out of data from quiz games. After that, the paper *Face-to-Face Contacts during a Conference - Communities, Roles, and Key Players* presents an analysis of social and ubiquitous data during a conference, and features community mining and role characterization in a conference scenario. Concluding the session, the work *Identifying Dense Structures to Guide the Detection of Misuse and Fraud in Network Data*, Ursula Redmond, Martin Harrigan and Padraig Cunningham describe techniques for suspicious structures in network data, specicially those indicating misuse or fraud.

The third session features the invited talk *Perception and Prediction Beyond the Here and Now* by Kristian Kersting, in which he presents techniques for handling large data with applications to ubiquitous and social environments.

We would like to thank the invited speaker, all the authors who submitted papers and all the workshop participants. We are also grateful to members of the program committee members and external referees for their thorough work in reviewing submitted contributions with expertise and patience. A special thank is due to both the ECML PKDD Workshop Chairs and the members of ECML PKDD Organizing Committee who made this event possible. Special thanks go to Steffen Staab for his help in organizing an independent review process of a selected publication.

Athens, September 2011

Martin Atzmueller
Andreas Hotho

Workshop Organization

Workshop Chairs

Martin Atzmueller	University of Kassel - Germany
Andreas Hotho	University of Würzburg - Germany

Program Committee

Harith Alani	KMi, Open University - United Kingdom
Ulf Brefeld	Yahoo Research - Spain
Ciro Cattuto	ISI Foundation - Italy
Michelangelo Ceci	University of Bari - Italy
Marco Degemmis	University of Bari - Italy
Fabien Gandon	INRIA - France
Claudia Müller-Birn	Carnegie Mellon University - USA
Ion Muslea	SRI International - Menlo Park, USA
Alexandre Passant	DERI - Ireland
Giovanni Semeraro	University of Bari - Italy
Sergej Sizov	University of Koblenz-Landau - Germany
Steffen Staab	University of Koblenz-Landau - Germany
Myra Spiliopoulou	University of Magdeburg - Germany
Gerd Stumme	University of Kassel - Germany
Maarten van Someren	University of Amsterdam - The Netherlands

Dealing with Collinearity in Learning Regression Models from Geographically Distributed Data

Annalisa Appice, Michelangelo Ceci, Donato Malerba, and Antonietta Lanza

Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
via Orabona, 4 - 70126 Bari - Italy
{appice, ceci, malerba, lanza}@di.uniba.it

Abstract. Despite the growing ubiquity of sensor deployments and the advances in sensor data analysis technology, relatively little attention has been paid to the spatial non-stationarity of sensed data which is an intrinsic property of the geographically distributed data. In this paper we deal with non-stationarity of geographically distributed data for the task of regression. At this purpose, we extend the Geographically Weighted Regression (GWR) method which permits the exploration of the geographical differences in the linear effect of one or more predictor variables upon a response variable. The parameters of this linear regression model are locally determined for every point of the space by processing a sample of weighted neighboring observations. Although the use of locally linear regression has proved appealing in the area of sensor data analysis, it also poses some problems. The parameters of the surface are locally estimated for every space point, but the form of the GWR regression surface is globally defined over the whole sample space. Moreover, the GWR estimation is founded on the assumption that all predictor variables are equally relevant in the regression surface, without dealing with spatially localized phenomena of collinearity. Our proposal overcomes these limitations with a novel tree-based approach which is adapted to the aim of recovering the functional form of a regression model only at the local level. A stepwise approach is then employed to determine the local form of each regression model by selecting only the most promising predictors and providing a mechanism to estimate parameters of these predictors at every point of the local area. Experiments with several geographically distributed datasets confirm that the tree based construction of GWR models improves both the local estimation of parameters of GWR and the global estimation of parameters performed by classical model trees.

1 Introduction

Underpinning geographic thinking is the assumption of spatial non-stationarity, according to which a phenomenon will vary across a landscape. In a regression task, where the predictor variables and the response variable are collected at several locations across the landscape, the major consequence of spatial non-stationarity is that the measurement of the relationship of the predictor variables upon the response variable may vary from location to location. The consequence

of this spatial variability is that of discouraging a spatial analyst from employing any conventional regression-based model which largely assumes the independence of observations from the spatial location and ignores the spatial non-stationarity much to the detriment of spatially varying relationships. In support of this consideration, LeSage and Pace [16] have shown how the application of conventional regression models leads to wrong conclusions in spatial analysis and generates spatially autocorrelated residuals. One of the best known approaches to spatial regression is GWR (Geographical Weighted Regression) [2], a spatial statistics technique which addresses the challenges of the spatial non-stationarity par excellence. In particular, GWR maps a local model as opposed to the global linear model conventionally defined in statistics, in order to fit the relationship between predictor variables and a response variable. In fact, unlike the conventional regression equation which defines single parameter estimates, GWR generates a *parametric* linear equation, where parameter estimates vary from location to location across the landscape. Each set of parameters is estimated on the basis of distance-weighted neighboring observations. The choice of a neighborhood is influenced by the observation that the positive spatial autocorrelation of a variable arises in almost all spatial data applications [11]. In particular, the positive spatial autocorrelation of the response variable is that the property of response takes values at pairs of locations a certain distance apart (neighborhood) to be more similar than expected for randomly associated pairs of observations [15].

The main motivation for focusing our attention on GWR is that a number of recent publications has demonstrated that this kind of local spatial model is appealing in areas of spatial econometrics, including climatology [5], urban poverty [1, 18], social segregation [17], industrialisation [12], environmental and urban planning [23] and so on. Although it covers a broad spectrum of applications, GWR still gives rise to research challenges. In the following, we analyze the major issues of GWR and define novel local algorithms which definitely take a step forward in the research on regression in spatial data analysis.

The principal issue we face here is that GWR outputs a single parametric equation which represents the linear combination of all the predictor variables in the task and considers the coefficients of the linear combination as parameters for the local estimation. This means that GWR assumes that predictor variables are all equally relevant for the response everywhere across the landscape, although it admits spatially varying parameters. Consequently GWR does not deal with the spatially localized phenomenon of *collinearity*. In general, collinearity is a statistical phenomenon in which two or more predictor variables in a multiple regression model are highly correlated. In this case, the coefficient estimates may change erratically in response to small changes in the model or the data. In conventional regression, the problem of collinearity is addressed by identifying the subset of the relevant predictor variables and outputting the linear combination of only the variables in this subset [9]. Based on this idea, we argue that a solution to collinearity in GWR is to determine a parametric regression surface, which linearly combines a subset of the predictor variables. As we expect that variables in the subset may vary in space, we define a new spatially lo-

cal regression algorithm, called GeWET (*GE*ographically *W*eighted r*E*gression *T*rees learner), which integrates a spatially local regression model learner with a tree-based learner. The tree-based learner recursively segments the landscape along the spatial dimensions (e.g. latitude and longitude), according to a measure of the positive spatial autocorrelation over the response values. In practice, the leaves of an induced tree represent a segmentation of the landscape into non-overlapping areal units which spatially reference response values positively autocorrelated within the corresponding leaf. Owing to the high concentration of positive spatial autocorrelated response values within a leaf, we assume that the search for a parametric surface equation associated to the leaf is plausible. A leaf surface reasonably combines only a subset of relevant predictor variables, although the parameters of this surface will be locally estimated across the leaf. In particular, at each leaf, the predictor variables and the local parameters are learned by adapting the forward stepwise approach [9], defined for a global aspatial models toward a local spatial model.

Therefore, the innovative contributions of this work with respect to original formulation of GWR are highlighted as follows:

1. We propose a tree-based learner which permits to segment the landscape in non-overlapping areal units that group response values which are positively autocorrelated;
2. We do not assume any global form of the geographically weighted regression model, but we allow the subset of predictive variables in the model to vary across landscape;
3. We design a stepwise technique to determine a geographically weighted regression model. The choice of the predictive variables for the model is carried out by an automatic procedure which permits to select only the most promising variables and to locally estimate the corresponding parameters;
4. We empirically prove that the proposed method is more accurate than the traditional GWR and the state-of-art aspatial model tree learner M5' [25] in predicting unknown ubiquitous response values spread across the landscape.

The paper is organized as follows. In the next Section we revise related work on regression in spatial statistics and spatial data mining. In Section 3, we illustrate the problem setting and introduce some preliminary concepts. In Section 4, we present our tree-based spatially local regression algorithm. In Section 5 we describe experiments we have performed with several benchmark spatial data collections. Finally, we draw some conclusions and outline some future work.

2 Background and Related Work

Several definitions of the regression task have been formulated in spatial data analysis over the years. The formulation we consider in this work is the traditional one, where a set of attribute-value observations for the predictor variables and the response variable are referenced at point locations across the landscape. So far, several techniques have been defined to perform this task, both in spatial

statistic and spatial data mining. A brief survey of these techniques (e.g. k-NN, geographically weighted regression, kriging) is reported in [22].

In particular, the k-Nearest Neighbor (k-NN) algorithm [20] is a machine learning technique which appears to be a natural choice for dealing with the regression task in spatial domains. Each test observation is labeled with the (weighted) mean of the response values of neighboring observations in the training set. A distance measure is computed to determine neighbors. As spatial coordinates can be used to determine the Euclidean distance between two positions, k-NN predicts the response value at one position by taking into account the observations which fall in the neighborhood. Thus, k-NN takes into account a form of positive autocorrelation over the response attribute only.

GWR [2] is a spatial statistic technique which extends the regression framework defined in conventional statistics by rewriting a globally defined model as a locally estimated model. The global regression model is a linear combination of predictor variables, defined as: $y = \alpha + \sum_{k=1}^{n} \beta_k x_k + \epsilon$ with intercept α and parameters β_k globally estimated for the entire landscape, by means of the least square regression method [9]. Then GWR rewrites this equation in terms of a *parametric* linear combination of predictor variables, where the parametric coefficients (intercept and parameters) are locally estimated at each location across the landscape. Formally, the parametric model at location i is in the form:

$$y(u_i, v_i) = \alpha(u_i, v_i) + \sum_{k=1}^{n} \beta_k(u_i, v_i) x_k(u_i, v_i) + \epsilon_i, \tag{1}$$

where (u_i, v_i) represents the coordinate location of i, $\alpha(u_i, v_i)$ is the intercept at location i, $\beta_k(u_i, v_i)$ is the parameter estimate at the location i for the predictor variable x_k, $x_k(u_i, v_i)$ is the value of the k-th variable for location i, and ϵ_i is the error term. Intercept and parameter estimates are based on the assumption that observations near one another have a greater influence on each other. The weight assigned to each observation is computed on the basis of a distance decay function centered on the observation i. This decay function is modified by a bandwidth setting, that is, at which distance the weight rapidly approaches zero. The bandwidth is chosen by minimizing the Akaike Information Criteria (AIC) score [6]. The choice of the weighting scheme is a relevant step in the GWR procedure. Several different weighting functions are defined in the literature [2], the more common kernels being the Gaussian and the bi-square weighting functions.

Kriging [4] is a spatial statistic technique which exploits positive autocorrelation and determines a local model of the spatial phenomenon. It applies an optimal linear interpolation method to estimate unknown response values $y(u_i, v_i)$ at each location i across the landscape. $y(u_i, v_i)$ is decomposed into a structural component, which represents a mean or constant trend, a random but spatially correlated component and a random noise, which expresses measurement errors or variations inherent to the attribute of interest.

A completely different approach is reported in [19], where the authors propose a spatial data mining technique, called Mrs-SMOTI, which adapts the model tree induction to spatial data. Mrs-SMOTI performs a tree-based segmentation

of the training data by considering Boolean tests on the predictor variables and associates each leaf of the tree with a global regression model built stepwise by considering only a subset of the predictor variables. The intriguing aspect of this proposal is that it allows the induction of a model tree which predicts the response value of a spatial object, by considering predictor variables of objects belonging to possibly different layers. Indeed, the authors propose using Mrs-SMOTI to predict the mortality rate of a district (response variable), by considering as predictor variables both the number of inhabitants of the district (same layer of the response variable) and the pollution rate of one or more rivers (new layer) crossing the district, as well as the traffic rate of one or more roads (new layer) crossing the district. The regression problem considered in this paper is different from task faced by Mrs-SMOTI, as we assume a single layer. In any case, the idea of partitioning data and learning a model in a stepwise fashion at each partition is appealing to solve the problem of linear collinearity also in spatial domains. We take inspiration from this idea, and significantly extend it. In fact, we propose to segment landscape in areal units according to Boolean tests on spatial dimensions and not on predictor variables as traditional model trees do. The segmentation is opportunely tailored to identify boundaries of areal units across the landscape which group (positively) autocorrelated response values. Finally, the regression model associated to each leaf is built stepwise and it is synthesized to be a locally estimated regression model.

3 Problem Setting and Preliminary Concepts

In this Swction, we define the regression relationship which may arise among the predictor variables and a response variable observed in a geographically distributed environment. This relationship is defined on the basis of a field-based [24] model of the variables which permits to fit the ubiquity of data across the landscape. The inductive regression task is then formulated to learn a definition of the regression relationship from a geographically distributed training sample. We propose to face this task by learning a piecewise definition of a space-varying (parametric) regression function which met the requirements of spatial non-stationarity posed by this task without suffering of collinearity problems.

Formally, the *regression relationship* in a geographically distributed environment is denoted as $\tau(Y, X_1, X_2, \ldots, X_m, U, V)$ and defines the (possibly unknown) space-varying relationship between a response numeric variable Y and m predictor numeric variables X_j (with $j = 1, \ldots, m$). This relationship reasonably varies across the landscape $U \times V$ (e.g. Latitude $\times$ Longitude) due to the phenomenon of spatial non-stationarity. In this formulation and according to the *field-based model*, the variation of both the response variable and the predictor variables across the landscape is mathematically defined by means of one response function $y(\cdot, \cdot)$ and m distinct predictor functions $x_j(\cdot, \cdot)$ which are respectively:

$$y : \mathbb{U} \times \mathbb{V} \mapsto \mathbb{Y}, \tag{2}$$

$$x_j : \mathbb{U} \times \mathbb{V} \mapsto \mathbb{X}_j \ (with\ j = 1, \ldots, m), \tag{3}$$

where $\mathbb{U} \times \mathbb{V} \subseteq \mathbb{R} \times \mathbb{R}$ is the range of the Cartesian product $U \times V$; $\mathbb{Y}$ is the numeric range of response function $y(\cdot, \cdot)$ (variable Y); $\mathbb{X}_j$ is the numeric range of the predictor function $x_j(\cdot, \cdot)$ (variable X_j).

An extensional definition D of the relationship τ comprises any set of observations which is, simultaneously collected across the landscape, according to both the response function ($y(\cdot, \cdot)$) and the predictor functions ($x_j(\cdot, \cdot)$ with $j = 1, \ldots, m$). The observation i of this set is the data tuple defined as follows:

$$[i,\ u_i, v_i,\ x_1(u_i, v_i),\ x_2(u_i, v_i),\ \ldots,\ x_m(u_i, v_i),\ y(u_i, v_i)], \tag{4}$$

where i is the primary key of the data tuple one-to-one associated to the point location with coordinates (u_i, v_i). $x_j(u_i, v_i)$ is the value measured for the predictor variable X_j at the location (u_i, v_i) across the landscape, while $y(u_i, v_i)$ is the (possibly unknown) value measured for the response variable Y at (u_i, v_i). The response value $y(u_j, v_j)$ may be unknown, in this case the tuple i is unlabeled.

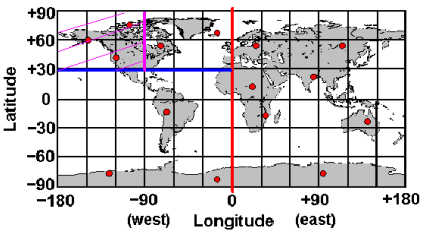
The *inductive regression task* associated to τ can be formulated as follows. Given a training data set $T \subset D$ which consists of a sample of n randomly tuples taken from D and labeled with the known values for the response variable. The goal is to learn a space-varying functional representation $f\colon \mathbb{U} \times \mathbb{V} \mapsto \mathbb{R}$ of the relationship τ such that f can be used to predict unlabeled responses at any location across the landscape. Our proposal to address this task consists of a new learner which receives training data T as input and outputs a *piecewise* definition for the space-varying function f which is defined as a geographically weighted regression tree. This tree recursively partitions the landscape surface along the spatial dimensions U and V and associates the areal unit at each leaf with a parametric (space-varying) linear combination of an opportunely chosen subset of the predictor variables. The parameters of this equation are locally estimated at each training location which fall in the leaf.

Formally a *geographically weighted regression tree* f is defined as a binary tree $f = (N, E)$ where:

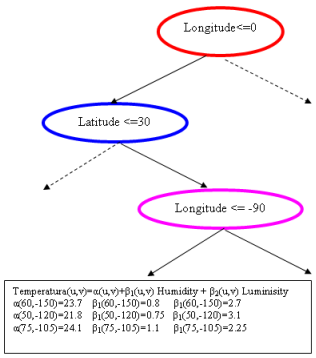
1. each node $n \in N$ is either an internal node or a leaf node ($N = N_I \cup N_L$)
   (a) an internal node $n \in N_I$ identifies a rectangular surface $s(n)$ over the landscape $U \times V$. The root identifies the entire landscape $U \times V$;
   (b) a leaf node $n \in N_L$ is associated with a parametric multiple linear regression function, that, for each location i falling in $s(n)$, allows the prediction of y_i according to predictor values and coefficients of the linear combination as they are locally estimated at the location i;
2. each edge $(n_i, n_j) \in E$ is a splitting edge labeled with a Boolean test over U or V which allows the identification of the rectangular surface $s(n_j) \subset s(n_i)$.

An example of a geographically weighted regression tree is reported in Figure 1.

Once the geographically weighted regression tree f is learned, it can be used to predict the response for any unlabeled observation $i' \in D$. During classification, the leaf of f which spatially contains i' is identified. The parametric function associated to this leaf is then applied to predict the unknown response value of i' by taking into account the $(u_{i'}, v_{i'})$ localization of i'.



(a) Sensors possibly installed across the Earth to obtain geographically distributed training observations of the Temperature (response), Humidity and Luminosity (predictors).



(b) A geographically weighted regression tree.

Fig. 1. An example of geographically weighted regression trees with spatial splits and space-varying parametric functions at the leaves (b) learned from spatially distributed training data (a).

4 The method GeWET

Based on the classical Top-Down Induction of Decision Trees (TDIDT) framework, GeWET recursively partitions the landscape in non-overlapping areal units and finds a parametric piecewise prediction model that fits the areal units. Details of both the partitioning phase and the regression model construction phase are discussed in this section. We also explain how geographically weighted regression trees induced by GeWet can be used to predict any unknown response values across the landscape.

4.1 Splitting phase

The partitioning phase is only based on the spatial dimensions of the data. The choice of the best split is based on the well known Global Moran autocorrelation measure [15], computed on the response variable Y and defined as follows:

$$I = \frac{N}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}} \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\sum_{i=1}^{N}(y_i - \overline{y})^2}, \quad (5)$$

where y_i is the value of the response variable at the observation i, $\overline{y}$ is the mean of the response variable, w_{ij} is the spatial distance based weight between observations i and j and N is the number of observations.

The Gaussian kernel is an obvious choice to compute the weights:

$$w_{ij} = \begin{cases} e^{(-0.5 d_{ij}^2 / h^2)} & \text{if } d_{ij} \leq h \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where h is the bandwidth and d_{ij} is the Euclidean spatial distance between observations i and j. The basic idea is that observations that show high positive spatial autocorrelation in the response variable should be kept in the same areal unit. Therefore, for a candidate node t, the following measure is computed:

$$I_t = \frac{(I_L N_L + I_R N_R)}{N_L + N_R} \tag{7}$$

where I_L (I_R) represents the Global Moran autocorrelation measure computed on the left (right) child of t and N_L (N_R) represents the number of training observations falling in the left (right) child of t. The higher I_t, the better the split.

The candidate splits are in the form $u_i \leq \gamma_u$ or $v_i \leq \gamma_v$, where (u_i, v_i) represents the spatial position of observation i. Candidate γ_u and γ_v values are determined by finding $n_{bins} - 1$ candidate equal-frequency cut points for each spatial dimension.

Our motivation of using autocorrelation measure as a splitting heuristics is that we are interested in looking for a segmentation of the landscape in regions of highly correlated data. Highly autocorrelated data pave the way for computing accurate GWR regression models [2].

The stopping criterion to label a node as a leaf requires the number of training observations in each node to be less than a minimum threshold. This threshold is set to the square root of the total number of training observations, which is considered a good locality threshold that does not allow too much loss in accuracy ever for rule-based classifiers [10].

