Network Regression with Predictive Clustering Trees

Daniela Stojanova[1], Michelangelo Ceci[2], Annalisa Appice[2], Sašo Džeroski[1]

[1]: Jožef Stefan Institute, Slovenia
[2]: Università degli Studi di Bari, Italy

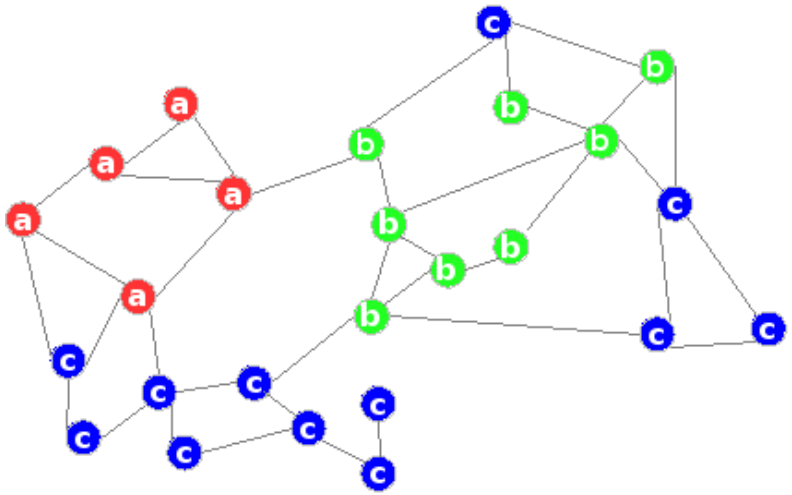
Google™ We would like to thank Google for funding the travel grant that enabled the first author to attend the conference and present the paper

UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

Mining network data

- Social networks, financial transaction networks, sensor networks, communication networks, ... are ubiquitous



Autocorrelation

- Autocorrelation (AC): the values of the response variable at a given node depend on the values of the variables (predictor and response) at the nodes connected to the given node
- In network data the Homophily's principle holds [McPherson et al., 2001], i.e. the tendency of nodes with similar values to be linked each other
- The major difficulty due to the autocorrelation is that the independence assumptions (i.i.d.), which typically underlies machine learning methods, are no longer valid.
  - The violation of the instance independence has been identified as the main responsible of poor performance of traditional machine learning methods [Neville et al. , 2004]

Autocorrelation

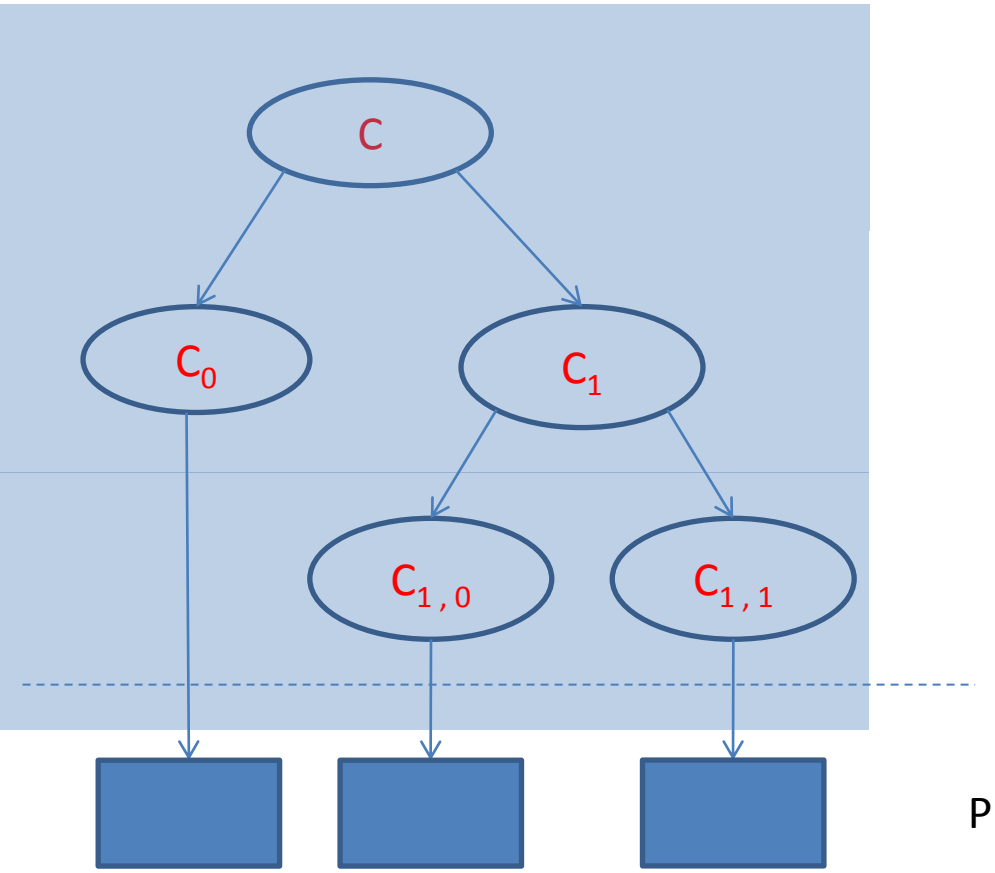
- Recently, there has been a number of methods that consider the autocorrelation phenomenon in different forms:
  - Relational
  - Spatial
  - Temporal
- One limitation of models that represent and reason with global autocorrelation is that the methods assume the autocorrelation dependencies are stationary throughout the relational data graph.
  - Exceptions in collective classification, but for classification tasks