4.2 Model Construction Phase

When the stopping criterion is satisfied, a leaf is built and a parametric linear regression function is associated to the areal unit associated to the leaf. Parameters of this function re estimated at each training location falling in such areal unit. After the tree is completely built, it defines a piecewice regression function f in this form:

$$f(u, v) = \sum_{i=1}^{l} I((u, v) \in D_i) \times f_i(u, v), \tag{8}$$

where:

– l is the number of leaves and $D_1, D_2, \ldots, D_l$ represent the segmentation of the landscape due to the spatial partition defined by the tree;
– $f_i(\cdot,\cdot)$ is the parametric linear function learned for the areal unit D_i;
– $I(\cdot)$ is an indicator function returning 1 if its argument is true and 0 otherwise.

The parametric linear regression function associated to each leaf is a parametric linear combination of a subset of predictor variables. The variables are selected according to a forward selection strategy. This way, the function is built stepwise. This process starts with no variable in the function and tries out the variables one by one, including the best variable if it is "statistically significant". For each variable included in the model, parameters of the output combination are locally estimated across the landscape covered by the leaf areal unit.

To explain the stepwise construction of a parametric regression function we illustrate an example. Let us consider the case we build a function of the response variable Y with two predictor variables X_1 and X_2 and estimate the space-varying parameters of this function at the location (u_i, v_i). Our proposal is to equivalently build the parametric function:

$$\hat{y}(u_i v_i) = \alpha(u_i, v_i) + \beta(u_i, v_i)\ x_1(u_i, v_i) + \gamma(u_i, v_i)\ x_2(u_i, v_i), \tag{9}$$

through a sequence of parametric straight-line regressions. At this aim, we start by regressing Y on X_1 and building the parametric straight line

$$\hat{y}(u_i, v_i) = \alpha_1(u_i, v_i) + \beta_1(u_i, v_i)\ x_1(u_i, v_i). \tag{10}$$

This equation does not predict Y exactly. By adding the variable X_2, the prediction might improve. However, instead of starting from scratch and building a new function with both X_1 and X_2, we follow the stepwise procedure. First we build the parametric linear model for X_2 if X_1 is given, that is, $\hat{x}_2(u_i, v_i) = \alpha_2(u_i, v_i) + \beta_2(u_i, v_i)x_1(u_i, v_i)$. Then we compute the parametric residuals on both the predictor variable X_2 and the response variable Y, that is:

$$x'_2(u_i, v_i) = x_2(u_i, v_i) - (\alpha_2(u_i, v_i) + \beta_2(u_i, v_i)x_1(u_i, v_i)) \tag{11}$$
$$y'(u_i, v_i) = y(u_i, v_i) - (\alpha_1(u_i, v_i) + \beta_1(u_i, v_i)x_1(u_i, v_i)). \tag{12}$$

Finally, we determine a parametric straight-line regression between parametric residuals Y' on X'_2, that is,

$$\hat{y}'(u_i, v_i) = \alpha_3(u_i, v_i) + \beta_3(u_i, v_i)x'_2(u_i, v_i). \tag{13}$$

By substituting the Equations 11-12, we reformulate the latter Equation as follows:

$$\begin{aligned} y(u_i, v_i) - (\alpha_1(u_i, v_i) + \beta_1(u_i, v_i)x_1(u_i, v_i)) = \alpha_3(u_i, v_i) + \\ + \beta_3(u_i, v_i)(x_2(u_i, v_i) - (\alpha_2(u_i, v_i) + \beta_2(u_i, v_i)x_1(u_i, v_i))). \end{aligned} \tag{14}$$

This equation can be equivalently written as:

$$\begin{aligned}\hat{y}(u_i, v_i) = (\alpha_3(u_i, v_i) + \alpha_1(u_i, v_i) - \alpha_2(u_i, v_i)\beta_3(u_i, v_i)) + (\beta_1(u_i, v_i) - \\ - \beta_2(u_i, v_i)\beta_3(u_i, v_i))x_1(u_i, v_i) + \\ + \beta_3(u_i, v_i)x_2(u_i, v_i).\end{aligned} \tag{15}$$

It can be proved that the parametric function reported in this Equation coincides with the geographically weighted model built with Y, $X1$ and $X2$ (in Equation 9) since:

$$\alpha(u_i, v_i) = \alpha_3(u_i, v_i) + \alpha_1(u_i, v_i) - \alpha_2(u_i, v_i)\beta_3(u_i, v_i), \tag{16}$$

$$\beta(u_i, v_i) = \beta_1(u_i, v_i) - \beta_2(u_i, v_i)\beta_3(u_i, v_i) \tag{17}$$

$$\gamma(u_i, v_i) = \beta_3(u_i, v_i). \tag{18}$$

By considering the stepwise procedure illustrated before, two issues remain to be discussed: how the parametric intercept and slope of each straight line regression (e.g. $\hat{y}(u_i, v_i) = \alpha(u_i, v_i) + \beta(u_i, v_i)\ x_j(u_i, v_i)$) are locally estimated across the landscape and how the predictor variables to be added to the function are chosen.

The parametric slope and intercept are defined on the basis of the *weighted* least squares regression method [9]. This method is adapted to fit the geographically distributed arrangement of the data. In particular, for each training location which contributes to the computation of the straight-line function, the parametric slope $\beta(u_i, v_i)$ is defined as follows:

$$\beta(u_i, v_i) = (L^T \mathbf{W}_i L)^{-1} L^T \mathbf{W}_i Z, \tag{19}$$

where L represents the vector of the values of X_j on the training observations, Z is the vector of Y values on the same observations and $\mathbf{W}_i$ is a diagonal matrix defined for the training locations (u_i, v_i)as follows: $\mathbf{W}_i = \begin{pmatrix} w_{i1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{iN} \end{pmatrix}$, where w_{ij} is computed according to Equation 6. Finally, the parametric intercept $\alpha(u_i, v_i)$ is computed according to the function:

$$\alpha(u_i, v_i) = \frac{1}{N}\sum z_i - \beta(u_i, v_i) \times \frac{1}{N}\sum l_i. \tag{20}$$

The choice of the best predictor variable to be included in the model at each step is based on the maximization of the Akaike information criterion (AIC) measure. The AIC is a measure of the relative goodness of fitting of a statistical model. First proposed in [3], AIC is based on the concept of information entropy, and offers a relative measure of the information lost when a given model is used to describe reality. It can be said to describe the trade-off between the bias and

the variance in model construction, or, loosely speaking, between the accuracy and the complexity of the model. In this work we use the corrected AIC (AIC_c) [13] that has proved to give good performance even for small datasets [6]:

$$AIC_c = 2N\ ln(\hat{\sigma}) + N\ ln(2\pi) + N\left(\frac{N+p}{N-2-p}\right), \tag{21}$$

where N is the number of training observations falling in the current leaf, $\hat{\sigma}$ is the standard deviation of training residuals for the response variable and p is the number of parameters (number of variables included in the model – degrees of freedom of a χ^2 test). AIC_c is used to compare regression models, however, it does not provide a test of a model in the usual sense of testing a null hypothesis; i.e. AIC can tell nothing about how well a model fits the data in an absolute sense. This means that if all the candidate models fit poorly, AIC will not give any warning. To overcome this problem, at each step, once the best variable to be added to the function has been identified, the new function is evaluated according to the partial F-test. This allows the evaluation of the statistical contribution of a new predictor variable to the model [9]. If the contribution is not statistically significant, the previous model is kept and no additional variable is added to the regression function.

4.3 Prediction

Once a geographically weighted regression tree T is learned, it can be used for prediction purposes. Let o be any georeferenced observation with unknown response value, then the leaf of T spatially containing o is identified. If a training parameter estimation exists in this leaf computed for the spatial coordinates (u_o, v_o), then these estimates are used to predict $y(u_o, v_o)$ according to the local function associated to the leaf; otherwise, the k closest training parameter estimations falling in the same leaf are identified. Closeness relation is computed by the Euclidean distance. These neighbor parameter estimated are used to obtain k predictions of the response value. A weighted combination of these responses is output. Weights are defined according to the Gaussian schema.

5 Experiments

GeWet is implemented in a Java system which interfaces a MySQL DBMS. In this paper, we have evaluated GeWet on several real benchmark data collections in order to seek answers to the following questions:

1. How does spatial segmentation of the landscape improve the aspatial segmentation performed by the state-of-art model tree learner M5'?
2. How does the stepwise construction of a piecewise space-varying parametric linear function solve the collinearity problem and improve the accuracy of traditional GWR?

3. How does the boundary bandwidth h and the neighborhood size k affect accuracy of geographically weighted regression trees induced by GeWET?

In the next subsections, we describe the data sets, the experimental setting and we illustrate results obtained with these data in order to answer questions 1-3.

5.1 Datasets

GeWET has been evaluated on six spatial regression data collections.

Forest Fires: this dataset[7] collects 512 observations of forest fires in the period January 2000 to December 2003 in the Montesinho Natural Park, Portugal. The predictor variables are: the Fine Fuel Moisture Code, the Duff Moisture Code, the Drought Code, the Initial Spread Index, the temperature in Celsius degrees, the relative humidity, the wind speed in km/h, and the outside rain in mm/m^2. The response variable is the burned area of the forest in ha (with $1ha/100 = 100\ m^2$). The spatial coordinates (U,V) refer to the centroid of the area under investigation on a park map.

USA Geographical Analysis Spatial Data (GASD): this dataset [21] contains 3,107 observations on USA county votes cast in the 1980 presidential election. For each county the explanatory attributes are: the population of 18 years of age or older, the population with a 12th grade or higher education, the number of owner-occupied housing units, and the aggregate income. The response attribute is the total number of votes cast. For each county, the spatial coordinates (U,V) of its centroid are available.

North-West England (NWE): this dataset concerns the region of North West England, which is decomposed into 1011 censual wards. Both predictor and response variables available at ward level are taken from the 1998 Census. They are the percentage of mortality (response attribute) and measures of deprivation level in the ward, according to index scores such as, Jarman Underprivileged Area Score, Townsend Score, Carstairs Score and the Department of the Environment Index. Spatial coordinates (U,V) refer to the ward centroid. By removing observations including null values, only 979 observations are used in this experiment.

Sigmea-Real: this dataset [8] collects 817 observations of the rate of herbicide resistance of two lines of plants (predictor variables, that is, the transgenic male-fertile (SIGMEAMF) and the non-transgenic male-sterile (SIGMEAMS) line of oilseed rape. Predictor variables are the cardinal direction and distance from the center of the donor field, the visual angle between the sampling plot and the donor field, and the shortest distance between the plot and the nearest edge of the donor field. Spatial coordinates (U,V) of the plant are available.

South California: this dataset [14] contains 8033 observations collected for the response variable, median house value, and the predictor variables, median income, housing median age, total rooms, total bedrooms, population, households in South California. Spatial coordinates represent the latitude and longitude of each observation.

5.2 Experimental Setting

The experiments aim at evaluating the effectiveness of the improvement of accuracy of the geographically weighted regression tree, with respect to the baseline model tree learned with the state of art model tree learner M5' and the geographically weighted regression model computed by GWR. We used M5' as it is the state-of-art model tree learner which is considered as the baseline in almost all papers on model tree learning. At the best of our knowledge, no study reveals the existence of a model tree learner which definitely outperforms M5'. The implementation of M5' is publicly available in WEKA, while the implementation of GWR is publicly available in software R. M5' is run in two settings. The first setting adds the spatial dimensions to the set of predictor variables (sM5), the second setting filters out variables representing spatial dimensions (aM5). The empirical comparison between systems is based on the mean square error (MSE). To estimate the MSE, a 10-fold cross validation is performed and the average MSE (Avg.MSE) over the 10-folds is computed for each dataset. To test the significance of the difference in performance, we use the non-parametric Wilcoxon two-sample paired signed rank test.

5.3 Results

In Table 1, we compare the 10-fold average MSE of GeWET with the MSE of M5' (both sM5 and aM5 settings) and GWR . GEWET is run by varying h and k as we intend to draw empirical conclusions on the optimal tuning of these parameters. The results show that MSE comparison confirms that GeWET outperforms both M5' and GWR, generally by a great margin. This result empirically proves the intuitions which lead us to synthesize a technique for the induction of geographically weighted regression trees. First, the spatial-based tree segmentation of the landscape aimed at the identification of rectangular areal units with positively autocorrelated responses improves the performance gained by the baseline M5' which partitions data (and not landscape) according to a Boolean test on the predictor variables. Second, the stepwise computation of a geographically weighted regression model at each leaf is able to select the appropriate subset of predictor variables at each leaf, thus solving the collinearity and definitely improving the baseline accuracy of traditional GWR. The statistical significance of the obtained differences is estimated in terms of the signed rank Wilcoxon test. The entries of Table 2 report the statistical significance of the differences between compared systems estimated with the signed rank Wilcoxon test. By insighting the statistical test results, we observe that there are three datasets, GASD, Forest Fires and South California, where GEWET statistically outperforms each competitor independently from the h and k setting (just with South California and h=20% the superiority of GEWET with respect to its competitor is not statistically obseravabe). The same primacy of GEWET is observable for the remaining three datasets, NWE, SigmeaMS and SigmeaMF, when we select higher values of h ($h >= 30\%$). In general, we observe that a choice of h between 30% and 40% leads to lower MSE in all datasets. GeWET seems to be less sensitive to the choice of k due to the weighting mechanism.

Table 1. 10-fold CV average MSE: GeWET vs M5' and GWR. GeWET is run by varying both neighborhood size k and bandwidth h. M5' is run either by including the spatial dimensions (sM5) in the set of predictor variables or by filtering them out (aM5). GWR is run with the option for automatic bandwidth estimation. The best value of accuracy is in boldface for each dataset.

k	5	5	5	5	10	10	10	10	sM5	aM5	GWR
h	20%	30%	40%	50%	20%	30%	40%	50%			
ForestFires	50.44	49.73	49.84	49.99	50.37	**49.64**	49.76	49.90	87.63	76.88	373.3
GASD	0.10	**0.09**	0.10	0.10	0.10	**0.09**	0.10	0.10	0.14	0.14	0.35
NWE	0.009	0.004	**0.003**	**0.003**	0.009	**0.003**	**0.003**	**0.003**	0.004	0.004	0.004
sigmeaMS	11.11	5.82	4.19	4.58	18.33	5.48	4.42	4.56	**3.98**	4.73	5.22
sigmeaMF	3.66	2.24	**1.81**	1.92	3.67	2.37	1.90	1.92	2.40	1.98	1.98
SouthCalifornia	32.1e5	7.1e4	**5.3e4**	5.4e4	17.4e5	6.9e4	**5.3e4**	5.4e4	6.1e4	8.7e4	8.2e4

6 Conclusions

We present a novel spatial regression technique to tackle specific problems posed by spatial non-stationarity and positive spatial autocorrelation in geographically distributed data environment. To deal with spatial non-stationarity we decide to learn piecewise space-varying parametric linear regression functions, where the parameters are the coefficients of the linear combination which are estimated to vary across the space. Moreover, we combine local model learning with a tree structured segmentation approach that is tailored to recover the functional form of a spatial model only at the level of each areal segment of the landscape. A new stepwise technique is adopted to select the most promising predictors to be included in the model, while parameters are estimated at every point across the local area. Parameter estimation solves the problem of least square weighted regression and uses the concept of a positively autocorrelated neighborhood to determine a correct estimate of the weights. Experiments with several benchmark data collections are performed. These experiments confirm that the induction of our geographically weighted regression trees significantly improves both the local estimation of parameters performed by GWR and the global estimation of parameters performed by classical model tree learners like M5'. As future work, we plan to investigate techniques for automating the tuning of both bandwidth and neighborhood size. Additionally, we plan to frame this work in a streaming environment, where geographically distributed sensors continuously transmit observations across the time. This means that, in addition to the space dimension, the time dimension of data would be also considered.

Acknowledgment

The paper is realized with the contribution of "Fondazione Cassa di Risparmio di Puglia". This work is partial fulfillment of the research objective of ATENEO-2010 project entitled "Modelli e Metodi Computazionali per la Scoperta di Conoscenza in Dati Spazio-Temporali".

Table 2. The signed Wilcoxon test on the accuracy of systems: GeWET vs M5 (sM5 or aM5); GeWET vs GWR. The symbol "+" ("-") means that GeWET performs better (worse) than sM5, aM5 or GWR. "++" ("–") denote the statistically significant values ($p \leq 0.05$).

k	GeWet vs.	5	5	5	5	10	10	10	10
h		20	30	40	50	20	30	40	50
ForestFires	sM5	++	++	++	++	++	++	++	++
	aM5	++	++	++	++	++	++	++	++
	GWR	++	++	++	++	++	++	++	++
GASD	sM5	++	++	++	++	++	++	++	++
	aM5	++	++	++	++	++	++	++	++
	GWR	++	++	++	++	++	++	++	++
NWE	sM5	–	++	++	++	–	++	++	++
	aM5	–	++	++	++	–	++	++	++
	GWR	–	–	-	-	–	–	-	-
sigmeaMS	sM5	–	–	+	+	–	–	+	+
	aM5	–	-	+	+	–	-	+	+
	GWR	–	+	+	+	–	+	+	+
sigmeaMF	sM5	–	+	++	+	++	=	++	+
	aM5	–	-	+	+	–	-	+	+
	GWR	–	+	+	+	-	+	+	+
SouthCalifornia	sM5	-	+	++	++	-	++	++	++
	aM5	=	++	++	++	=	++	++	++
	GWR	=	++	++	++	=	++	++	++

References

1. C. S. A. Petrucci, N. Salvati. The application of a spatial regression model to the analysis and mapping of poverty. *Environmental and Natural Resources Series*, 7:1–54, 2003.
2. M. C. A. S. Fotheringham, C. Brunsdon. *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley, 2002.
3. H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
4. V. Bogorny, J. F. Valiati, S. da Silva Camargo, P. M. Engel, B. Kuijpers, and L. O. Alvares. Mining maximal generalized frequent geographic patterns with knowledge constraints. In *ICDM 2006*, pages 813–817. IEEE Computer Society, 2006.
5. C. Brunsdon, J. McClatchey, and D. Unwin. Spatial variations in the average rainfallaltitude relationships in great britain: an approach using geographically weighted regression. *International Journal of Climatology*, 21:455–466, 2001.
6. K. Burnham and D. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer-Verlag, 2002.
7. P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. In *EPIA 2007*, pages 512–523. APPIA, 2007.
8. D. Demšar, M. Debeljak, C. Lavigne, and S. Džeroski. Modelling pollen dispersal of genetically modified oilseed rape within the field. In *Annual Meeting of the Ecological Society of America*, page 152, 2005.
9. N. R. Draper and H. Smith. *Applied regression analysis*. Wiley, 1982.

10. G. Góra and A. Wojna. Riona: A classifier combining rule induction and k-nn method with automated selection of optimal neighbourhood. In *ECML '02*, pages 111–123. Springer-Verlag, 2002.
11. L. Hordijk. Spatial correlation in the disturbances of a linear interregional model. *Regional and Urban Economics*, 4:117–140, 1974.
12. Y. Huang and Y. Leung. Analysing regional industrialisation in jiangsu province using geographically weighted regression. *Journal of Geographical Systems*, 4:233–249, 2002.
13. C. M. Hurvich and c.-L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, June 1989.
14. P. Kelley and R. Barry. Sparse spatial autoregressions. *Statistics and Probability Letters*, 33:291–297, 1997.
15. P. Legendre. Spatial autocorrelation: Trouble or new paradigm? *Ecology*, 74:1659–1673, 1993.
16. J. LeSage and K. Pace. Spatial dependence in data mining. In *Data Mining for Scientific and Engineering Applications*, pages 439–460. Kluwer Academic, 2001.
17. C. Levers, M. Brckner, and T. Lakes. Social segregation in urban areas: an exploratory data analysis using geographically weighted regression analysis. In *13th AGILE International Conference on Geographic Information Science 2010*, 2010.
18. P. Longley and A. Tobon. Spatial dependence and heterogeneity in patterns of hardship: an intra-urban analysis. *Annals of the Association of American Geographers*, 94:503–519, 2004.
19. D. Malerba, M. Ceci, and A. Appice. Mining model trees from spatial data. In *PKDD 2005*, volume 3721 of *LNCS*, pages 169–180. Springer-Verlag, 2005.
20. T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
21. P. Pace and R. Barry. Quick computation of regression with a spatially autoregressive dependent variable. *Geographical Analysis*, 29(3):232–247, 1997.
22. S. Rinzivillo, F. Turini, V. Bogorny, C. Körner, B. Kuijpers, and M. May. Knowledge discovery from geographical data. In *Mobility, Data Mining and Privacy*, pages 243–265. Springer-Verlag, 2008.
23. N. Shariff, S. Gairola, and A. Talib. Modelling urban land use change using geographically weighted regression and the implications for sustainable environmental planning. In *Proceeding of the 5th International Congress on Environmental Modelling and Software Modelling for Environments Sake*. iEMSs, 2010.
24. S. Shekhar and S. Chawla. *Spatial databases: A tour*. Prentice Hall, 2003.
25. Y. Wang and I. Witten. Inducing model trees for continuous classes. In M. van Someren and G. Widmer, editors, *Proceedings of the 9th European Conference on Machine Learning, ECML 1997*, volume 1224 of *Lecture Notes in Computer Science*, pages 128–137. Springer-Verlag, 1997.

Spatio-Temporal Reconstruction of Un-Sampled Data in a Sensor Network

Pietro Guccione[1], Anna Ciampi[2], Annalisa Appice[2], Donato Malerba[2], and Angelo Muolo[3]

[1] Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari - Bari, Italy
pietro.guccione@ieee.org
[2] Dipartimento di Informatica, Università degli Studi di Bari via Orabona, 4 - 70126 Bari - Italy {aciampi,appice,malerba}@di.uniba.it
[3] Rodonea s.r.l. - via Fiume, 1 - Monopoli(Bari), Italy angelo.muolo@rodonea.com

Abstract. Acquisition of information by a pervasive sensor network raises a fundamental trade-off: higher density of sensors provides more measurements, higher resolution and better accuracy, but requires more communication and processing activities. This paper proposes an approach to significantly reduce the number of transmitting nodes while maintaining a reasonable degree of accuracy when the measurements of the inactive nodes are restored. The approach is based on the combination of a spatio-temporal algorithm to represent the measured data in trend-based cluster prototypes and a multivariate interpolation, the Inverse Distance Weighting (IDW), used to estimate measurements where they are unavailable. The combination of the twos allows both to efficiently summarize the data stream and to keep enough accuracy in estimating the unknown measurements at un-sampled locations. Our method has been evaluated within a real sensor monitoring environment.

1 Introduction

Spatially distributed sensor networks are emerging as a tool for monitoring ubiquitous environments. Each sensor node has computing and storage abilities and its longevity depends on the smart use of energy. The uncertainty of the application environments, the scarcity of communication ability and of the transmission bandwidth suggest to adopt a meaningful subset of the whole network and to estimate the un-sampled measures from the available measures as accurately as possible [11]. In this paper we do not face the problem of how to optimally select the number and the location of the sensors in a network [6, 5, 9]; instead we focus on the estimation of missing (un-sampled) values after an efficient summarized description of the network data stream has been performed.

Although the missing values estimation has been studied in many fields, from artificial intelligence to data mining and signal processing, sensor networks pose new challenges since the sensed physical quantity is temporal and spatial correlated. The idea is to interpolate the missing measures of a network on the basis of a spatio-temporal aggregate of the observed stream, the trend clusters [2].

The stream is segmented in consecutive *windows*. A trend cluster is a cluster of sensors which transmit values whose temporal variation, called *trend polyline*, is similar along the time of the window. The set of trend clusters which is discovered over each window is a stream aggregate which reflects spatial correlation and temporal dependence in data. In [2] authors show that trend clusters can be discovered in real time and stored in database as a summary of the stream. Past measures of sensors can be approximately reconstructed from the summary by selecting, for each window, the cluster they belong to and returning the associated trend polyline. By performing a step toward, the polyline of each cluster could be, in principle, associated to every position of the *spatial cloud* of the cluster, so that for each missing (or generic) position inside the area the unknown data would be provided by the polyline itself. This is a naive way of interpolation which approximates a random position with the polyline of the nearest cluster. It is a sort of zero-order (or nearest neighbor) interpolation. However, it is reasonable to think that sensors that lie halfway between two or more clusters shall measure values which are a *combination* of the polylines of these neighbor clusters, due to reciprocal influence of physical measures. The combination is actually achieved by means of the Inverse Distance Weighting [8] interpolation which assigns values to un-sampled positions by using a weighted sum of the known values. The weights are inversely proportional to the distance of the known points from the un-sampled location, thus giving more importance to the nearest data. The contribution of this work is the use of the trend cluster model to provide a reliable interpolation of missing data. The interpolation task starts from a reduced set of information, the trend clusters, which accounts for both spatial and temporal data correlation.

The paper is organized as follows. In Section 2 the spatially distributed data stream, the trend cluster model and the interpolation task are described. Details of the interpolation algorithm are given in Section 3. Results on a real sensor network stream and discussion are exposed in Section 4; then conclusions follow.

2 Problem Formulation

2.1 Snapshot model and Window Model

The *snapshot model* [1], allows to represent geo-referenced data by means of 2-D points of an Euclidean space and timestamped on a regular time axis. By considering a time domain $\mathbb{T}$ which is absolute, linear and discrete, a snapshot D_i models the set of the geo-referenced measurements which are timestamped with t_i ($t_i \in \mathbb{T}$). The spatial arrangement of D_i is modeled by means of a field function [7] defined as $f_i : K_i \mapsto \mathit{Attribute\ domain}$. The field domain K_i ($K_i \subseteq \mathbb{R}^2$) is the set of the 2-D points geo-referencing the sampled sensors which transmit at t_i. This way, a sensor network stream D is a stream of snapshots ($D = D_1, D_2, \ldots D_i, ...$) which continuously feed a server at the equally spaced time points of $\mathbb{T}$.

The *count-based window model* [3] decomposes D into consecutive windows of w snapshots arriving in series. Windows are enumerated such that the i-th

window comprises w snapshots timestamped in the time interval $[t_{(i-1)w+1}, t_{iw}]$. Snapshots can be locally enumerated in the owner window. In this formulation, D is seen as a stream of windows of snapshots. Formally,

$$\underbrace{D_1^1, D_2^1, \ldots, D_w^1}_{W_1}, \underbrace{D_1^2, D_2^2, \ldots, D_w^2}_{W_2}, \ldots, \underbrace{D_1^i, D_2^i, \ldots, D_w^i}_{W_i}, \ldots$$

In this window-based representation of the stream, a snapshot is denoted as D_n^i, where the index n ranges between 1 and w and enumerates a snapshot within the owner window W_i. The index i is to denote the i-th window of the the stream.

2.2 Trend Cluster Model

A trend cluster is a *cluster* of spatially close sensors which transmit numeric values whose temporal variation, called trend polyline, is similar over a time horizon. Formally, a trend cluster is the triple [*time horizon*, *spatial cluster*, *trend polyline*]. The *time horizon* is the range of time of a count-based window. The *spatial cluster* is the set of the *spatially close* sources which transmit values whose temporal variation is similar along the time horizon of the window. The *trend polyline* is the representation of the temporal variation of the measurements in the cluster. It is represented by a series of w median values computed at the consecutive time points of the window. Formally,

$$\textit{trend polyline} = \langle v_1^i[tc], v_2^i[tc], \ldots, v_{w-1}^i[tc], v_w^i[tc] \rangle \quad (1)$$

where v_j^i is the median of the measurements transmitted at the time point $t_{(i-1)w+j}$ $(j = 1, 2, \ldots, w)$ by sources grouped into the spatial cluster tc.

The *spatial closeness* between sources is computed according to some user-defined distance relation (e.g., nearby, far away) implicitly defined by the latitude-longitude coordinates of the sensors. The similarity in the temporal variation of the clustered sources depends on a user-defined parameter which is called *trend cluster similarity threshold* and is denoted by δ. According to δ, each sensor grouped in a cluster transmits values along the window horizon which differs at worst δ from the trend polyline of the cluster. The differences are computed time point-by-time point by resorting to the Euclidean distance. The algorithmic details concerning the trend cluster computation are reported in [2].

2.3 Interpolation Task

The goal of a spatial interpolation task is to generate new data on un-sampled positions exploiting the existing known values. The main hypotheses to have reliable estimates is that the physical measure is spatially correlated (near nodes exhibit similar observations) and that this similarity is as high as sensors are close each others, i.e. the power spectral density of the process implied by measures is band-limited [4]. Under these hypotheses the interpolation is the natural choice

to solve the problem of finding data at un-sampled locations. The idea is to interpolate along the windows by using the aggregated measures (the polylines) provided by the trend clusters which are expected to be close to the un-sampled locations. However, since each polyline refers to a group of nodes (the cluster), we must *assign* it to a specific spatial location, that we call *cluster centroid*.

The cluster centroids are not equally spaced on the region; for this reason we use an interpolation method that accounts for irregular sampling, the IDW interpolation. This method allows to determine the interpolated value in a generic position $u = (x, y)$ by means of a weighted average of the values at the known points. The weights are proper-defined functions of the distance between u and each known cluster centroid.

3 The Trend Cluster based Interpolation Algorithm

Once the trend clusters are known for a window, the interpolation is performed in two steps: (1) for each cluster of nodes the centroid position, $(\hat{x}_{tc}, \hat{y}_{tc})$ is computed and it is assigned to the polyline of the trend cluster tc; (2) for each un-sampled location $u = (x_u, y_u)$ its interpolated polyline along the window is computed by properly combining the polylines associated to the neighbor centroids.

Centroid Computation. Given a trend cluster tc which groups both a set of c sensors with coordinates $\{(x_1, y_1), (x_2, y_2), \cdots, (x_c, y_c)\}$ and a trend polyline $\langle v_1^i[tc], v_2^i[tc], \ldots, v_w^i[tc]\rangle$, the location $\hat{c}_{tc} = (\hat{x}_{tc}, \hat{y}_{tc})$ of the centroid of tc is computed as follows:

$$\hat{x}_{tc} = \frac{1}{c}\sum_{j=1}^{c} x_j \qquad \hat{y}_{tc} = \frac{1}{c}\sum_{j=1}^{c} y_j \tag{2}$$

By means of Eq. (2) we perform the geo-referencing of a cluster's trend polyline to a specific location, i.e. that of the cluster centroid.

Interpolation. The *Inverse Distance Weighting (IDW)* is used to interpolate the unknown measure at any generic position $u = (x_u, y_u)$ by means of a weighted average of the measures associated at the known centroid points. The weights are properly-defined functions of the distance between u and each centroid. For each window, let N be the number of centroids. For each time-point t_i of the window ($i = 1 \cdots w$), the interpolated measure $v_i[u]$ at the un-sampled position u and at the time point t_i is computed as follows:

$$\hat{v}_i[u] = \frac{\sum_{tc=1}^{N} w(u, \hat{c}_{tc}) v_i[tc]}{\sum_{tc=1}^{N} w(u, \hat{c}_{tc})}. \tag{3}$$

where the denominator normalizes the sum of weights.

The weights $w(u, \hat{c}_{tc})$ are computed as the inverse of a power of the distance between u and $\hat{c}_{tc}$, that is:

$$w(u, \hat{c}_{tc}) = (\mathbf{d}(u, \hat{c}_{tc}))^{-p} \tag{4}$$

In a geographical context, it is natural to use the Euclidean distance to define $\mathbf{d}(u, \hat{c}_{tc})$. The concept behind Eq. (3) is that the interpolation at an un-sampled position is a function of all the known values and depends from them in a relation inversely proportional to the distance, i.e. the nearer is a known value the stronger is its influence. The power of this influence is driven by the *power parameter* p, which is a positive and real number. If the value of p is less than the dimensionality of the problem (in our context the dimensionality is 2) the interpolation is rather dominated by the distant points. On the other hand, greater values of p give more influence to the values closer to the un-sampled position; for $p \to \infty$ the IDW method provides basically the same results of the nearest neighbor interpolation.

Although in Eq. (3) all the cluster centroids are considered to compute the interpolated value, we expect that a maximum *correlation distance* exists and the too distant centroids should not be included in the interpolation. Thus, we set a bound such that the centroids contribute to interpolation only if they are inside a given *sphere*. The center of this *interpolation sphere* is the un-sampled position and the radius is a parameter R_c. The undesirable consequence of setting R_c is that a different number of centroids is used for different un-sampled locations; but, on the other hand, this is the unavoidable effect of dealing with irregularly sampled data. Where data are more dense interpolation shall be more reliable, fading and breaking off in areas where the known values are too far. Under these consideration we re-define (4) as follows:

$$w(u, \hat{c}_{tc_k}) = \begin{cases} (\mathbf{d}(u, \hat{c}_{tc_k}))^{-p} & \text{if } \mathbf{d}(u, \hat{c}_{tc}) \leq R_c \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

The IDW interpolation provides a method which works both online and off-line. The result is a trade off between accuracy (better than nearest neighbor) and computational load, which would be higher with statistical methods. On the other hand, IDW result is a function of all the parameters introduced before. These parameters should be set such that the gap between real and interpolated measures, in a statistical sense (i.e. null average and minimum root mean square of the error $\epsilon = v - \hat{v}$) is minimum. A statistical model computation should be performed but it would affect computation load. Thus, we opted for a qualitative setting of parameters, which are dependent on the specific domain of the measured feature.

4 Experimental Results

Experiments are performed on the South American Air Temperature dataset [10]. Our aim is to evaluate the accuracy of the proposed technique when just a

subset of sensors transmit measures and IDW interpolation is used to retrieve the measures of un-sampled sensors. Experiments are run on Intel®Core(TM) 2 DUO CPU $E4500$ @2.20GHz with 2.0 GiB of RAM Memory, running Ubuntu 11.04.

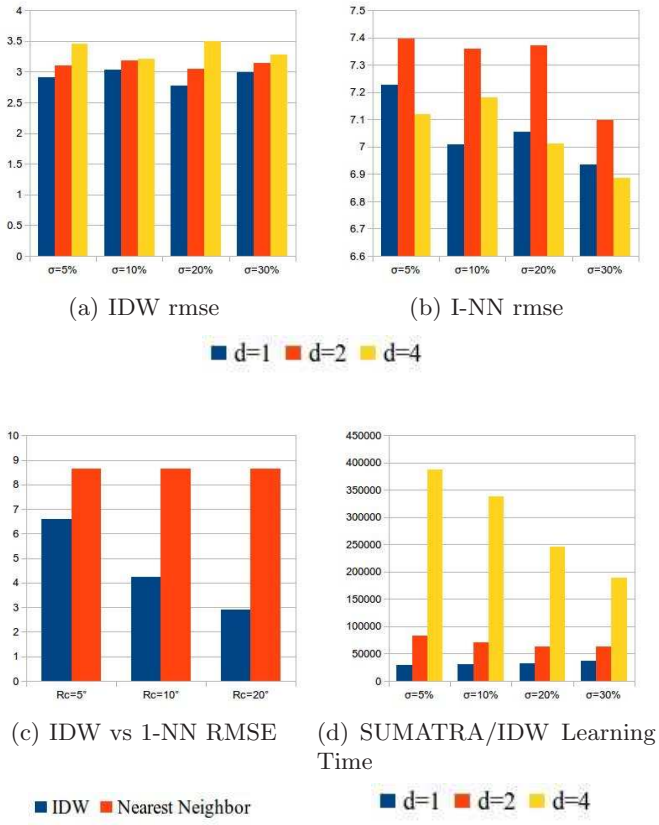
Sensor Network Description. The South American Air Temperature network [10] was used to collect monthly-mean air temperature measurements (in ^{o}C) between 1960 and 1990; available data are provided as interpolated over a 0.5deg by 0.5deg of latitude/longitude grid. The grid nodes are centered on 0.25deg for a total of 6477 sensors. The number of nearby stations that influence a grid-node estimate is 20 in average. The temperature values range between -7.6 and 32.9^oC.

Experimental Settings. We randomly choose a subset of sensors of the network which are considered switched-on and we use SUMATRA [2] to learn trend clusters from these sensors. Then, we use trends and centroids of the clusters to interpolate the unknown temperature measure for the sensors which have been considered switched-off. We use the experimental setting described in [2], i.e., window size $w = 12$ and trend similarity threshold $\delta = 4^o$C. The IDW interpolation is performed by setting $p = 3$. The IDW interpolation is compared to the 1-Nearest Neighbor interpolation (1NN) that corresponds to simply assign the trend polyline of the closest centroid to each switched-off sensor.

Accuracy We evaluate the accuracy of the proposed method by means of the root mean square error (rmse) which is found when the interpolated value is used to approximate the real value.

We first evaluated the interpolation accuracy by varying the percentage σ of switched-off nodes in the network ($\sigma = 5\%, 10\%, 20\%, 30\%$). It is noteworthy that when the number of switched-off nodes increases a sensor may loose most of its neighbors in the network. In [2] each sensor is connected to those located in the $d^\circ \times d^\circ$ around cells of the grid with $d = 0.5$. In this set of experiments, we considered d ranging among 1, 2 and 4 and $R_c = 20$. The rmse performed by IDW and 1-NN is plotted in Fig. 1(a) and Fig. 1(b), respectively. The results, plotted by varying d and σ, confirm that IDW is always more accurate than 1-NN and basically insensible to σ. Additionally the rmse of IDW remains below δ ($\delta = 4$ in the experiments), i.e. the theoretical upper bound of the summarization error performed by SUMATRA. The IDW rmse increases with d, as expected: lower values of d better describe the real network relation, thus, as d increases, the *quality* of neighbor relation becomes weaker. On the other hand, keeping d low causes a higher cluster fragmentation (and so a major number of bytes to describe the clusters summarization), particularly as σ increases. We hence evaluated how the IDW rmse is affected by R_c. The rmse of IDW interpolation is then plotted in Fig. 1(c) for $R_c = 5, 10, 20$ when $d = 1$. What we found is that rmse decreases as R_c increases, as a higher number of cluster centroids is used. Higher R_c makes the interpolation more accurate and this confirms our hypothesis of spatial correlation in data. Final considerations concern the

analysis of the computation time. In Fig. 1(d) are reported the times spent in average to discover the trend clusters over each window, to store them in the data warehouse and to interpolate the missing values. The computation time are collected by varying σ and d and setting $R_c = 20$. Results show that the summarization/interpolation task can be performed on real-time because the time to process a window is lower, in average, than the time needed to buffer a new window.



(a) IDW rmse (b) I-NN rmse

(c) IDW vs 1-NN RMSE (d) SUMATRA/IDW Learning Time

Fig. 1. RMSE: IDW(a) and 1-NN (b) by varying distance d to construct the network and percentage σ of switched-off nodes. IDW vs 1-NN (c) by varying R_c, but setting $\sigma = 5\%$ and $d = 1$. Learning time (d): Average time $[msec]$ spent to perform algorithm (SUMATRA trend cluster performed on switched-on sensors and interpolation on switched-off). Results refer to the averaged measures over the 31 windows of the entire dataset.

5 Conclusion

Trend cluster represents an effective model to summarize spatio-temporal data of a sensors network, as reconstruction of original data from the summaries stored in a data warehouse is efficient and reliable (in term of accuracy). A spatial irregular interpolation provides the possibility to reconstruct the values in missing locations. Results show that the method is efficient and accurate, at least w.r.t. 1-NN method. Interpolation can be driven by changing a set of parameters that should be properly assessed for each case-study under exam. The proposed method provides meaningful results with a significant percentage of missing nodes w.r.t. the whole network. Assessment of the IDW parameters and of the relation between distance, radius and percentage of switched-off nodes is anyway worthy of further investigation.

Acknowledgments. The paper is realized with the contribution of "Fondazione Cassa di Risparmio di Puglia".

References

1. C. Armenakis. Estimation and organization of spatio-temporal data. In *GIS*, 1992.
2. A. Ciampi, A. Appice, and D. Malerba. Summarization for geographically distributed data streams. In *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2010*, volume 6278 of *LNCS*, pages 339–348. Springer-Verlag, 2010.
3. M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *ACM SIGMOD Record*, 34(2):18–26, 2005.
4. E. H. Isaaks and R. M. Srivastava. *An Introduction to Applied Geostatistics*. Oxford University Press, 1989.
5. M. Perillo, Z. Ignjatovic, and W. Heinzelman. An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique. In *In Information Processing in Sensor Networks*, pages 116–123, 2004.
6. H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-noy, and T. Brown. A survey of sensor selection schemes in wireless sensor networks. In *In SPIE Defense and Security Symposium Conference on Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, 2007.
7. S. Shekhar and S. Chawla. *Spatial databases: A tour*. Prentice Hall, 2003.
8. D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *ACM National Conference*, pages 517–524, 1968.
9. P. Szczytowski, A. Khelil, and N. Suri. Asample: Adaptive spatial sampling in wireless sensor networks. In *SUTC/UMC*, pages 35–42, 2010.
10. S. A. A. Temperature. `http://climate.geog.udel.edu/~climate/html_pages/sa_air_clim.html`.
11. R. Willett, A. Martin, and R. Nowak. Backcasting: A new approach to energy conservation in sensor networks. In *of Information Processing in Sensor Networks (IPSN04*, 2003.