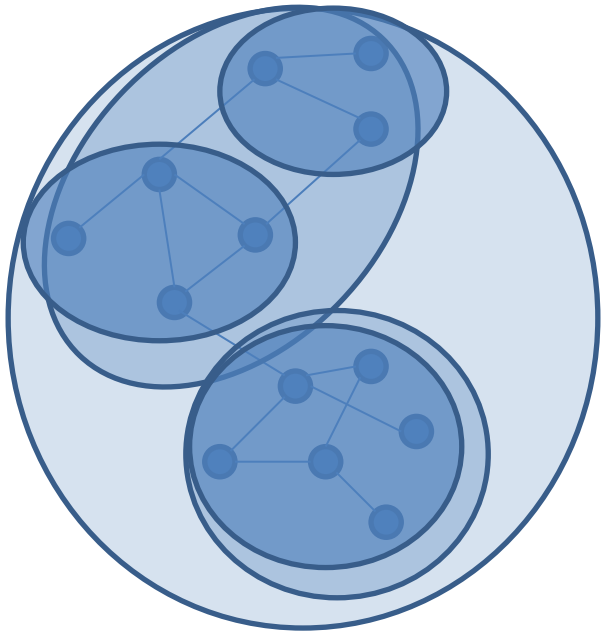
- Goal of this work is to model non-stationary autocorrelation in a relational data network when mining regression models.

The Basic Idea

- We develop an approach to modeling non-stationary autocorrelation in network data by using predictive clustering.
  - Predictive clustering combines elements from both prediction and clustering.
  - As in clustering, clusters of examples that are similar to each other are identified, but
  - A predictive model is associated to each cluster.
- Clustering is based on autocorrelation: each cluster should contain highly autocorrelated entities.

The Basic Idea: learning NPCTs





Predictive models

Measures of Network Autocorrelation

- Global Moran's I (spatial AC)

$$I_Y = \frac{N \sum_i \sum_j w_{ij}(Y_i - \overline{Y})(Y_j - \overline{Y})}{\sum_i \sum_j w_{ij} \sum_i (Y_i - \overline{Y})^2}$$

where $W=[w_{ij}]$ is the weighting matrix

Euclidean	$w_{ij}=1-{}^{d_{ij}}/_{b}$
Modified	$w_{ij}=1-{}^{d_{ij}{}^2}/_{b^2}$
Gaussian	$w_{ij}=e^{(-\frac{d_{ij}{}^2}{b^2})}$