Face-to-Face Contacts during a Conference: Communities, Roles, and Key Players

Martin Atzmueller[1], Stephan Doerfel[1], Andreas Hotho[2], Folke Mitzlaff[1], and Gerd Stumme[1]

[1] University of Kassel, Knowledge and Data Engineering Group
Wilhelmshöher Allee 73, 34121 Kassel, Germany

[2] University of Würzburg, Data Mining and Information Retrieval Group
Am Hubland, 97074 Würzburg, Germany

{atzmueller, doerfel, mitzlaff, stumme}@cs.uni-kassel.de, hotho@informatik.uni-wuerzburg.de

Abstract. This paper analyzes profile data and contact patterns of conference participants in a social and ubiquitous conferencing scenario: We investigate user-interaction and community structure of face-to-face contacts during a conference, and examine different roles and their characteristic elements. The analysis is grounded using real-world conference data capturing descriptive profile information about participants and their face-to-face contacts.

1 Introduction

During the last decade, Web 2.0 and social semantic web applications have already woven themselves into the very fabric of everyday life. Many applications, e.g., social networks (Facebook, LinkedIn, Xing) or Web 2.0 messaging tools (Twitter) are extensively used in various application domains. However, conferences usually do not make use of more dynamic and community-based features, e.g., schedules are commonly arranged in a static way. Knowledge discovery techniques could often be applied ahead of the conference, e.g., for recommending reviewers to submissions or later talks to participants. Furthermore, dynamic adaptions are enabled during the conference by ubiquitous computing approaches, e.g., based on RFID-tokens.

In this paper, we focus on the analysis of social data and contact patterns of conference participants: We consider communities of participants and their visited talks. Additionally, we analyze face-to-face contacts of conference participants during the duration of the conference. We examine different explicit and implicit roles of the participants, validate the community structures, and analyze various structural properties of the contact graph.

Our contribution is three-fold: We present an in-depth analysis of the social relations and behavioral patterns at the conference, identify characteristics of special roles and groups, and sketch approaches on how the mined information can be implemented in social conferencing applications. We focus on profiles of the participants and their face-to-face contacts. Considering these, we analyze community structures during the conference. Additionally, we consider the special interest groups as given by

a participant during registration in comparison to the emerging communities at the conference. Finally, we perform a description and characterization of different roles and groups, e.g., organizers and different subcommunities at a conference, in order to identify characteristic factors.

The rest of the paper is structured as follows: In Section 2 we discuss some social applications for conferences, and issues of privacy and trust. After that, Section 3 considers related work. Next, Section 4 provides the grounding of our approach presenting an in-depth analysis and evaluation of real-world conference data. Finally, Section 5 concludes the paper with a summary and interesting directions for future research.

2 Social Conferencing

During a conference, participants encounter different steps and phases: Preparation (before the conference), during the actual conference, and activities after the conference. Appropriate talks and sessions of interest need to be selected. Talks and discussions, for example, need to be memorized. Additionally, social contacts during a conference are often essential, e.g., for networking, and are often revisited after a conference, as are the visited talks. All of these steps are supported by the CONFERATOR system: It is under joint development by the School of Information Sciences, University of Pittsburgh (conference management component, as a refinement of the Conference Navigator [1]) and the Knowledge and Data Engineering group at the university of Kassel (social and ubiquitous PEERRADAR[3] component).

A first prototype of CONFERATOR [2], developed by the Knowledge and Data Engineering group was successfully applied at the LWA 2010 conference at the University of Kassel in October 2010. The applied system is based on the UBICON framework featuring the PEERRADAR application for managing social and ubiquitous/real contacts. This is implemented by embedding social networks such as Facebook, XING, and LinkedIn. Furthermore, advanced RFID-Proximity technology for detecting the location of participants and contacts between conference participants is implemented utilizing active RFID proximity-tags, cf., [3]. The system also provides the conference information using a visual browser for managing the conference content and phases, i.e., by providing information about talks and the conference schedule, in preparation for integrating the Conference Navigator application.

In CONFERATOR, privacy is a crucial issue: A variety of user data is collected and therefore appropriate steps for their secure storage and access were implemented.

CONFERATOR implements privacy measures using a refined trust system: It features several privacy levels (private, trusted, public) for organizing access to different items, e.g., location, profile, and contact information. In addition, in the analysis we aim at providing implicit k-anonymity in the presentation and discussion, since we provide results at the level of special interest groups, or provide detailed results only targeting groups containing at least five participants.

[3] `http://www.ubicon.eu`

3 Related work

Regarding the tracking and analysis of conference participants, there have been several approaches, using RFID-tokens or Bluetooth-enabled devices. Hui et al. [4] describe an application using Bluetooth-based modules for collecting mobility patterns of conference participants. Furthermore, Eagle and Pentland [5] present an approach for collecting proximity and location information using Bluetooth-enabled mobile phones, and analyze the obtained networks.

One of the first experiments using RFID tags to track the position of persons on room basis was conducted by Meriac et al. (cf., [6]) in the Jewish Museum Berlin in 2007. Cattuto et al. [7] added proximity sensing in the Sociopatterns project[4]. Barrat et al. [8] did further experiments. Alani and colleagues, e.g., [3], also added contact information from social online networks. Our work uses the same technical basis (RFID-tokens with proximity sensing), on top of the Sociopatterns project, which allows us to verify their very interesting results independently. Furthermore, in this paper we significantly extend the analysis, since we are able to use further techniques in order to characterize different roles, communities and participant relations.

The conference navigator by Brusilovsky [1] allows researchers attending a conference to organize the conference schedule and provides a lot of interaction capabilities. However, it is not connected to the real live activity of the user during the conference. In the application, we measured face-to-face conctacts, increased the precision of the localization component compared to previous RFID-based approaches, and linked together tag information and the schedule of a workshop week. Furthermore, we implemented a light-weight integration with BibSonomy and other social systems used by participants. This is the basis for new insights into the behavior of all participants.

Thus, in comparison to the approaches mentioned above, we are able to perform a much more comprehensive evaluation of the patterns acquired in a conference setting, since our data provides a stable ground truth for communities (the special interest groups). This provides a grounding not only considering the verification of the structural properties of the mobility patterns, but also given by the roles, and communities.

Considering different "roles" of nodes and finding so called "key actors" has attracted a lot of attention. Ranging from different measures of centrality (cf., [9]) to the exploration of topological graph properties [10] or structural neighborhood similarities [11]. We focus on a metric of how much a node connects different communities, cf., [12], since it allows to consider initially given community structures.

4 Grounding

In this section, we present an analysis of the collected conferencing data. After introducing some preliminaries, we first discuss a grounding of the communities given through the assignment of participants to special interest groups. After that, we consider explicit roles (Prof., PostDoc, PhD-Student, Student) and organizing roles (organizers vs. regular participants).

[4] http://www.sociopatterns.org

4.1 Preliminaries

In the following section, we briefly introduce basic notions, terms and measures used throughout this paper. We presume familiarity with the concepts of *directed* and *undirected* Graphs

$$G = (V, E)$$

for a finite set V of nodes with edges $(u, v) \in V \times V$ and $\{u, v\} \subseteq V$ respectively.

In a *weighted* graph, each edge is associated with a corresponding edge weight, typically given by a mapping from E to $\mathbb{R}$. We freely also use the term *network* as a synonym for a graph. For more details, we refer to standard literature, e.g., [13,14].

In the context of social network analysis, a *community* within a graph is defined as a group of nodes such that group members are densely connected among each other but sparsely connected to nodes outside the community [15] (based on the underlying observation that individuals tend to interact more tightly within a group of somehow related persons). Community structure was observed in several online social networks [16,17] and is sometimes also called "virtual community" [18].

For formalizing and assessing community structure in networks, this work focuses on the modularity measure [15] which is based on comparing the number of edges within a community to the expected such number given a null-model (i.e., a randomized model). Thus, the modularity of a community clustering is defined to be the fraction of the edges that fall within the given clusters minus the expected such fraction if edges were distributed at random.

This can be formalized as follows: The modularity Mod of a graph clustering is given by

$$Mod = \frac{1}{2m} \sum_{i,j} \big(A_{i,j} - \frac{k_i k_j}{2m}\big) \delta(C_i, C_j)\,,$$

where A is the adjacency matrix, C_i and C_j are the clusters containing the nodes i and j respectively, k_i and k_j denote the degree of i and j, $\delta(C_i, C_j)$ is the *Kronecker delta* symbol that equals 1 if $C_i = C_j$, and 0 otherwise; $m = \frac{1}{2} \sum_{ij} A_{ij}$ is the total number of edges in the graph.

A straightforward generalization of the above formula to a modularity measure $wMod$ in weighted networks [19] considers A_{ij} to be the weight of the edge between nodes i and nodes j, and replaces the degree k_i of a node i by its strength $str(i) = \sum_j A_{ij}$, i. e., the sum of the weights of the attached edges.

4.2 Available Data

For capturing social interactions, RFID proximity tags of the Sociopatterns project were applied. 70 out of 100 participants volunteered to wear an RFID tag which (approximately) detected mutual face-to-face sightings among participants with a minimum proximity of about one meter. Each such sighting with a minimum length of 20 seconds was considered as a contact which ended when the corresponding tags did not detect an according sighting for more than 60 seconds.

Using the contact data we generated undirected networks $\mathsf{LWA}[\geq i]_*$, $\mathsf{LWA}[\geq i]_\Sigma$, and $\mathsf{LWA}[\geq i]_\#$. An edge $\{u, v\}$ is created, iff a contact with a duration of at least i *minutes* among participants u and v was detected ($i = 1, \ldots, 15$). For $i \geq 5$[minutes], for example, we can filter out "small talk" conversations. In $\mathsf{LWA}[\geq i]_\#$ the edge $\{u, v\}$ is weighted with the *number* of according contacts, in $\mathsf{LWA}[\geq i]_\Sigma$ it is weighted with the *sum* of all according contact durations whereas $\mathsf{LWA}[\geq i]_*$ is unweighted.

Table 1 contains some statistics for $\mathsf{LWA}[\geq i]_*$, $i = 0, 5, 10$. The diameters and average path lengths coincide with those given in [8] for the Hypertext Conference 2009 (HT09). Figure 1 shows the degree and contact length distribution for $\mathsf{LWA}[\geq 0]_*$. The latter exhibits characteristics comparable with those given for HT09, whereas the degree distributions differ by exhibiting two peaks – one around 10 and one around 20 – in contrast to only one peak around 15 for HT09. We hypothesize that this deviation is due to a more pronounced influence of the conference organizers at LWA 2010 in relation to the total number of participants (approx. 15% of the participants in $\mathsf{LWA}[\geq 0]_*$ were organizers). This hypothesis is supported by removing all organizers from $\mathsf{LWA}[\geq 0]_*$ and recalculating the degree distribution, yielding a single peak in the interval $[15, 20)$.

Other statistics (e. g., strength distribution, among others) also suggest evidence for structural similarities among HT09 and LWA 2010. Therefore, we conclude, that LWA 2010 was a typical technical conference setup, and results obtained at the LWA 2010 are expected to hold in other conference scenarios with similar size, too.

Table 1. High level statistics for different networks: Number of nodes and edges, Average degree, Average path length APL, diameter d, clustering coefficient C, number and size of the largest weakly connected component #CC and $|\text{CC}|_{\max}$ respectively. Additionally the size of the contained special interest groups is given.

Network	$\|V\|$	$\|E\|$	Avg.Deg.	APL	d	Density	C	#CC	$\|\text{CC}\|_{\max}$	|KDML|	|WM|	|IR|	|ABIS|
$\mathsf{LWA}[\geq 0]_*$	70	812	23.20	1.72	4	0.34	0.55	1	70	37	16	10	7
$\mathsf{LWA}[\geq 5]_*$	65	227	6.99	2.53	5	0.11	0.33	1	65	34	15	9	7
$\mathsf{LWA}[\geq 10]_*$	56	109	3.89	3.09	7	0.07	0.31	3	50	31	12	7	6

Furthermore, we extracted the "*visited talks*", i.e., the talks visited by each participant using the RFID information, resulting in 773 talk allocations for the conference participants.

4.3 Community Structure

LWA 2010 was a joint workshop week of four special interest groups of the German Computer Science Association (GI).

- **ABIS** focuses on *personalization and user modeling*.
- **IR** is concerned with *information retrieval*.
- **KDML** focuses on all aspects of *knowledge discovery and machine learning*.
- **WM**, for 'Wissensmanagement', considers all aspects of *knowledge management*.

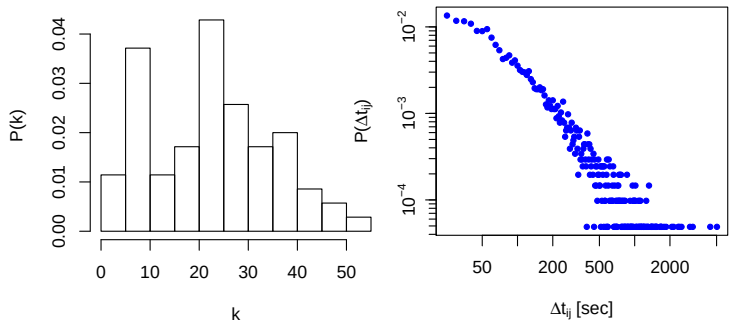


Fig. 1. Degree distribution $P(k)$ (left) and distribution of the different contact durations (right) in LWA$[\geq 0]_*$.

During the registration for LWA 2010, each participant declared his affiliation to exactly one special interest group: KDML (37), WM (16), ABIS (7), IR (10), for a total of 70 participants. Since these interest groups capture common research interests as well as personal acquaintance, the set of participants is naturally clustered accordingly.

As a first characteristic for the interest groups, we aggregated the visited talks groupwise per track. Although several sessions were joint sessions of two interest groups, Figure 2 clearly shows for each group a strong bias towards talks of the associated conference track.

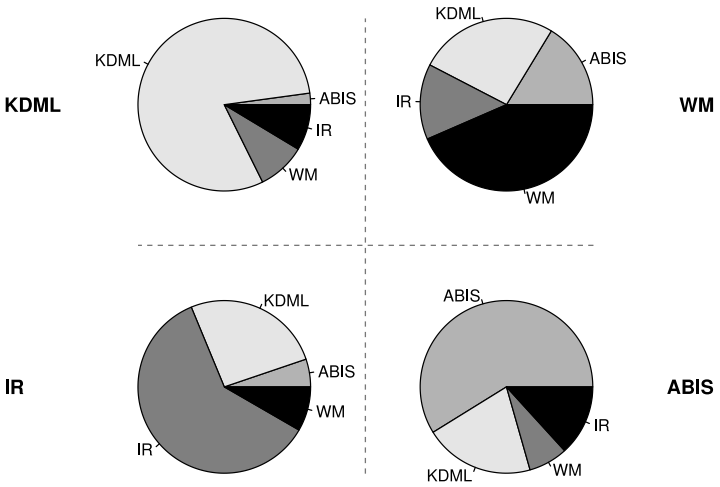


Fig. 2. Distribution of the conference tracks of the talks visited by members of the different interest groups. The top-left figure, for example, shows the distribution of tracks visited by the KDML special interest group.

The question arises, whether or not an according community structure may be observed in the contact graphs obtained during the conference. Figure 3 shows the obtained weighted and unweighted modularity scores for the contact graphs $\mathsf{LWA}[\geq i]_{\Sigma}$ and $\mathsf{LWA}[\geq i]_{\#}$ with $i = 1, \ldots, 15$, considering the interest groups as communities. We first observe that the modularity monotonically ascends with increasing minimal conversation length. This conforms to the intuition that more relevant (i. e., longer) conversations are biased towards dialog partners with common interests, as captured by the interest group membership.

For analyzing the impact of repeated or longer conversations, we calculated the weighted modularity score on the same networks, given the number of conversations or the aggregated conversation time between two participants as edge weights. Figure 3 shows that the obtained modularity scores are nearly constant across the different networks. This suggests that peers tend to talk more frequently and longer within their associated interest groups. To rule out statistical effects induced by structural properties

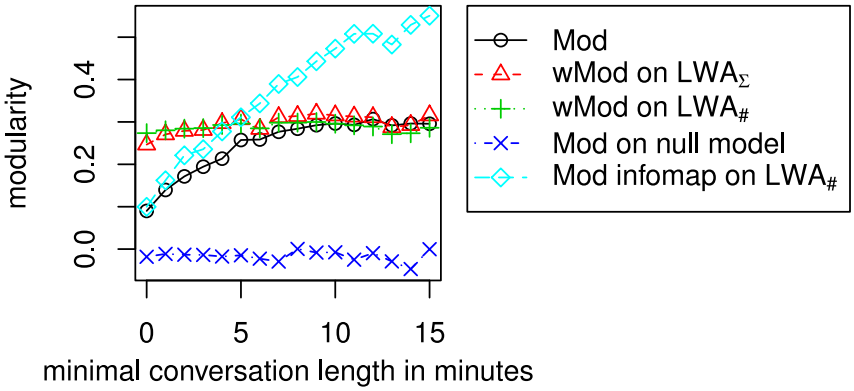


Fig. 3. Modularity score for varying minimum conversation length.

of the contact graphs, we created a null model by repeatedly shuffling the group membership of all participants and averaging the resulting unweighted modularity scores. As Figure 3 shows, the shuffled group allocation shows no community structure in terms of modularity as expected.

Additionally, the standard community detection algorithm Infomap [20] which is shown to perform well [21] was chosen for reference and applied to the same contact graphs. Figure 3 also shows the unweighted modularity scores for the obtained communities. The resulting line strictly ascends with increasing minimal conversation length. It coincides with the modularity scores of the interest group induced community structure, with parity around five minutes and nearly doubles both weighted and unweighted modularity scores in $\mathsf{LWA}[\geq 15]_{\Sigma}$ and $\mathsf{LWA}[\geq 15]_{\#}$.

Inspection of the obtained communities suggests that the applied algorithm yields communities in $\mathsf{LWA}[\geq 0]_{*}$ which are similar to the given interest groups and mines more specialized (i. e., sub communities) in $\mathsf{LWA}[\geq i]_{*}$, $i \geq 5$. Figure 4 shows for

reference in $\mathsf{LWA}[\geq 0]_*$, that ABIS and IR are nearly perfectly captured by Infomap but KDML is split mainly across two communities, one of which shared with WM. We do not aim at evaluating any community detection algorithm: We rather exemplify the application of such algorithms and approximate an upper bound of the modularity score in the contact graphs.

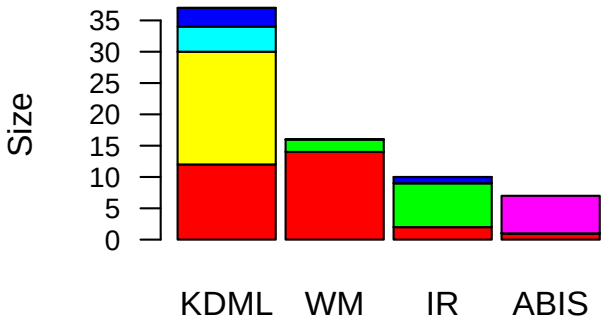


Fig. 4. Distribution of the interest groups across the six communities mined by the Infomap algorithm on $\mathsf{LWA}[\geq 0]_*$. Each color corresponds to a single (non-overlapping) community.

For analyzing social interactions across different interest groups, Table 2 shows the density in correspondingly induced sub graphs – that is, for each pair of interest groups $V_i, V_j \subseteq V$ in the complete contact graph $G = (V, E)$, the fraction of all actually realized edges in the set of possible edges between V_i and V_j.

Within the interest groups, the density values are strictly above the global density (cf., Table 1), but strictly below across different groups. This suggests that participants actually tend to interact more frequently with members of their own interest group.

Table 2. Density in the the contact graph $\mathsf{LWA}[\geq 0]_*$.

	ABIS	IR	KDML	WM
ABIS	**0.62**	0.23	0.19	0.28
IR	0.23	**0.44**	0.21	0.20
KDML	0.19	0.21	**0.38**	0.31
WM	0.28	0.20	0.31	**0.58**

4.4 Roles and Key Players

The assignment of roles to nodes in a network is a classification process that categorizes the players by common patterns. In this section, we discuss the connection between the academic status of the conference participants and the classic centrality measures and a community based role assignment. The latter was introduced in [12] together with the

Table 3. Group size and average graph centralities per academic position and for organizers and non-organizers in $\mathsf{LWA}[\geq 5]_*$: degree deg, strengths $str_\#$ and str_Σ, eigenvalue centralities eig_*, $eig_\#$ and eig_Σ, betweenness bet, closeness clo and the average community metric $rawComm$.

position/ function	$size$	deg	$str_\#$	str_Σ	eig_*	$eig_\#$	eig_Σ	bet	clo	$rawComm$
Prof.	10	7.500	16.700	11893.200	0.310	0.285	0.337	49.565	0.407	0.525
PostDoc	11	7.727	15.545	9793.364	0.303	0.213	0.198	75.973	0.419	0.675
PhD-student	33	7.152	15.091	9357.182	0.309	0.201	0.165	46.221	0.409	0.567
Student	5	3.600	12.400	6514.400	0.099	0.068	0.027	17.989	0.347	0.417
Other	6	6.667	14.333	8920.000	0.288	0.211	0.209	38.234	0.413	0.581
Organizer	11	10.000	23.727	15227.545	0.459	0.424	0.417	94.497	0.447	0.699
Non-Organizer	54	6.370	13.389	8408.056	0.256	0.162	0.144	39.565	0.397	0.542

$rawComm$ metric that it is based on. The metric is defined as the sum:

$$rawComm(u) = \sum_{v \in N(u)} \tau_u(v)\,,$$

where the function $\tau_u(v)$ assigns the contribution of a node v to connect communities of u given by

$$\tau_u(v) = \frac{1}{1 + \sum_{v' \in N(u)} \left(I(v,v') * p + \bar{I}(v,v')(1-q)\right)}.$$

In the formulas $N(u)$ is the neighborhood of a node u, $I(v,v') = 1$ if there is an edge between v and v' and 0 else; $\bar{I} = 1 - I$, p is the probability that an edge in the graph connects two communities and q is the probability that two non-linked nodes are in different communities.

The $rawComm$ score can be interpreted as a measure of how much a node connects different communities (for details, cf., [12]). To estimate the probabilities p and q we used the community clustering given by the four special interest groups.

Global Characterization Table 3 displays the average values of several graph structure metrics of $\mathsf{LWA}[\geq 5]_*$ aggregated by academic position and for the conference organizers and non-organizers (regular conference participants), respectively. Note, that while the categories referring to academic status are disjoint (the category *other* includes all participants that do not fit one of the other four) organizers and non-organizers both include participants from all the 'status' categories.

A first observation is that the organizers have significantly higher scores in all nine measures under observation. In the considered conference scenario this is highly plausible due to the nature of an organizer's job during a conference – which in the case of LWA 2010 also included the supervision and maintenance of the RFID-experiment and the CONFERATOR. Among the four academic positions, striking differences can be noticed. First of all, the student scores in all centralities are lower than those of the other

categories. We attribute this phenomenon to the fact, that students are less established in their scientific communities than scientists in higher academic positions and usually have little conference experience. This example motivates the need of social tools that assist participants in initiating contact to their communities and persons of interest.

Within the categories "Prof.", "PostDoc" and "PhD-student" the eigenvalue centralities show a particular behavior. While the unweighted eigenvalue centrality eig_* does not fluctuate much, the weighted versions eig_Σ and $eig_\#$ increase strongly from one position to the next higher one. Eigenvalue centralities are considered a measure of importance. It seems plausible, that in a contact graph among scientists, the players with longer scientific experience – including a higher and broader degree of knowledge within scientific areas and more previous contacts and collaborations with their colleagues – are considered more important and that this attitude is reflected in their contacts. The node strength measures show similar results. While the degree deg is only slightly different among the three positions, the weighted versions $str_\#$ and especially str_Σ show large differences and increase together with the position. The considerable difference between the weighted and unweighted measures can be indicates the relevance of the frequency and the length of the contacts: Professors, for example, have longer and more contacts to other participants than postdocs.

Another aspect is illustrated by the betweenness (bet) scores: Relatively to the other groups, a lot of shortest paths of $\mathsf{LWA}[\geq 5]_*$ run through nodes of PostDoc's. We attribute this to the structure of scientific institutes, where usually one professor supervises several PostDocs who again each supervise several PhD-students. PostDocs are the connection between professors and the postgraduates and thus assume the role of gatekeepers in their working environment.

Finally, for the $rawComm$ metric it is harder to come up with a plausible explanation for the difference and order of the academic positions. However, as described in [12], it can be combined with $ndeg$ – the degree divided by the maximum degree – to gain a role classification for the network's nodes in the following way: One out of four roles is assigned to a node v according to

$$role(v) := \begin{cases} \text{Ambassador} & ndeg(v) \geq s, rawComm(v) \geq t \\ \text{Big Fish} & ndeg(v) \geq s, rawComm(v) \lneq t \\ \text{Bridge} & ndeg(v) \lneq s, rawComm(v) \geq t \\ \text{Loner} & ndeg(v) \lneq s, rawComm(v) \lneq t \end{cases}$$

where s and t are thresholds that we chose as $s = t = 0.5$ – the same choice as in [12].

Ambassadors are characterized by high scores in both degree and $rawComm$ which means that they connect many communities in the graph. A *Big Fish* has contacts to a lot of other nodes, however, mostly within the same community. *Bridges* connect communities, however, not as many as ambassadors. Finally, *Loners* are those with low scores in both measures.

In the following, we investigate how nodes in their explicitly given roles like the academic position and the job (organizer) fill those implicitly given graph structure-based roles. Therefore, we applied the role classifier to the graphs $\mathsf{LWA}[\geq 0]_*$ through $\mathsf{LWA}[\geq 15]_*$ to determine – under the assumption, that longer contacts indicate more serious and scientific discussions – how this changes the community roles.

The first immediate finding is, that in none of the graphs any participant was ever classified as Big Fish, i. e., whenever a node has a high degree it also has a high $rawComm$ score. We attribute this peculiarity to the fact, that the very nature of social interaction at conferences usually is exchanging ideas with participants outside the own peer group. Especially during the LWA 2010, participants were encouraged to engage in interdisciplinary dialogue for example by including several joint sessions in the schedule and a combined event of social dinner and poster session.

The first of the three diagrams in Figure 5 displays the percentage of participants with a common academic position or job that were classified as Ambassador. The line marked with triangles displays that fraction of all participants together. The second and third diagram display the same fractions for the roles Bridge and Loner. For example in $\mathsf{LWA}[\geq 0]_*$, 40% of the professors were classified as Ambassador, 60% as Bridge and 0% as Loner. In each diagram the size of the nodes indicates the size of the group of participants with the examined position/job in the respective graph. The PhD-students, for example, are the largest section, while the students form the smallest. For $\mathsf{LWA}[\geq 5]_*$ those sizes are given in Table 3.

While all curves in Figure 5 fluctuate, there are several clearly visible tendencies. In all three diagrams, the fractions of PhD-students is very close to the fraction of all participants. The simple reason for that is, that PhD-students are the majority within the conference population and therefore dominate the general behavior. Many of the organizers start out as Ambassador or Bridge. This is again consistent with their job description. However, filtering out short contacts and thus the typical quick organizational conversations, the relevance of the organizers decreases with a higher limit to the minimum contact length. More and more organizers become Loners; in the last graph $\mathsf{LWA}[\geq 15]_*$, they are almost equally distributed among the three roles. On should keep in mind, that organizers contain persons in all academic positions. Therefore, after filtering out most of the contacts that presumably contain their organizational work, the organizers act mainly in their different role as conference participants, which might explain the stronger fluctuations in the right part of the curve.

Very consistent with the findings described above is the role distribution among the students. While in the first graphs, where short contacts dominate the longer ones, some of them are classified as Bridge or Ambassador, they quickly disappear from those roles and are classified as Loner.

Compared to the PhD-students, the fractions of the PostDocs are with few exceptions higher for the roles Ambassador and Bridge and lower for Loner. This is again consistent with the previous observations concerning the graph structure measures. Due to their greater experience PostDocs seem to have more access to colleagues in other communities. However, with the increasing filter limit, like most of the participants they become classified as loners.

Finally, the curve of the professors in the role Ambassador shows the most radical deviation from the mainstream. While in that role all other group's fractions decrease, that of the professors increases significantly up to 70% which is far more than any of the other academic positions. In summary, we observe, that the chosen method of role assignment seems to be highly correlated to the roles like academic position and the organizer job.

Characterization of Explicit Roles In the following, we aim to characterize the roles in more detail; for the dataset, we focus on the majority classes, i.e., we consider the target concept *non-organizer* concerning roles, and the target concept *PhD-students* concerning academic position. For the analysis, we applied a method for mining characteristic patterns [22] based on subgroup discovery techniques, e.g., [23]. For the data preprocessing, we first discretized the numeric features described above into three intervals (low, medium, high) using equal-width discretization.

The most descriptive factors for the role *non-organizer* are shown in Table 4 (upper). They confirm the averaged results shown above, in that the most characteristic *single* factors are given by the closeness, eigenvalue centrality, and the degree of the non-organizers, for which lower values than those of the organizers are measured.

However, if we consider combinations of factors, we observe, that there are subgroups regarding the role non-organizer for which extreme values, e.g., of the closeness together with the eigenvalue centrality yield a significant increase in characterization power, as shown by the quality increase in Table 4.

Table 4. Role = Non-Organizer / Position = PhD-student for the aggregated count information with an aggregated contact length $\geq$ 5 min. The tables show the lift of the pattern comparing the fraction of non-organizers / PhD-students covered by the pattern p compared to the fraction of the whole dataset, the size of the pattern extension (number of described non-organizers / PhD-students), and the description itself.

target	#	lift	p	size	description
Non-Organizer	1	1.06	0.88	51	clo=$\{low; medium\}$
	2	1.05	0.87	61	eig_*=$\{low; medium\}$
	3	1.04	0.86	59	deg=$\{low; medium\}$
	4	1.10	0.92	12	clo=$\{low; medium\}$ AND deg=$\{high; medium\}$
	5	1.12	0.93	30	clo=$\{high; low\}$ AND eig_*=$\{low; medium\}$
PhD-student	1	1.07	0.54	59	bet=$\{high; low\}$
	2	1.07	0.54	48	str=$\{high; low\}$
	3	1.14	0.58	26	deg=$high$
	4	1.31	0.67	12	bet=$\{high; low\}$ AND eig_*=$high$
	5	1.38	0.70	20	deg=$high$ AND bet=$\{high; low\}$
	6	1.58	0.80	10	deg=$high$ AND bet=$\{high; low\}$ AND eig_*=$\{high; low\}$

If we consider the largest group *PhD-student* (concerning the academic positions), we observe the single factors shown in Table 4 (lower), also confirming the averaged results presented above. Similarly to the non-organizers, we see that extreme values, i.e., sets of high and low values, are also very significant for distinguishing PhD students. As expected the combination with other strong influence factors increases the precision of the patterns (indicated by the *lift* parameter).

5 Conclusions

In this paper, we have presented results of an in-depth analysis of user-interaction and community structure of face-to-face contacts during a conference.

We have performed various analyses on data collected during the LWA 2010 in Kassel in October 2010 by using a social conference guiding system. We analyzed and described high-level statistics of the collected network data, examined the different communities, the roles and key players concerning these and the conference in total, and discussed various issues of user interaction.

The results of the analysis show that there is consistent community structure in the face-to-face networks, and that structural properties of the contact graphs obtained at the LWA conference reflected different aspects of interactions among participants and their position and roles.

For future work, we aim to consider the community related methods further, since communities play a central role for a social conferencing system and should allow and support emergence and evolution of community structure. Furthermore, identifying key actors according to their roles is an interesting task, e.g., being used for creating virtual sessions or recommendations.

Acknowledgements

This work has been supported by the VENUS research cluster at the interdisciplinary Research Center for Information System Design (ITeG) at Kassel University. CONFERATOR applies active RFID technology which was developed within the SocioPatterns project, whose generous support we kindly acknowledge. We also wish to thank Milosch Meriac from Bitmanufaktur in Berlin for helpful discussions regarding the RFID localization algorithm. Our particular thanks go the SocioPatterns team, especially to Ciro Cattuto, who enabled access to the Sociopatterns technology, and who supported us with valuable information concerning the setup of the RFID technology.

References

1. Wongchokprasitti, C., Brusilovsky, P., Para, D.: Conference Navigator 2.0: Community-Based Recommendation for Academic Conferences. In: Proc. Workshop Social Recommender Systems, IUI'10. (2010)
2. Atzmueller, M., Benz, D., Doerfel, S., Hotho, A., Jäschke, R., Macek, B.E., Mitzlaff, F., Scholz, C., , Stumme, G.: Enhancing Social Interactions at Conferences. it - Information Technology **53**(3) (2011) 101–107
3. Alani, H., Szomszor, M., Cattuto, C., den Broeck, W.V., Correndo, G., Barrat, A.: Live Social Semantics. In: Proc. Intl. Sem. Web Conf. (2009) 698–714
4. Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., Diot, C.: Pocket Switched Networks and Human Mobility in Conference Environments. In: Proc. 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking. WDTN '05, New York, NY, USA, ACM (2005) 244–251
5. Eagle, N., Pentland, A.S.: Reality Mining: Sensing Complex Social Systems. Personal Ubiquitous Comput. **10** (March 2006) 255–268

6. Meriac, M., Fiedler, A., Hohendorf, A., Reinhardt, J., Starostik, M., Mohnke, J.: Localization Techniques for a Mobile Museum Information System. In: Proceedings of WCI (Wireless Communication and Information). (2007)
7. Cattuto, C., den Broeck, W.V., Barrat, A., Colizza, V., Pinton, J.F., Vespignani, A.: Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks. PLoS ONE **5**(7) (07 2010)
8. Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., den Broeck, W.V.: What's in a Crowd? Analysis of Face-to-Face Behavioral Networks. CoRR **abs/1006.1260** (2010)
9. Brandes, U., Erlebach, T., eds.: Network Analysis: Methodological Foundations. In Brandes, U., Erlebach, T., eds.: Network Analysis. Volume 3418 of Lecture Notes in Computer Science., Springer (2005)
10. Chou, B.H., Suzuki, E.: Discovering Community-Oriented Roles of Nodes in a Social Network. In: Data Warehousing and Knowledge Discovery. Volume 6263 of Lecture Notes in Computer Science. Springer, Berlin / Heidelberg (2010) 52–64
11. Lerner, J.: Role assignments. In Brandes, U., Erlebach, T., eds.: Network Analysis. Volume 3418 of Lecture Notes in Computer Science. Springer, Berlin / Heidelberg (2005) 216–252
12. Scripps, J., Tan, P.N., Esfahanian, A.H.: Node Roles and Community Structure in Networks. In: Proc. 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web mining and Social Network Analysis, New York, NY, USA, ACM (2007) 26–35
13. Diestel, R.: Graph theory. Springer, Berlin (2006)
14. Gaertler, M.: Clustering. [9] 178–215
15. Newman, M.E., Girvan, M.: Finding and Evaluating Community Structure in Networks. Phys Rev E Stat Nonlin Soft Matter Phys **69**(2) (2004) 026113.1–15
16. Mitzlaff, F., Benz, D., Stumme, G., Hotho, A.: Visit Me, Click Me, Be My Friend: An Analysis of Evidence Networks of User Relationships in Bibsonomy. In: Proceedings of the 21st ACM conference on Hypertext and Hypermedia, Toronto, Canada (2010)
17. Leskovec, J., Lang, K.J., Mahoney, M.W.: Empirical Comparison of Algorithms for Network Community Detection (2010) cite arxiv:1004.3539.
18. Chin, A., Chignell, M.: Identifying Communities in Blogs: Roles for Social Network Analysis and Survey Instruments. Int. J. Web Based Communities **3** (June 2007) 345–363
19. Newman, M.E.J.: Analysis of Weighted Networks (2004) http://arxiv.org/abs/cond-mat/0407503.
20. Rosvall, M., Bergstrom, C.: An Information-theoretic Framework for Resolving Community Structure in Complex Networks. Proc. Natl. Acad. of Sciences **104**(18) (2007) 7327
21. Lancichinetti, A., Fortunato, S.: Community Detection Algorithms: A Comparative Analysis (2009) cite arxiv:0908.1062.
22. Atzmueller, M., Lemmerich, F., Krause, B., Hotho, A.: Who are the Spammers? Understandable Local Patterns for Concept Description. In: Proc. 7th Conference on Computer Methods and Systems. (2009)
23. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97). (1997) 78–87

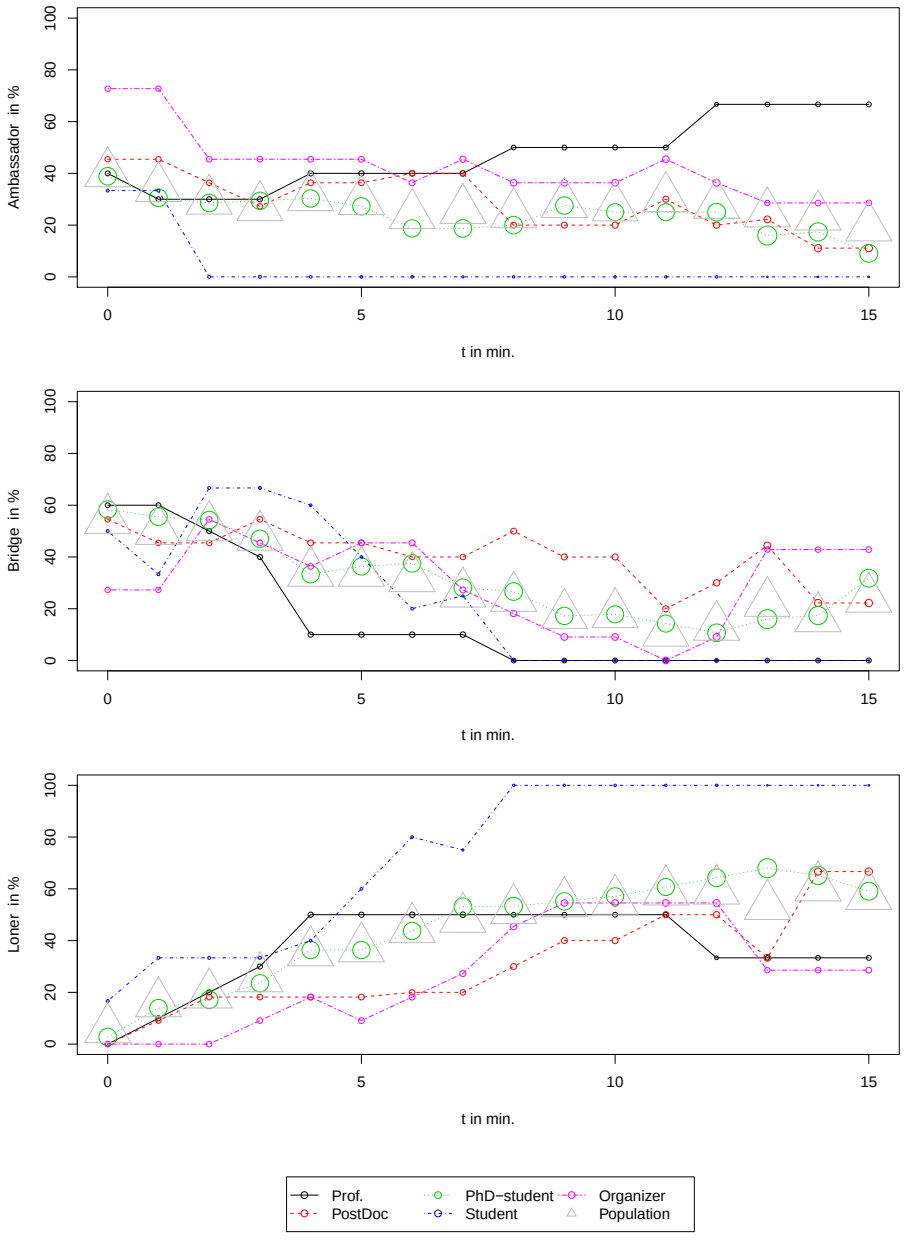


Fig. 5. Fraction of participants that assume the roles Ambassador, Bridge and Loner.