- Randić Connectivity Index (CI)

$$\chi = \sum_{edges\ ij} \frac{1}{\sqrt{D(i)D(j)}}$$

where $D(i)$ and $D(j)$ represent the weighted node degree vector

- Relational AC Coefficient (P)

$$P_Y = \frac{\sum_{ij\ s.t.\ (u_i,u_j)\in P_R}(Y_i - \overline{Y})(Y_j - \overline{Y})}{\sum_{ij\ s.t.\ (u_i,u_j)\in P_R}(Y_i - \overline{Y})^2}$$

where P_r represents the set of related pairs

Top down induction of NPCTs

Algorithm 1. Top-down induction of NetworkPCTs

1: **procedure** NetworkPCT(A) **returns** tree
2: **if** stop(A) **then**
3: **return** leaf(Prototype(A))
4: **else**
5: $(c^*, h^*, \mathcal{P}^*) = (null, 0, \emptyset)$
6: **for each** possible test c **do**
7: $\mathcal{P}$ = partition induced by c on A
8: $h = \frac{\alpha}{|Y|} \sum_{\mathbf{Y}} \Delta_Y(A, \mathcal{P}) + \frac{(1-\alpha)}{|Y|} \sum_{\mathbf{Y}} S_Y(A, \mathcal{P})$
9: **if** $(h > h^*)$ **then**
10: $(c^*, h^*, \mathcal{P}^*) = (c, h, \mathcal{P})$
11: **end if**
12: **end for**
13: **for each** $A_k \in \mathcal{P}^*$ **do**
14: $tree_k$ = NetworkPCT(A_k)
15: **end for**
16: **return** node(c^*, $\bigcup_k \{tree_k\}$)
17: **end if**

Δ_Y variance reduction (N)PCTs

S_Y network autocorrelation NPCTs

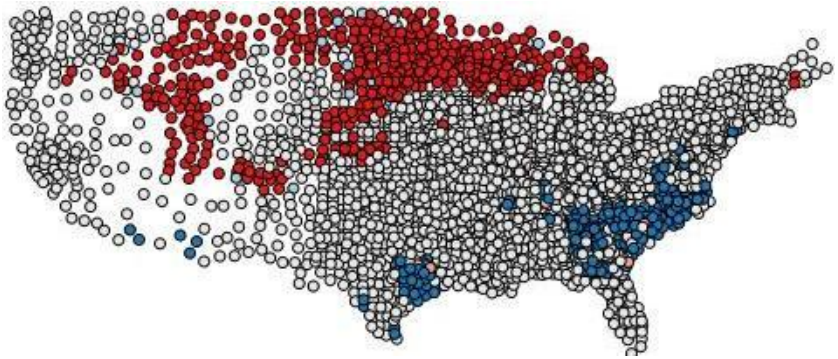
Learning NPCTs: practical aspects

- Splitting criterion on induction of PCTs modified to be
  - Linear weighted function of the variance reduction & AC measures
- Stopping criterion
  - Minimum number examples in leaf $\sqrt{N}$ (Gora and Wojna, 2002)
- Estimating the bandwidth *b* (size of neighborhood in AC)
  - Minimize the leave-one-out CV RMSE error of linear model
  - Using the Golden ratio search
  - This parameter is critical since it
    - controls the degree of smoothness
    - influences the level of the autocorrelation
- Time complexity of learning PCT in this way is $O(z * m * NlogN)$ where

  z is number of internal nodes in the tree
  m is number of descriptive attributes
  N is number of examples in the training set

Datasets

- NWE - Mortality rate in North-West England at ward level
- SIGMEA (MS, MF) - Pollen dispersal rates of plants
- FOIXA - Pollen dispersal rates from neighboring GM fields
- FF - Forest fires in Montesinho Park, Portugal
- GASD - US county vote casts in 1980 presidential election



Spatial autocorrelation of the GASD dataset

Experimental Setup

- Each dataset is mapped into a data network *G=(V, E)*
  - node *u ϵ V* is associated with each observation (x_1, … ,x_n,y)
  - *u, v* are distinct nodes and there is an edge from *u* to *v* labelled with *w* in *E ((u, v, w) ϵ G)*, iff *v* is within a bandwidth *b*

- Bandwidth *b* (size of neighborhood in AC)
  - Percentage of the max distance between points in the data (1, 5, 10, 20) %
  - Automatically estimated bandwidth