Perception and Prediction Beyond the Here and Now

Kristian Kersting

Fraunhofer IAIS,
Schloß Birlinghoven,
53754 Sankt Augustin, Germany
kristian.kersting@iais.fraunhofer.de

Abstract. Inexpensive sensors, value added data products, health records, social networks and the world-wide-web provide us with information that holds the promise to overcome longstanding temporal and spatial boundaries to human perception and prediction. Machine learning and data mining can and should play a key role to realize this dream. Unforuntately, the amount and the complexity of the data generated often presents unique computational problems in scale and interpretability. This talk illustrates that meeting the challenge is not insurmountable. Specifically, I will discuss an interpretable matrix factorization technique that is easy to implement and that scales well to billion of entries. Its usefulness will be illustrated on several web-scale datasets including 1.5 million twitter tweets, 150 million votes on World of Warcraft guilds, and a matrix of 3.3 Billion entries for phenotyping drought stress in barley from sequences of hyperspectral images. For prediction, however, asking the data only is likely to be not enough. For instance, coronary artery calcification levels in adults are likely the result of a complex web of uncertain interactions between the measured risk factors from their early adulthood. I will demonstrate that by combining probability and (subsets of) first-order logic to deal with the uncertainty and complexity respectively machines can identifying long-range, complex interactions between risk factors from data collected from the Coronary Artery Risk Developments in Young Adults (CARDIA) study. This longitudinal study began in 1985-86 and measured several risk and vital factors as well as family history, medical history, physical activity, etc. in different years (5, 10, 15, and 20) respectively for about 5000 patient.

This talks is based on joint works with Agim Ballvora, Christian Bauckhage, Jeff Carr, Marc Langheinrich, Jens Leon, Sriraam Natarajan, Lutz Pluemer, Christoph Roemer, Uwe Rascher, Albrecht Schmidt, Christian Thurau, and Mirwaes Wahabzada.

The Generation of User Interest Profiles from Semantic Quiz Games

Magnus Knuth, Nadine Ludwig, Lina Wolf, and Harald Sack

Hasso-Plattner-Institut für Softwaresystemtechnik GmbH,
Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Germany
{magnus.knuth,nadine.ludwig,
lina.wolf,harald.sack}@hpi.uni-potsdam.de
http://www.hpi.uni-potsdam.de

Abstract. Personalized web search and recommendation systems aim to provide results that match the users' personal interests and lead to a more satisfactory and effective information access. Building user profiles that reflect a large spectrum from continuous (long-term) to specific (short-term) interests is an essential task when developing personalized web applications. In this paper we present a method to generate user interest profiles without direct user interaction generated out of data sourced from quiz games played by the user. Both utilized games, *WhoKnows?* and *RISQ!* , have originally been developed as serious games with the intention to rank facts in knowledge representations as well as to find inconsistencies in a given knowledge base.

Key words: user profiles, interests, DBpedia, serious games

1 Introduction

The anyhow enormous number of information objects on a diverse range of topics on the world wide web (WWW) is continually growing. Search engines can be regarded as signposts in this information universe mandatory for any information access in the WWW today. But even with the help of search engines, the overwhelming amount of information resources being delivered as search results often simply overloads the user.

One possible way out of this information overflow is considered in progressing personalization. Ideally, the users should only be provided with information that fit to their personal information needs. Search engines apply various technologies to provide personalized search results, as e. g., user history, bookmarks, community behaviour as well as click-through rate or stickiness to a specific web page. In general, the main approaches for personalization are the reranking and filtering of search results and the development of personal recommendation systems. But, to achieve results according to the user's interests and information needs, not only statistical usage information but also explicit descriptions of the user's interests are necessary that are able to reflect a large spectrum of user interests in an interoperable way. Besides personalized search results user interest profiles

can also be used to determine the user's expertise and know-how. These profiles can be applied in social networking as well as in finding experts for specific topic areas.

In this paper we present a knowledge-based approach for the creation of user interest profiles by mining and aggregating logfile data from quiz games that shift the conventional user interrogation to an entertaining setting.

There are various approaches to collect data for building user profiles, some of them are summarized in Sec. 2. In contrast to most of the existing approaches, the use of quiz games provides valuable and sufficient reliable information about how firm a user's knowledge is on given topic areas. In Sec. 3 the two games, *WhoKnows?* and *RISQ!* are introduced that have been utilized for this research. Sec. 4 summarizes the applied ontologies and category systems used to represent the user interests, while Sec.5 explains our approach in detail. In Sec. 6 first results are shown and possible future applications are pointed out. Sec. 7 provides a brief evaluation and discussion of the achieved results and Sec.8 concludes the paper with an outlook of ongoing and future work.

2 Related Work

Personalised services demand the aggregation of user profiles that represent the interests of users adequately to fulfil their mission. Such services can be personalised and adaptive web applications, such as e. g. *Persona* [1], *PResTo!* [2] or *OBIWAN* [3], as well as recommendation systems like *Quickstep* [4] for scientific papers that apply interest profiles.

Observing user behaviour is a common way to create user interest profiles, as e. g., by web usage mining or relevance feedback, c. f. [2]. These behaviour-based approaches rely on machine learning or clustering algorithms and suffer from a cold start problem, i. e. initially there are no valid recommendations for the user that can be suggested to request feedback. Another approach to obtain user interest profiles is knowledge-based profiling that employs questionaires and interviews to acquire the users' interests, which often appears intrusive or disturbing to the user.

Though there are vocabularies to model user interests only few user profiling systems apply Semantic Web technologies. The widely used *FOAF vocabulary* [5] allows to represent interests by linking documents with the `interest` property. Based on that, the *E-foaf:interest Vocabulary*[1] provides the possibility to specify more detailed statements about interests like the period of time and the value of interest. The *Cognitive Characteristics Ontology*[2] likewise allows to specify skills, expertise and interests having a weight and time relation. The *Recommendation Ontology*[3] provides a vocabulary for describing recommendations that can be ranked and addressed to groups or agents.

[1] http://wiki.larkc.eu/e-foaf:interest, released January 2010
[2] http://smiy.sourceforge.net/cco/spec/cognitivecharacteristics.html
[3] http://smiy.sourceforge.net/rec/spec/recommendationontology.html

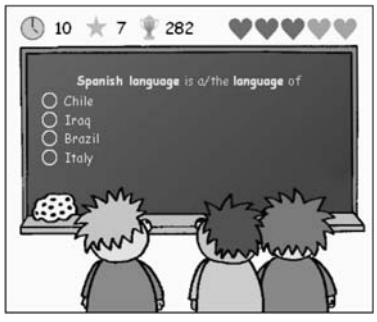
For Quickstep [4] a research topic ontology containing 32 classes has been purified from the *Open Directory Project*[4] to model user interests in the computer science domain. Research papers are classified according to classes within the topic ontology by a k-Nearest-Neighbour classifier based on the documents' term vectors, which uses a manually created training set. The user interest profiles are computed from the classifications of recently accessed research papers. A 7 % to 15 % higher topic acceptance is observed using a topic hierarchy compared to a flat topic list and a small improvement in recommendation accuracy.

Serious games have been utilized before in the area of semantic web. The games *Guess What?!* [6] and the *Virtual Pet Game* [7] are used for ontology building while the quiz game *SpotTheLink* [8] tries to align concepts from the DBpedia to the PROTON upper ontology. Serious games have also been developed to annotate images including the *ESP Game* [9], and *Phetch* [10]. Most of these games can only be played in competitive multiplayer mode, in contrast to the games utilized in this paper, where a single player can play alone. Therefore, correctly and wrongly identified questions can only be identified assuming the statements within the applied ontology are correct. The purpose of these quiz games is the ranking of properties in an existing ontology.

3 Utilized Games

We have analyzed the log files of two games, namely *WhoKnows?* [11] and *RISQ!* [12] that have been developed for relevance ranking of facts in DBpedia in order to get information about entities that are known to single players. The data is anonymized, but we managed to reassign the identity of 14 persons from our research institute that can be used for evaluation.

3.1 *WhoKnows?*



Subject	Property	Object
Chile	**language**	**Spanish language**
Iraq	language	Arabic language
Brazil	language	Portuguese language
Italy	language	Italian language

Fig. 1. Screenshot and triples used to generate a One-To-One question.

[4] http://dmoz.org/

WhoKnows? is based on the principle to present questions to the user that have been generated out of facts stemming from DBpedia RDF triples. The game has been designed to evaluate the ranking heuristics proposed in [13]. These heuristics are based on the RDF graph structure and use statistical arguments to rank RDF properties according to their relevance. Fig. 1 shows the sample question *'Spanish language is the language of ...?'* with the correct answer *'Chile'*. The question originates from the RDF triple

```
dbp:Chile dbpprop:language dbp:Spanish_language .
```

and is composed by turning the order upside down:

```
Object is the property of: subject1, subject2, subject3...
```

Fig. 1 also shows the RDF triples for the remaining choices. In addition, false answers 'Iraq', 'Brazil', and 'Italy' are randomly selected from other triples meeting the requirement that the RDF triples' subjects belong to the same or a similar category and are not related to the object used in the question.

To add variety and to increase the user's motivation, the game is designed with different game variants: *One-To-One:* only one answer is correct, *One-To-N:* one or more answers are correct, and the *Hangman* game asks to fill the correct answer in a cloze. While playing the game, the variants are used alternately. After selecting the answer, the user immediately receives feedback about the correctness of her choice. *WhoKnows?* is described in more detail in [11].

Within the log files used for this study 5,889 rounds have been played. Approximately half of the rounds have been played in Facebook[5] by 83 players, the remaining have used the anonymous standalone version[6], whereas 11 players have given an answer in 100 to 300 rounds.

3.2 *RISQ!*

RISQ! has been developed as a serious game to rank the facts about renowned persons in DBpedia. The game can be played in the social network Facebook[7] as well as standalone[8]. The flow of *RISQ!* is similar to the famous TV-show *Jeopardy!*. Questions are presented to the contestants in four different topics and three different price categories. In contrast to the original *Jeopardy!* game less topics and price levels are used in order to decrease the number of questions and increase the game speed.

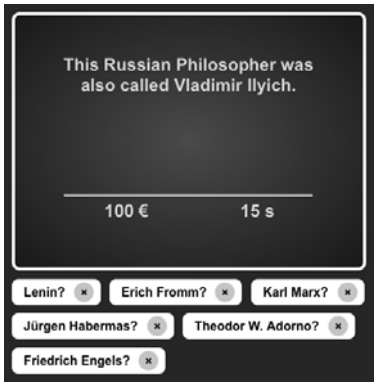
In each question a clue is presented to the player that points to the solution. The clues are constucted by using an RDF triple from DBpedia, replacing the property by a template to form a valid sentence, and replacing the solution by a category it belongs to. An example for such a hint is *'This New York State Senator was nominee of United States presidential election, 1940.'*, whose solution would be *'Franklin D. Roosevelt'*. Since automatically constructed hints

[5]

[6] http://tinyurl.com/whoknowsgame

[7] http://tinyurl.com/facebook-risq

[8] http://tinyurl.com/risqgamefb



Subject	Property	Object
Lenin	category	**Russian Philosophers**
Lenin	**altName**	**Vladimir Iliych**
Lenin	category	Philosophers
Fromm	category	Philosophers
Marx	category	Philosophers
...	category	Philosophers

Fig. 2. Screenshot and triples used to generate an active clue.

are not helpful at times and the game aims at finding the most helpful properties to identify a person, the contestants can buy additional clues with the game money.

In the orginal TV-show three contestants are playing against each other, whereas in *RISQ!* the game can be played only in single user mode so far. We introduced a timeout to prevent people from looking up the correct solution and log all player actions for later analysis.

RISQ! logged 117 unique players of which 14 has been identified as being members of our research institute. The users have played an average of 197 questions. In total 23,093 questions have been logged.

4 Entity Hierarchies

To classify the users' interests we refer to multiple entity classifications. We assume that if a user knows facts about several entities of a certain category, she is interested therein, as e. g., a player frequently answers questions about individual german soccer players correctly, it can be said she is interested in this domain, represented by Wikipedia category `GermanFootballers`, and further generalized in the `Football` domain.

The entities utilized in the games originate from DBpedia and are therefore organized in multiple hierarchical category systems, namely the DBpedia Ontology [14] and YAGO [15], and also linked to Wikipedia categories by `dc:subject` and the Freebase type system [16] via `owl:sameAs`. Each hierarchy constitutes one layer in a huge directed acyclic graph with leaf nodes containing the entities.

The DBpedia Ontology [14] is a high-level ontology, which has been manually created and contains 272 classes. Its subsumption hierarchy is comparatively shallow having a maximum depth of 6. The entities' types base on mappings of infoboxes within Wikipedia article pages to the DBpedia ontology classes.

YAGO is an automatically generated ontology based on Wikipedia and WordNet [15]. The class system is extracted from Wikipedia categories and WordNet hyponym relations, it embraces 149,162 classes.

On Wikipedia, categories are used to organize the articles, enabling users to find and navigate related articles. According to the guidelines each article should be placed in at least one category and all of the most specific ones it logically belongs to. The category system embraces more than 450,000 categories and forms a poly-hierarchie.

Freebase [16] is a collaborative database of structured knowledge having a rather lightweight type system that consists of conceptual 'topics', that are grouped in 'types', i. e. the fixed depth is 2. The type hierarchy is not determinated, hence types can be mixed independently as needed (e. g. to assign a certain property) and are created by users. Each entity is at least of type `common/topic`.

5 Method

Answering a question in a quiz game demands that the player has knowledge about that certain entity. We assume that frequently known entities are in the users scope of interest, hence the categories, whose member entities are known frequently are assumed to form a topic relevant to the user. The level of 'proven' knowledge about an entity e is represented by a numeric value, which is calculated from the number of correctly and wrongly answered questions. As shown in (1) this value gets weighted by the ratio of given facts, respectively RDF statements in the knowledge base, that have been answered. We employed a square root for the weight to reduce the impact of subjects having a great many of statements. The rated interest in e is calculated by

$$int_{e,u} = \left(\frac{factsplayed_{e,u}}{facts_e}\right)^{1/2} \cdot \frac{correct_{e,u} - wrong_{e,u}}{correct_{e,u} + wrong_{e,u}}, (-1 \leq int_{e,u} \leq 1). \quad (1)$$

The users' interest in a certain category c is determined as the mean value of interests in the entities played within this category, which gets weighted by the ratio of played entities.

$$int_{c,u} = \left(\frac{entitiesplayed_{c,u}}{|\{c/e \in c\}|}\right)^{1/2} \cdot \frac{\sum_{entitiesplayed_{c,u}} int_e}{entitiesplayed_{c,u}}, (-1 \leq int_{c,u} \leq 1). \quad (2)$$

These ratings can be specified for a single user ($int_{x,u}$) by applying only answers of user u, as well as for the whole group of users ($int_{x,\forall}$), applying all answers, which gives us an indicator for the general knowledge. Since both games are embedded in a social network, whose members considerably differ in characteristical attributes like age, gender, origin and social background, one can suppose an adequate diversity of interests within the group, who plays the games. To distinguish special user interests from general knowledge we derive the users' performance for an entity or within a category as shown in (3). By subtracting the general knowledge of the peer group, the personal impact becomes

recognizable.

$$perf_{x,u} = \frac{int_{x,u} - int_{x,\forall}}{2}, (-1 \leq perf_{x,u} \leq 1). \quad (3)$$

A positive performance value indicates a special interest of the user in x and the higher this value, the more distinctive is the knowledge of the user compared to the general knowledge within the group.

The interests are ranked according to the users interest and performance. These outcomes can be modeled using the *Cognitive Characteristics Ontology* and integrated in the users FOAF profile, exemplary we described a main interest of Magnus here:

```
ex:magnus a foaf:Person ;
    foaf:name "Magnus Knuth" ;
    cco:interest <http://dbpedia.org/resource/Greece> ;
    cco:habit [
        a cco:CognitiveCharacteristic ;
        cco:topic <http://dbpedia.org/resource/Greece> ;
        cco:characteristic cco:interest ;
        wo:weight [
            a wo:Weight ;
            wo:weight_value 0.22 ;
            wo:scale ex:AScale
            ]
        ] .
ex:AScale a wo:Scale ;
    wo:min_weight -1.0 ;
    wo:max_weight 1.0 ;
    wo:step_size 0.01 .
```

Having computed the interest for a player in each category, we also have tried to deduce the interest in entities have never been played. Therefore the value of interest in an entity e is determined as the mean value of the categories it is a member of.

$$dint_{e,u} = \frac{\sum_{c/e \in c} rating_{c,u}}{|\{c/e \in c\}|}, (-1 \leq rating_{e,u} \leq 1). \quad (4)$$

The interest value of entities that never have been played is computed by the membership categories that serve here as a common feature. Entities that are located in branches of the hierarchy that not have been played will earn the interest of the parent categories. As an extension of this it would be possible to use further common features or relationships that could be retrieved from other properties than `rdfs:typeOf` and `dc:subjectOf`.

6 Results

Table 1 shows the top and least ranked entities for the authors of this paper. Though the complete rankings could not be strictly validated, the general rank-

ings of entities appeared to the individual players, with few exceptions, entirely acceptable.

Table 1. Entity rankings

Rank	Player A	Player B
1	Gwyneth Paltrow (0.28)	Country music (0.25)
2	Ludwig van Beethoven (0.26)	Major League Baseball (0.18)
3	Sylvester Stallone (0.24)	India (0.17)
...	...	...
n-1	Duke Ellington (-0.20)	Ohio (-0.03)
n	Richard Wagner (-0.24)	Michigan (-0.10)
Rank	**Player C**	**Player D**
1	Jack Nicholson (0.37)	Pop music (0.27)
2	Sophia Loren (0.32)	Rudyard Kipling (0.24)
3	Robert De Niro (0.30)	Royal Navy (0.24)
...	...	...
n-1	Franz Marc (-0.26)	Nigeria (-0.03)
n	Gustav Mahler (-0.29)	Wolfgang Pauli (-0.14)

One advantage of semantic search is also a disadvantage: the user, who is searching for information by intitially entering a keyword needs to decide for an entity to resolve disambiguities originating from homonymous terms. To support this task we can rank the resources according to his personal interests. As e. g., looking for the programming language 'Python' this can support someone fimiliar to information science or other programming languages. In Table 2 a comparison of the rankings made for a computer scientist and for the average user is shown. Although none of these entities have ever been played within one of the games, a tendency can be observed. Unfortunately, 'Pythonydae' and 'Monty Python' lag behind, though they seem of interest in this context.

Table 2. Rankings of different entities for the term 'Python'

Rank	computer scientist	average player
1	Python (programming language)	Python (Efteling)
2	Python (mythology)	Python (roller coaster)
3	Python (Efteling)	Python (programming language)
4	Python (film)	Python (missile)
5	Python II	Python (mythology)
6	Python of Byzantium	Colt Python
7	Python Automobile	Armstrong Siddeley Python
...	...	...

7 Evaluation

To evaluate the achieved interest profiles for the identified 14 persons, they have been asked to select ten categories from each hierarchy system and to order them according to their personal interests. We received the asked for orderings back from 12 participants. At a first glance some users' self-assessment correponds quite well with the computed ranking, though for others there have been extensive differences. To compare the players' ordering with the computed list, the longest common subsequences have been computed. Therefore, we put the computed rankings in the players' stated order and extract the *longest increasing subsequence* of that permutation with a *patience sort algorithm* [17]. The longest common subsequences have an average length of 5.5, that is more than the half was ordered correctly. There was no hierarchy that performed preferably better.

It is sometimes surprising how categories are ranked, but considering the underlying entities reveals the relationships to the known entities. In subsequent interviews we could figure out, that one reason for some players' striking differences was their intensional interpretation of the category names which deviated from the extension of the category, which is used for computation. Someone might not at all be interested in the *'Rectors of the University of Edinburgh'* but still know facts about Sir Winston Churchill or Mr Gordon Brown, who are members of this Wikipedia category.

8 Summary and Outlook on Future Work

The data gathered from the games allowed us to derive players' interests in certain entities and categories. The mapping there has been straight-forward without any detours like natural language processing or machine learning.

Given the increasing importance of social semantic web it can be valuable to publish user interests within FOAF profiles automatically or give recommendations about which topics to use.

The main problem of the application is that of incomplete coverage, since the applied datasets in both games have been filtered in advance, the data of *WhoKnows?* was filtered for entities having maximum divergence in their statements while *RISQ!* comprises solely data related to persons. This partiality excludes entire domains from being reasonably rated and must be dissolved to achieve more reliable results. Therefore, we plan to extend the entity base for both games. Nonetheless, it seems reasonable to observe further kinds of relatedness than the entities categorization.

The user profiles derived with this approach reflect rather long-term interests, in combination with specific short-term interest, that e. g. originate from log file analysis, they can be purposed for personalization of semantic search, which is one objective in our project Yovisto[9], an academic video search engine.

[9] http://yovisto.com/

9 Acknowledgement

We would like to thank our students Emilia Wittmers, Eyk Kny, Sebastian Kölle, and Gerald Töpper for their outstanding contribution for *WhoKnows?* in the seminar 'Linked Open Data Application Engineering' at Hasso Plattner Institute in summer term 2010.

References

1. Tanudjaja, F., Mui, L.: Persona: A contextualized and personalized web search (2002)
2. Keenoy, K., Levene, M.: Personalisation of web search. In Mobasher, B., Anand, S., eds.: Intelligent Techniques for Web Personalization. Volume 3169 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 201–228
3. Pretschner, A., Gauch, S.: Ontology based personalized search. In: ICTAI. (1999) 391–398
4. Middleton, S., Shadbolt, N., De Roure, D.: Ontological User Profiling in Recommender Systems. In: ACM Transactions on Information Systems. Volume 22. (2004) 54–88
5. Brickley, D., Miller, L.: The Friend Of A Friend (FOAF) vocabulary specification (November 2007) `http://xmlns.com/foaf/spec/`.
6. Markotschi, T., Völker, J.: Guess What?! Human Intelligence for Mining Linked Data. Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data (KIELD) at the International Conference on Knowledge Engineering and Knowledge Management (EKAW) (2010)
7. Kuo, Y.l., Lee, J.C., Chiang, K.y., Wang, R., Shen, E., Chan, C.w., Hsu, J.Y.j.: Community-based game design: experiments on social games for commonsense data collections. In: Proceedings of the ACM SIGKDD Workshop on Human Computation. HCOMP '09, New York, NY, USA, ACM (2009) 15–22
8. Thaler, S., Simperl, E., Siorpaes, K.: Spotthelink: playful alignment of ontologies. In: Proceedings of the 2011 ACM Symposium on Applied Computing. SAC '11, New York, NY, USA, ACM (2011) 1711–1712
9. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004. (2004) 319–326
10. von Ahn, L., Ginosar, S., Kedia, M., Liu, R., Blum, M.: Improving accessibility of the web with a computer game. In Grinter, R.E., Rodden, T., Aoki, P.M., Cutrell, E., Jeffries, R., Olson, G.M., eds.: CHI, ACM (2006) 79–82
11. Waitelonis, J., Ludwig, N., Knuth, M., Sack, H.: WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia. International Journal of Interactive Technology and Smart Education (ITSE) **8**(3) (2011)
12. Wolf, L., Knuth, M., Osterhoff, J., Sack, H.: Risq! renowned individuals semantic quiz a jeopardy like quiz game for ranking facts. In: Proceedings of 7th Int. Conf. on Semantic Systems I-SEMANTICS 2011, Graz, Austria, ACM (September 2011)
13. Waitelonis, J., Sack, H.: Towards Exploratory Video Search Using Linked Data. In: Proc. of the 2009 11th IEEE Int. Symp. on Multimedia (ISM2009), Washington, DC, USA (December 2009)

14. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives:, Z.: DBpedia: a nucleus for a web of open data. In: The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea. (2008) 722–735
15. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th international World Wide Web conference (WWW 2007), New York, NY, USA, ACM Press (2007)
16. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2008) 1247–1250
17. Aldous, D., Diaconis, P.: Longest increasing subsequences: from patience sorting to the baik-deift-johansson theorem. Bulletin of The American Mathematical Society **36** (1999) 413–433

Identifying Dense Structures to Guide the Detection of Misuse and Fraud in Network Data*

Ursula Redmond, Martin Harrigan, and Pádraig Cunningham

School of Computer Science & Informatics
University College Dublin
ursula.redmond@ucdconnect.ie,{martin.harrigan,padraig.cunningham}@ucd.ie

Abstract. The identification of specific types of suspicious network structure is of interest in a number of application areas, including network intrusion detection and the analysis of mobile telecommunications data. In the work presented here, the focus is on financial transactions. We are particularly interested in scenarios where the objective is to identify misuse or fraud. We are working towards a tool that will identify examples of specific structures in networks, and provide an index of these which may be browsed through a visualization system. We present some preliminary work on identifying such dense structures and provide examples from a basic system for visualizing these in peer-to-peer lending data.

1 Introduction

This work is part of a project that has the objective of locating examples of specific types of structure contained in a network which may represent misuse or fraud. The project aims to provide a browsing facility for these structures so that they can be examined by a user in a graphical interface, in order to confirm the existence of fraud. Preprocessing the network to identify these structures allows for subsequent fast access to them for the user, although conceding the cost of an increase in storage space. This approach aims to reduce as much as possible the necessity to perform subgraph isomorphism testing for graph search, which is known to be an NP-complete problem [1].

The layout of this paper is as follows. Section 2 reviews work in the fields of fraud detection in network data, community finding and graph mining. Section 3 describes peer-to-peer lending systems which provide the data to be analyzed during the course of this project. Section 4 outlines our progress with system implementation to date. Section 5 presents some preliminary results. Section 6 concludes the paper and suggests future work.

* This work is supported by Science Foundation Ireland under Grant No. 08/SRC/I1407.

2 Related Work

This section reviews some characteristics of subgraphs which have been identified as fraudulent in different network contexts. A notable feature of these subgraphs is their high density, which relates to the community activity of fraudsters. To facilitate fast retrieval of structures of a given type from a network, a popular approach has been to enumerate frequent subgraphs, and create an index based on these. Previous work in these areas is presented in this section.

2.1 Fraud Detection in Network Data

The fan-out-fan-in (FOFI) subgraph is of interest in financial networks as it may indicate money laundering in the form of smurfing or layering [2]. The structure represents the movement of funds from a source to a destination account, with the transaction divided among an intermediate set of accounts, in order to keep each transaction under a threshold. The detection of layering is not a computationally complex problem in networks, and may be described simply. Let A be a transaction matrix, where $A_{i,j} = 1$ if there is a transaction between accounts i and j, and 0 otherwise. $(A^2)_{x,y}$ is the number of paths of length two between x and y. This analysis extends easily to longer paths, which represent more layering [3].

Some other types of network structure are commonly linked to fraudulent activity. In a telecommunications network, the accounts of fraudsters tend to be nearer to other fraudsters than the accounts of random customers are. It has been observed that not many legitimate accounts are directly adjacent to fraudulent ones, and that fraudsters do not often work on their own [4]. These facts suggest a near-clique representation for fraudulent activity. In the case of an online auction community, fraudsters may interact in small and densely connected near-cliques with the aim of mutually boosting their credibility [5]. Once an act of fraud is committed, the near-clique can be identified by the auction site and the accounts of fraudsters removed.

2.2 Dense Structure Mining

In social networks, dense network regions indicate the existence of a community. Fortunato [6] presents a comprehensive review of the broad field of community finding. The densest type of network structure is a clique, where each node connects to every other node in the clique.

There are a number of relaxations of the notion of a clique, for example, k-plexes [7], k-cores [8] and k-trusses [9,10]. *k-trusses* in particular strike a good balance between restrictiveness and tractability. A k-truss is a non-trivial connected subgraph in which every edge is part of at least $k-2$ triangles within the same subgraph. A maximal k-truss is not contained in a larger k-truss. Cohen [9] shows that a clique with k nodes contains a k-truss, although a k-truss need not contain a clique with k nodes. A k-truss contains a $k-1$-core, although a $k-1$-core need not contain a k-truss. On the tractability side, if we assume a

hash table with constant time lookups, then for a graph $G = (V, E)$, all maximal k-trusses can be enumerated in $\mathcal{O}\sum_{u \in V} deg(u)^2$-time, where $deg(u)$ is the degree of a node u. The running time can be reduced in practice by using either the maximal $k-1$-cores or the maximal $k-1$-trusses of G as input.

2.3 Frequent Structure Mining

The Apriori algorithm is extended by Vanetik *et al.* to discover typical subgraphs [11]. Subgraphs are treated as sets of paths, and the Apriori join operates on these paths. The size of the maximal independent set of instances of a subgraph is proposed as an admissible support measure, where instances must be edge-disjoint. GREW [12] is a heuristic algorithm which finds connected subgraphs with a large number of node-disjoint embeddings. The algorithms HSIGRAM and VSIGRAM [13] discover frequent subgraphs using breadth-first and depth-first search paradigms respectively. Connected subgraphs with a specified number of edge-disjoint embeddings are found in an undirected labeled sparse graph.

An index is an effective way to store mined structures. The gIndex algorithm [14] mines frequent subgraphs in the database and uses these as an index. An index of frequent subgraphs is relatively stable to database updates, and represents the data well. The FG-index [15] proposes a nested inverted-index, based on the set of frequent subgraphs. If a graph query is a frequent subgraph in the database, this index returns the exact set of query answers, with no candidate verification performed. If the query is an infrequent subgraph, a candidate answer set close to the exact answer set is produced. In this case, the number of subgraph isomorphism tests will be small, since, by construction, an infrequent subgraph will appear in few graphs in the database.

3 Peer-to-Peer Lending Systems

Given our objective of developing tools for browsing suspicious structures in networks, peer-to-peer lending systems present an interesting application area to test our tools due to the open availability of data[1].

Peer-to-peer lending and crowdfunding have emerged in recent years as the financial dimension of the social web. Initiatives in these areas offer the potential benefit of disintermediating the process of fundraising and borrowing money. Example peer-to-peer lending systems are prosper.com, lendingclub.com and zopa.com. Closely related to peer-to-peer lending is the idea of crowdfunding or crowdsourced investment funding. Two typical crowdfunding companies for startup funding are seedups.com and crowdcube.com There have also been a number of initiatives around crowdfunding for creative projects (e.g. pozible.co.uk). However, these initiatives are of less relevance here as they are altruistic rather than investment initiatives.

[1] `http://www.prosper.com/tools`.

The risk of misuse and fraud is at least as great in these new online systems as in traditional bricks-and-mortar finance. It is important to monitor the networks of transactions on these sites to check for money laundering and fraud. This is important for reasons of regulation but it is also important if users are to trust the service. Indeed for this reason prosper.com, a prominent peer-to-peer lending system, make all of their transaction data available for scrutiny.

3.1 The Prosper Lending System

Prosper opened to the public in February 1996. It was closed briefly during 2008 and again during 2009 due to regulatory issues. However, it has reported significant growth in recent months[2]. As of October 2010, there was a record of 7 451 482 bids on 379 916 listings between 1 062 266 members. Of these listings, 34 938 were accepted as loans for an average loan amount of $5 586,70. Furthermore, there were 4 058 approved groups organized into 1 566 categories. Borrowers join groups in order to improve their chances of having their listings funded; many groups enforce additional identity checks on their members and garner a certain amount of trust.

4 System Description

We used the Java programming language and yFiles [16] version 2.7 to develop our system. The network data used to test the system is derived from the Prosper dataset. The construction of the specific networks analyzed is outlined in this section, along with the types of structure sought.

4.1 Prosper Network Data

The Prosper website allows members to post a listing (describing their intentions for borrowing) or apply to sponsor a listing. A network can be constructed where nodes are members, and edges exist from lenders to borrowers who interacted via the same listing. This yields a predominantly bipartite network, with borrowers and lenders in mainly disjoint sets, although a small number of the members take on both roles.

Since members of the same set are unlikely to have links to each other, the likelihood of clique formation is low. To detect a near-clique, an effective procedure is to first detect a maximal clique and then expand it. To explore clique mining, a network can be derived consisting of only borrowers, where an edge between borrowers implies that they interacted with the same lender, and vice versa. Given a matrix A where rows and columns represent distinct lenders and borrowers, respectively, let $A_{i,j} = 1$ if there is an interaction between i and j, and 0 otherwise. Post-multiplying this matrix by its transpose will produce the matrix of borrowers. To derive a matrix of lenders, A must be pre-multiplied

[2] WSJ.com - Peer-to-Peer Loans Grow `http://on.wsj.com/kFUAuy`

by its transpose. This operation enables simpler exploration of edges between borrowers. For our purposes in the detection of cliques, edge weights are not relevant.

Attribute information associated with the members may be extracted for further analysis, such as the amount requested or supplied, as well as city and state of residence. Members who share a common interest may belong to a group. Group membership often ensures better interest rates for borrowers since lenders have confidence in trusted groups. The city and state of a group can be extracted, as well as the rating and name of the group.

4.2 Structures of Interest

The current system allows a user to browse through a set of structures, whose inclusion has been motivated by their potential association with fraudulent activity. *FOFI* subgraphs may correspond to layering in money laundering. An example is shown in Figure 3. *Maximal cliques* are the densest of dense structures. Bron and Kerbosh [17] proposed an algorithm for finding all cliques and consequently all maximal cliques (those not contained in larger cliques) in a graph. Figure 1 illustrates an example. *k-Trusses* broaden our definition of dense structure while maintaining tractability. Figure 2 shows an example.

In the complete system, cliques will be expanded into near-cliques by adding nodes and edges, subject to constraints. Such an *expanded clique* may be a *frequent*, less dense structure that exceeds a certain level of support in the overall graph, or *dense*, subject to a density criterion such as that outlined by Lee *et al.* [18].

5 Results

The network shown in Figure 1 is built from transactions from April 2009 and consists of 85 nodes and 1 676 edges. Only the borrowers are included, with links demonstrating their interaction via a lender, as described in Section 4. Perhaps due to the small number of nodes in the network, the attribute values within any given clique compared with those of the overall graph aren't particularly illuminating. For example, almost all members come from California, which lessens the discriminative power of geographical attributes.

The borrower-lender network used to illustrate the notions of k-trusses and FOFI subgraph occurs during February 2006, and contains 638 nodes and 7 154 edges. Figure 2 displays a 5-truss. Since the network is predominantly bipartite, it is noteworthy that such large quantities of triangles exist to reinforce the edges. This results from the existence of members who acted as both borrowers and lenders. It is interesting to note that this 5-truss accounts for significantly more of the money exchanged within the period than would be expected. This subgraph, which comprises only 3.76% of the members, accounts for 15.58% of the money transfer. One particular group stands out in the 5-truss. In the entire graph, 1.25% of members are in the "SF State Teaching Credentials" group.

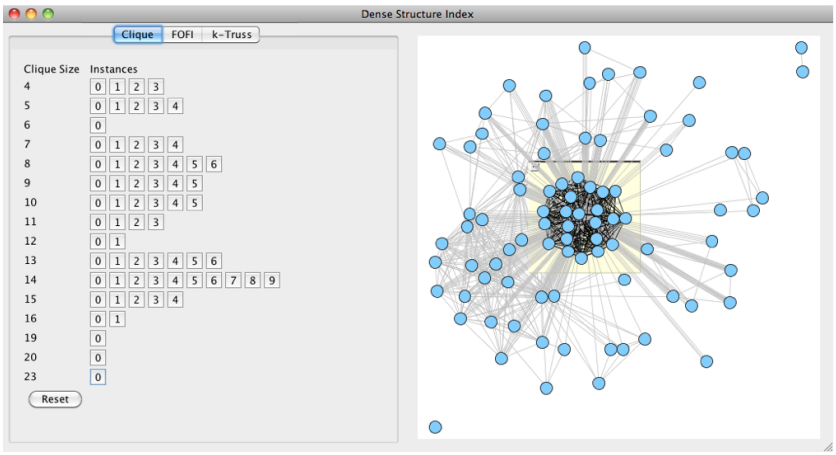


Fig. 1. The largest maximal clique in the borrower-borrower graph.

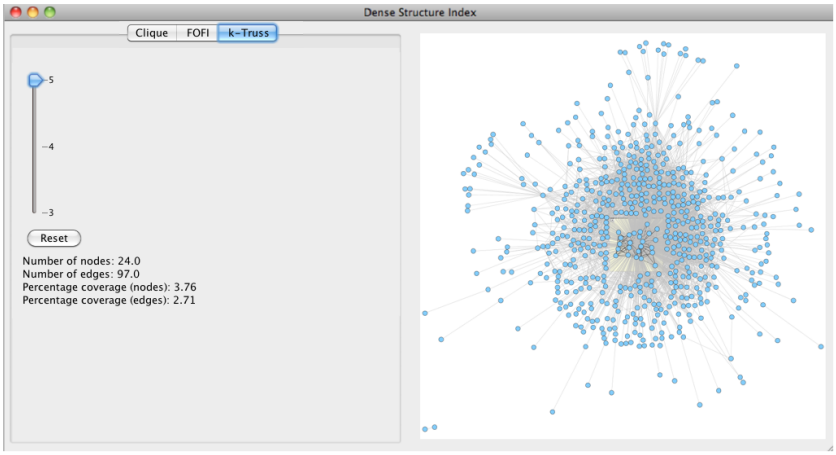


Fig. 2. The maximal 5-truss of the borrower-lender graph.

However, this group makes up 16.6% of membership of the 5-truss. Such a dense region with a high level of money transfer is remarkable. However, the increased membership from this group may indicate that there is an innocent explanation in this case.

Finally, Figure 3 shows a FOFI subgraph of size five containing a source and destination account, along with three intermediaries. This is remarkable in that the graph as a whole is predominantly bipartite so it is very unusual to have three nodes acting as both borrowers and lenders and linking source and destination accounts in this way.

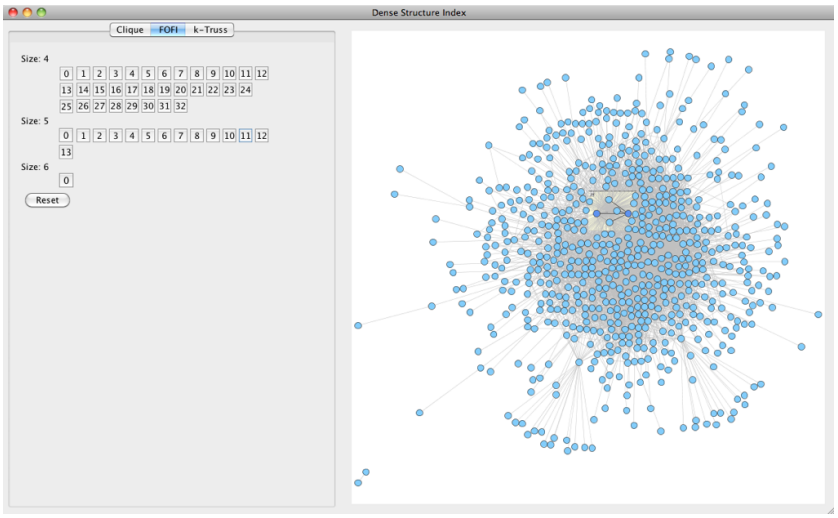


Fig. 3. A FOFI subgraph in the borrower-lender graph with three intermediate accounts.

To identify this structure a square matrix was constructed by listing all members as column indices as well as row indices, with an entry of 1 if an edge exists, and 0 otherwise. The method described in Section 2 was employed to detect layering. The subgraphs returned were filtered so as to only preserve those with the most suspicious directionality. It was required that an edge from the source pointed to each intermediate node, and that an edge from each intermediate node pointed to the target.

6 Conclusions and Future Work

In this paper, the Prosper peer-to-peer lending network has been described from the point of view of fraud detection. Using techniques from graph mining, the development of a system to examine this network data has been discussed. Dense structure has been identified and visualized, facilitating the exploration of structures with a noteworthy distribution of attribute values in some cases.

Following on from this work, a more comprehensive method of indexing suspicious structures will be developed. The visualization system will be altered accordingly, so as to provide an intuitive means of exploring relevant structures.