- 10-fold cross validation

- Evaluation measures:
  - Mean Square Error (MSE)
  - Wilcoxon test

Results

- The effect of different definitions of distances between objects in the network:
  - distance over descriptive attributes (Desc.)
  - distance over spatial attributes (Spatial)
  - distance over descriptive & spatial attributes (Desc.+Spatial)
- Comparison over 12 combinations of: AC indices, combinations of variance reduction and AC indices and weighting functions
- The table shows the count of significantly better (++), better (+), worse (-) and significantly worse (--) Wilcoxon test results at 0.05

Dataset	Desc.+Spatial vs. Desc.				Desc.+Spatial vs. Spatial				Desc. vs. Spatial			
	++	+	−	−−	++	+	−	−−	++	+	−	−−
NWE	3	4	3	0	2	5	5	0	0	6	6	0
MS	0	0	11	0	0	5	7	0	0	10	2	0
MF	0	6	6	0	0	12	0	0	0	6	5	0
FOIXA	0	4	4	4	1	4	4	3	3	4	3	2
GASD	11	1	0	0	0	12	0	0	0	0	12	0
FF	0	0	0	0	0	0	0	0	0	0	0	0
Total.	14	15	24	4	3	38	16	3	3	26	28	2

- Best results using distance over descriptive & spatial attributes

Dataset	est b (%)	NCLUS CI						NCLUS P	
		α=0			α=0.5			α=0	α=0.5
		Mod.	Gauss.	Euc.	Mod.	Gauss.	Euc.		
NWE	7.67	0.0023	0.0023	0.0023	0.0024	**0.0022**	0.0024	0.0026	0.0024
MS	4.8	7.1220	6.1312	7.1220	6.8863	6.8863	6.8863	6.2860	7.1380
MF	9.14	2.4718	3.2346	2.4718	2.4718	2.4718	2.4718	2.4981	2.5133
FOIXA	64.62	1.0672	0.9220	1.0672	**0.7666**	0.8011	0.8011	0.9687	0.7313
GASD	2.5	0.1800	0.1808	0.1800	0.1770	0.1663	0.1780	0.1762	0.1734
FF	100	**47.224**	**47.224**	**47.224**	**47.224**	**47.224**	**47.224**	47.385	47.385

Dataset	NCLUS Global Moran						CLUS	CLUS*	ITL
	α=0			α=0.5					
	Mod.	Gauss.	Euc.	Mod.	Gauss.	Euc.			
NWE	0.0024	0.0024	0.0023	0.0023	0.0023	0.0024	0.0025	0.0025	0.0025
MS	7.1844	6.1311	7.3152	6.7851	6.9259	**5.0951**	5.9114	6.6845	5.8532
MF	2.4718	3.0922	2.4718	2.4718	2.4718	4.2877	**2.3532**	2.5390	2.4085
FOIXA	0.8231	0.8240	0.7751	0.8445	1.0201	0.7718	0.8920	0.8710	
GASD	0.1790	0.1688	0.1719	0.1695	0.1688	0.1688	0.1590	0.1590	**0.1316**
FF	**47.224**	**47.224**	**47.224**	**47.224**	**47.224**	**47.224**	47.950	47.998	64.731

Average MSE. For each dataset, the best results are in bold

NCLUS - The proposed algorithm

CLUS - The original algorithm for learning PCTs [Blockeel, 1998]

CLUS* - Modification of CLUS, considers the coordinates as targets, along with actual ones

ITL - Iterative Transductive Learning algorithm [Appice et.al., 2009]

Conclusions

- Network PCTs - extension of PSTs that explicitly considers autocorrelation
  - uses heuristic that is a weighted combination of variance reduction (predictive performance) & autocorrelation of response variable
  - when calculating the autocorrelation, considers:
    - known measures
    - different neighborhoods (b) sizes
    - different weighting schemes (degrees of smoothing)
  - automatically determine the appropriate bandwidth
- Empirical evaluation on real spatial data networks
- Performs better than
  - CLUS, CLUS* and ITL in most of the cases

Further Work

- Consider multi-objective problems
- Study different evaluation measures for multi-objective problems (autocorrelation on the combination of the target variables)
- Additional experiments/datasets
- Use network evaluation setup (e.g. network CV)
- Embed automatic determination of the relative weight (α) to balance variance reduction and autocorrelation

Q & A