References

1. Garey, Michael R., Johnson, David S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)

2. Reuter, P., Truman, E.M.: Chasing Dirty Money: The Fight Against Money Laundering. Peterson Institute (2004)
3. Lu, L., Zhou, T.: Link Prediction in Complex Networks: A Survey. Physica A: Statistical Mechanics and its Applications (2010)
4. Cortes, C., Pregibon, D., Volinsky, C.: Communities of Interest. Lecture Notes in Computer Science **2189** (2001) 105–114
5. Pandit, S., Duen Horng Chau, Wang. S., Faloutsos, C.: NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks. In: Proceedings of the 16^{th} International Conference on World Wide Web. (2007)
6. Fortunato, S.: Community Detection in Graphs. Physics Reports **486**(3–5) (2010) 75–174
7. Seidman, S., Foster, B.: A Graph-Theoretic Generalization of the Clique Concept. Journal of Mathematical Sociology **6** (1978) 139–154
8. Seidman, S.: Network Structure and Minimum Degree. Social Networks **5**(3) (1983) 269–287
9. Cohen, J.: Graph Twiddling in a MapReduce World. IEEE Computing in Science & Engineering **11**(4) (2009) 29–41
10. Cohen, J.: Trusses: Cohesive Subgraphs for Social Network Analysis. Technical report, National Security Agency (2008)
11. Vanetik, N., Gudes, E., Shimony, S. E.: Computing Frequent Graph Patterns from Semistructured Data. In: Proceedings of the International Conference on Data Minning. (2002) 458–465
12. Kuramochi, M., Karypis, G: Grew: A Scalable Frequent Subgraph Discovery Algorithm. In: Proceedings of the IEEE International Conference on Data Mining. (2004)
13. Kuramochi, M., Karypis, G.: Finding Frequent Patterns in a Large Sparse Graph. Data Mining and Knowledge Discovery **11** (2005) 243–271
14. Yan, X., Yu, P., Han, J.: Graph Indexing: A Frequent Structure-Based Approach. Proceedings of the ACM SIGMOD International Conference on Management of Data (2004) 335–346
15. Cheng, J., Ke, Y., Ng, W., Lu, A.: FG-Index: Towards Verification-Free Query Processing on Graph Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. (2007) 857–872
16. yWorks: yFiles – Java Graph Layout and Visualization Library (2011)
17. Bron, C., Kerbosch, J.: Algorithm 457: Finding All Cliques of an Undirected Graph. Communications of the ACM **16**(9) (1973) 575–577
18. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting Highly Overlapping Community Structure by Greedy Clique Expansion. KDD SNA (2010)

A Data Generator for Multi-Stream Data

Zaigham Faraz Siddiqui†, Myra Spiliopoulou†,
Panagiotis Symeonidis*, and Eleftherios Tiakas*

University of Magdeburg†; University of Thessaloniki*.
[siddiqui,myra]@iti.cs.uni-magdeburg.de; [symeon,tiakas]@csd.auth.gr

Abstract. We present a stream data generator. The generator is mainly intended for multiple interrelated streams, in particular for objects with temporal properties, which are fed by dependent streams. Such data are e.g. customers their transactions: learning a model of the customers requires considering the stream of their transactions. However, the generator can also be used for conventional stream data, e.g. for learning the concepts of the transaction stream only.
The generator is appropriate for testing classification and clustering algorithms on concept discovery and adaptation to concept drift. The number of concepts in the data can be specified as parameter to the generator; the same holds for the membership of an instance to a class. Hence,it is also appropriate for synthetic datasets on overlapping classes or clusters.

1 Introduction

Most of the data stored in databases, archives and resource repositories are not static collections: they accumulate over time, and sometimes they cannot (or should not) even be stored permanently - they are observed and then forgotten. Many incremental learners and stream mining algorithms have been proposed in the last years, accompanied by methods for evaluating them [3, 1]. However, modern applications ask for more sophisticated stream learners than can currently be evaluated on synthetically generated data. In this work, we propose a generator for complex stream data that adhere to multiple concepts and exhibit drift. The generator can be used for the evaluation of (multi-class) stream classifiers, stream clustering algorithms over high-dimensional data and relational learners on streams.

Our generator is inspired by recommendation engines, where data are essentially a combination of static objects and adjoint streams: people rank items - the rankings constitute a *fast* (or conventional) stream; new items show up, while old items are removed from the provider's portfolio - the items constitute a *slow* stream; new users show up, while old users re-appear and rank items again, possibly exhibiting different preferences as before - users also constitute a *slow* stream. In [4] we devised the term *perennial objects* for a stream of objects that appear more than once and may change properties. In conventional relational learning, these objects would be static and have one fixed label, while in relational stream learning they may have different labels at different times. In [4] we proposed a decision tree stream classifier for a stream of perennial objects.

Our generator extends the generator of static item recommendations presented in [6] in two ways. First, our generator builds a *stream* of ratings, hence it can be used to test recommenders designed for dynamic data. Second, and most importantly, our generator builds the ratings' stream upon a synthetic set of evolving profiles, thus allowing the evaluation of learners designed for complex dynamic environments. In particular, consider an application that categorizes customers into classes A, B, C (A is best), given their response to recommendations and considering some demographic attributes (e.g. having children, having a car etc). What synthetic data are needed to evaluate a learner for this application? A synthetic set (or stream) of ratings is not sufficient, because it does not contain customers (and their labels). Next to the stream of ratings, we need a set that describes the customers and associates them to their ratings in a *non-random* way. Further, since customer preferences may change, concept drift must be incorporated into the relationship between customers and their ratings. Our generator is designed for this kind of multi-relational (stream) learning.

The core idea of our generator is as follows. The preference of a user towards some item(s) defines its behaviour. Multiple users' exhibiting similar behaviour can be grouped/categorised together as a single user profile. Conversely to learning task where these profiles are learned from among a group of users, the generator first creates these user profiles which serve as a prototypes. These profiles are then used to generate individual user data according the item preferences stored in them. Noise can be imputed to the data by forcing a user to rank in discordance to her profile with some probability. Drift is imputed to the data by allowing a profile to exist only for some timepoints and then forcing it to mutate to one or more profiles with some probability.

The paper is organized as follows. In Section 2 we give a formal description of the problem along with the related work. We explain the multi-relational generator in Section 3 with some results in Section 4. We summarise and discuss future improvements in Section 5.

2 Problem Specification and Background Literature

In this section we explain the learning task over the multiple interrelated streams (i.e., slow and fast streams) in more detail, which our generator is going to simulte. In subsection 2.2 we also discuss the studies that address similar problems.

2.1 Problem Specification

In our introductory example, we considered a slow stream of users $\mathcal{T}$ and fast ranking/transaction stream $\mathcal{S}^1$ that feeds the user stream (i.e., the users') with rantings/purchases from a slow stream of items $\mathcal{S}^2$. Stream of user is referred as *target* streams as the learning task concerns solely $\mathcal{T}$, e.g. finding groups of users that show similar behaviour when rating/purchasing different items, predicting whether a user will like a certain item (Y) or not (N), or labelling users on their "lifetime value" for the company (typically in four classes A, B, C, D where A

is best and D is worst). The contents of the inter-connected streams (i.e., the ranking stream and the item stream via ranking stream) should be taken into account when building the classifier.

Learning on a Stream of Perennial Objects. The stream of perennial objects (i.e., slow stream) $\mathcal{T}$ exhibits three properties that are atypical for streams. Differently from conventional stream objects that are seen, processed and forgotten, objects of $\mathcal{T}$ may not be deleted: an examinations office may file away the results of a successful exam, but does not file away the students who passed the exam; a product purchase may be shifted to a backup medium after completion, but the customer who did the purchase remains in the database. One can even argue that $\mathcal{T}$ is not a stream at all. However, it is obvious that new objects arrive (new customers, new students), while old objects are filed away after some time (e.g. students who completed their degree and customers who have quitted the relationship with the company). We use the term *perennial objects* or *stream of perennial objects* for the slow stream $\mathcal{T}$.

Second, the objects in $\mathcal{T}$ may appear several times, e.g. whenever the properties of a user (e.g. her address) change and whenever this user is referenced by a fast stream (i.e. when the user assigns a new rating or purchases a new item). Third, the label of a $\mathcal{T}$ object may change over time: a user who earlier responded positively towards and item (Y) may stop doing so (label becomes N); a B-user may become an A-user or a C-user. Hence, the label of a perennial objects x is not a constant; at every timepoint t, at which x is observed, its label is $label(x, t)$. The learning task is to predict this label at t, given the labelled data seen thus far and given the streams that feed $\mathcal{T}$.

2.2 Related Work

Our generator is inspired from the properties of the perennial stream and builds on a generator for recommender systems by Symeonidis et al.[6]. This generator is intended for learning in a static context. We outline it here briefly.

The generator of [6] produces a unipartite user-user (friendship) network and a bipartite user-item rating network. In contrast to purely random (i.e., Erdos-Renyi) graphs, where the connections among nodes are completely independent random events, the synthetic model ensures dependency among the connections of nodes, by characterizing each node with a m-dimensional vector with each element a randomly selected real number in the interval [-1,1]. This vector represents the initial user profile used for the construction of the friendships and ratings profiles, which are generated as follows:

- For the construction of the friendship network, two nodes are considered to be similar and thus of high probability to connect to each other if they share many close attributes in their initial user profile. Given a network size N and a mean degree k of all nodes, the generator starts with an empty network with N nodes. At each time step, a node with the smallest degree is randomly

selected (there is more than one node having the smallest degree). Among all other nodes whose degrees are smaller than k, this selected node will connect to the most similar node with probability $1 - p$, while a randomly chosen one with probability p. The parameter $p \in [0, 1]$ represents the strength of randomness in generating links, which can be understood as noise or irrationality that exists in almost every real system.
- For the construction of the user-item rating network, the generator follows a similar procedure. It uses the following additional parameters as well: (i) the ratings range, (ii) the mean number of rated items by all users. Notice that each user can rate different items from others and has in his profile a different number of rated items, following the power law distribution.

xSocial is a multi-modal graph generator that mimics real social networking sites to produce simultaneously a network of friends and a network of their co-participation [2]. In particular, xSocial consists of a network with N nodes, each of which has a preference value $calf_i$. At each time, every node performs three independent actions (write a message, add a friend and comment on a message). A node chooses his friends either by their popularity of by the number of messages on which they have commented together, which is determined by his preference $calf_i$. A node can also follow the updated status of his friends by putting comments on the corresponding newly written messages.

The generator of Symeonidis et al., [6] uses both structural (friendship network) and content-based information (item-rating network) for making recommendations to the users. However, it is only suitable for static learning. xSocial, on the other hand, simulates the temporal/stream-based problem but it only uses the structural information (i.e., networks of friends and their interactions) for making recommendations to the users. Different from [6], our generator aims to alleviate the problem of static recommendations by making use of the dynamic ratings profile (i.e., a user's rating preferences may change overtime), and unlike xSocial [2] its primary focus is on content-based recommendations.

3 Generating Profiles & Transactions with Concept Drift

Our generator is inspired by the idea of predicting user ratings in a recommendation engine, and builds upon the generator of [6] (c.f. Section 2.2). In particular, our generator creates data according to the following scenario:

> Each user adheres to a *user profile*, while each item adheres to an *item profile*[1]; the user profiles correspond to classes. The rating of a user u for an item i depends upon affinity/preference the of the user profile of u towards the item profile of i; ratings are generated at each timepoint t. At certain timepoints, user profiles mutate, implying that the ratings of the users for the items change. An example of learner object for some

[1] just as user profiles serve as prototypes for the generation of concrete user data (c.f. Section 1), item profiles serve the same purpose

Table 1. Parameters of the generator.

Param	Description
P^i	set of item profiles with $N^i = \|P^i\|$ number of profiles
P^u	set of user profiles with $N^u = \|P^u\|$ number of profiles
n^i	number of items per item profile
n^u	number of users per user profile
v^i	number of synthetic variables that describe an item profile
v^u	number of synthetic variables that describe a user profile
τ_d	number of drift levels across the time axis
A_d^u	number of active user profiles at drift level d
$\mathcal{L}$	max lifetime of a drift level as number of timepoints
R	max number of items rated by a user at any timepoint
$\phi_{\mathcal{U}\to\mathcal{I}}$	the probability of a user profile $\mathcal{U}$ selecting an item from item profile $\mathcal{I}$ for rating.

Item Profiles	Var 1	Var 2	Var 3	Var 4
IP1	23 ±4	52 ±9	97 ±2	8 ±9
IP2	2 ±9	1 ±7	46 ±3	91 ±6
IP3	72 ±7	71 ±3	26 ±2	52 ±1
IP4	3 ±4	2 ±4	27 ±5	25 ±3

Fig. 1. Sample item profiles with $v^i = 4$ synthetic variables. The mean and variance associated with each variable are used to generate items according to the normal distribution.

learner is to predict the profile of a user at a given timepoint, when provided with the users' ratings data.

In more detail, our generator takes as input the parameters depicted in Table 1, and described in sequel. It generates: item profiles and from them items; user profiles and from them users; and ratings of users for items at each timepoint. A user profile may live at most $\mathcal{L}$ timepoints before it mutates.

Generation of item profiles and items.

Item profiles are described by v^i synthetic variables. The generator creates a set P^i with N^i item profiles and stores for each one the mean and variance of each of the v^i variables. Next, each of these item profiles is used as prototype for the generation of n_i items, producing $n^i \times N^i$ items in total. Items also adhere to the v^i variables; the value of each variable in an item adhering to profile $\mathcal{I}$ is determined by the mean and variance of this variable in the profile $\mathcal{I}$. The description of item profiles and is depicted in Figure 1.

The number of items considered (rated) at some timepoint may vary from one timepoint to the next, but there is no bias towards items of some specific profile(s). Hence, item profiles are not exhibiting concept drift.

User Profiles	Var 1	Var 2	Var 3	Item Profile Probabilities			
				IP1	IP2	IP3	IP4
UP1	13 ±0	22 ±5	51 ±1	0.1	0.6	0.25	0.05
				20 ±5	50 ±8	80 ±9	29 ±1
				IP1	IP2	IP3	IP4
UP2	34 ±4	55 ±0	68 ±9	0.7	0.1	0.1	0.1
				90 ±2	25 ±5	93 ±4	9 ±1
				IP1	IP2	IP3	IP4
UP3	21 ±5	98 ±4	1 ±5	0.2	0.5	0.1	0.2
				42 ±2	95 ±2	10 ±0	12 ±5

Fig. 2. Sample user profiles with mean and variance for synthetic variables with probabilities of selecting an item from a certain item profile (row 1) and mean and variance of the rating that item (row 2), where $v^u = 3$.

Generation and transition of user profiles.

User profiles are described by a set of parameters v^u. These are synthetic variables. The generator creates a set P^u with N^u user profiles and stores for each one the mean and variance of each variable in v^u. User profiles serve as templates for the generation of users, in much the same way as item profiles are used to generate items. However, there are two main differences. First, user profiles are subject to transition, and not all of them are active at each drift level $d = 1, \ldots, \tau_d$. Second, a user profile exhibits affinity towards some item profiles, expressed through the probabilities between the item profile and the user profile. The description of user profiles is depicted in Figure 2.

The affinity of user profiles towards item profiles manifests itself in the user ratings: a generated user adheres to some user profile and rates items belonging to the item profile(s) preferred by her user profile. The affinity $\phi_{\mathcal{U}\to\mathcal{I}}$ is defined as the probability of a user profile $\mathcal{U}$ selecting an item from item profile $\mathcal{I}$ for rating. The probability $\phi_{\mathcal{U}\to\mathcal{I}}$ is controlled by a user-defined global parameter $UP2IP \in [0, 1]$. If $UP2IP$ is close to zero, user profiles show strong affinity towards a certain item profile while if the value is closer to 1, the probabilities are initialised randomly.

At each drift level $d = 1, \ldots, \tau_d$, only a subset of user profiles $A^u_d \subseteq P^u$ are active[2]. The active profiles at each drift level is determined at the beginning. For drift level $d > 1$, the generator maps the profiles A^u_{d-1} of level $d - 1$, to the new profiles A^u_d of level d on similarity, i.e. the transition probability from an old to a new profile is a function of the similarity between the two profiles. The result is a *profile transition graph*, an example of which is depicted in Figure 3. This graph is generated and then the thread of each profile is recorded for inspection.

The coupling of profile transition to the profile similarity function ensures that profile mutation corresponds to a gradual drift rather than an abrupt shift. The extent of profile mutation is further controlled by a user-defined global variable $\in [0, 1]$ that determines the *true preference* of an old user profile for a

[2] the number of active profiles at each drift level d is an integer and calculated using $\frac{N^u}{d}$ and is similar for each drift level

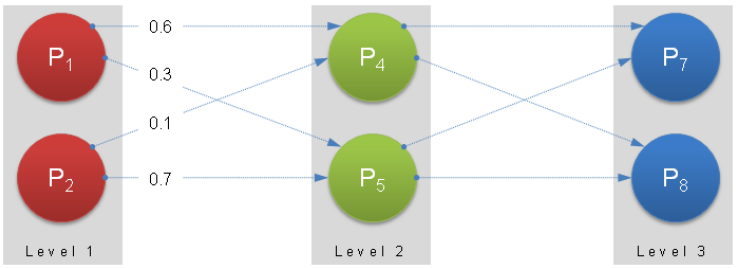


Fig. 3. A profile transition graph; each column corresponds to a timepoint, indicating that the number of profiles/classes may change from one timepoint to the next (transition probabilities between level 2 and level 3 have been omitted)

new user profile. A value close to zero means that the most similar new profile will always be preferred. Larger values allow for a weaker preferential attachment, while a value close to 1 means that the new profile is chosen randomly, and the transition is essentially a concept shift rather than a drift (a "drift" is a change of gradual nature, e.g. when a profile mutates into a similar one; while a "shift" is a more drastic and abrupt change, e.g. a profile being *replaced* by another one). The similarity between two user profiles $\mathcal{U}$ and $\mathcal{U}'$ is defined in Equation 1.

$$sim(\mathcal{U},\mathcal{U}') = \sqrt{\sum_{\mathcal{I}\in P^i} (\phi_{\mathcal{U}\to\mathcal{I}} - \phi_{\mathcal{U}'\to\mathcal{I}})} \tag{1}$$

where $\phi_{\mathcal{U}\to\mathcal{I}}$ is the prob. of rating an item from profile $\mathcal{I}$ for $\mathcal{U}$, $\mathcal{U} \in A^u_d$ and $\mathcal{U}' \in A^u_{d+1}$.

The affinity of user profiles towards item profiles manifests itself in the user ratings: a generated user adheres to some user profile and rates items belonging to the item profile(s) preferred by her user profile. Affinity is also affected by profile transitions. Once a user profile $\mathcal{U}$ mutates to $\mathcal{U}'$, all its users adhere to the $\mathcal{U}'$ profile: they prefer the item profiles to which $\mathcal{U}'$ shows affinity, and rate items adhering to these item profiles.

Generation of users and ratings.

For each user profile $\mathcal{U} \in P^u$, the generator creates n^u users. As for items, users adhere to the set of parameters v^u as user profiles; the value of each parameter in a user adhering to profile $\mathcal{U}$ is determined by the mean and variance of this variable in the profile $\mathcal{U}$.

The profiles of each drift level d exists for at most $\mathcal{L}$ timepoints, before profile transition occurs; the lifetime of a profile is chosen randomly. At each of these timepoints, the generator creates ratings for all users in each active profile. For each user profile $\mathcal{U}$ and user u adhering to $\mathcal{U}$, and for each item profile $\mathcal{I}$ is selected based on the probability $\phi_{\mathcal{U}\to\mathcal{I}}$. An item i is randomly chosen from the

$\mathcal{I}$ and a rating value is generated based on mean and variance in $\mathcal{U}$ for rating (c.f. Figure 2). A user can rank at most R items per timepoint.

4 Working with the Data Generator

We have used our generator to evaluate the performance of the classification rule miner (CRMPES) [5] which is incorporated into a tree induction algorithm (TrIP) [4]. CRMPES is used to generate new richer attributes that exploit dependency between attributes by using classification rules which otherwise are ignored by the TrIP. To test CRMPES, we developed the generator with user and item profiles. The purpose was to study how CRMPES adapts to complex patterns (spanning multiple attributes) in the presence of concept drift. The details on the experiments can be found in the paper [5].

5 Conclusions

We presented a multi-stream generator that has been inspired from the domain of recommendation system. It generates ratings data for users according to user profiles. With time the profiles mutates into newer ones. The mutation can be adjusted to simulate drastic shifts as well more gradual drifts. The generator can be used for evaluating supervised and unsupervised learning task for discovering and adaptation to concept drift.

Acknowledgements: Work of the first author was partially funded by the German Research Foundation project SP 572/11-1: IMPRINT: Incremental Mining for Perennial Objects. The cooperation between the two research groups is supported by the DAAD project travel grant 50983480.

References

1. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. KDD '09, pages 139–148. ACM, 2009.
2. N. Du, H. Wang, and C. Faloutsos. Analysis of large multi-modal social networks: Patterns and a generator. ECML/PKDD '10, pages 393–408. Springer, 2010.
3. J. Gama, R. Sebastião, and P. P. Rodrigues. Issues in evaluation of stream learning algorithms. KDD '09, pages 329–338. ACM, 2009.
4. Z. F. Siddiqui and M. Spiliopoulou. Tree induction over perennial objects. SSDBM'10, pages 640–657. Springer-Verlag, 2010.
5. Z. F. Siddiqui and M. Spiliopoulou. Classification rule mining for a stream of perennial objects. RuleML@IJCAI '11. Springer-Verlag, 2011.
6. P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. RecSys '10, pages 183–190. ACM, 2010.