technische universität dortmund

SFB 876 - Providing Information by Resource-Constrained Data Analysis

Mining Smartphone and Mobility Data Through Graphical Models

Katharina Morik, Artificial Intelligence,
TU Dortmund University, Germany

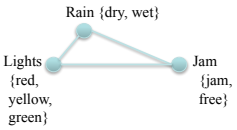
Overview

- Graphical Models: Markov Random Fields
- Spatio-Temporal Random Fields
  - App Usage Mining
  - Dublin Traffic Prediction
- Integer Markov Random Fields
- Smartphone – The human sensor

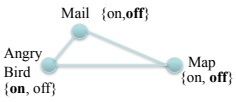
Analyzing Sensor Data

- Select a field of measurements.
- Establish nodes with possible states.
- Indicate dependencies of nodes by edges.
- Look for the joint probability mass function.

Streets in a city (area)



App usage of a person



Likelihood of joint realizations

- Graph G=(V,E)
- Multi-variate random variable **X** with finite discrete domain $\mathcal{X}$
- Mapping joint vertex assignment into vector space $\phi(\mathbf{x})$: $\mathcal{X} \longrightarrow \mathcal{R}^d$
- Parameter to be learned: $\boldsymbol{\theta}$ in $\mathcal{R}^d$

MRF :

$$p(x) = \frac{1}{Z(\theta)} \exp\left(\sum_i \theta_i \phi_i(x)\right) \quad \left|\frac{1}{a} = \exp(-\ln a)\right.$$

$$= \exp\left[\left(\sum_i \theta_i \phi_i(x)\right) - \ln Z(\theta)\right]$$

$$= \exp\left[\langle \theta, \phi(x) \rangle - A(\theta)\right]$$

Probabilistic Graphical Model -- Example

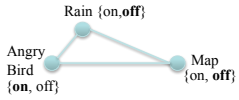
Observation
- $\mathbf{x}_1$:(on,off,off)
- $\phi(\mathbf{x}_1)$ = (1,0,0,1,0,1,1,0,0,0,1,0,0,0,0,0,1,0)
- d = 18

Learning
- Likelihood of joint realizations

$$p_\theta(x) = \exp\left(\langle \theta, \phi(x) \rangle\right) - A(x)$$

- Maximum a posteriori prediction of an unobserved node H, given observed nodes O

$$x_H^* = \operatorname{argmax}\ p_\theta\left(x_H \middle| x_O\right)$$



Graph G=(V,E)

$\phi(\mathbf{x})$: (/* Nodes*/
1. **on**, /*dom(angry)*/
2. off,
3. on, /*dom(rain)*/
4. **off**,
5. map, /*dom(map)*/
6. **off**,

/*Edges*/
1. **on, off**, /* edge angry-rain*/
2. on,on,
3. off,off,
4. off, on,
5. **on,off**, /*edge angry-map*/
6. on, on,
7. off, off,
8. off,on,
9. on, off, /*edge rain-map*/
10. on, on,
11. **off, off,**
12. off,on)

)

Parameter Estimation

- Maximum Likelihood Estimation (MLE)
- Average loglikelihood
  - empirical expectation of the value $\phi(x)$
- Optimization: partial derivative of l ($\boldsymbol{\theta}$|D); gradient descent minimizing –l ($\boldsymbol{\theta}$|D)
- Delivers optimal $\boldsymbol{\theta}$.

$$L(\theta|D) = \prod_{x \in D} p(x)$$

$$l(\theta|D) = \frac{1}{|D|} \sum_{x \in D} \log p(x)$$

$$= \frac{1}{|D|} \sum_{x \in D} \left[\langle \theta, \phi(x) \rangle - A(\theta)\right]$$

$$= \left\langle \theta, \frac{1}{|D|} \sum_{x \in D} \phi(x) \right\rangle - A(\theta)$$

$$= \left\langle \theta, E\left(\phi(x)|D\right) \right\rangle - A(\theta)$$

Loopy Belief Propagation

- What is the expectation of $\boldsymbol{\phi}(x)$?
- All nodes need to fit together!
- For all pairs of linked nodes u,v send messages $m_{vu}(x_u)$

$$m_{vu}(x_u) = \sum_{x_v \in X} \varphi_{v,u}(x_v, x_u)\varphi(x_v) \prod_{w \in neighbor_v} m_{wv}(x_v)$$

$$M_v(x) := \prod_{u \in neighbor_v} m_{uv}(x)$$

$$p_v(x_v) = \frac{\varphi_v(x_v) M_v(x_v)}{\sum_{x \in X} \varphi_v(x_v) M_v(x)}$$

Probabilistic Graphical Model -- Example

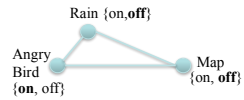
Observation
- $\mathbf{x}_1$: (on,off,off)
- $\boldsymbol{\phi}(\mathbf{x}_1)$ = (1,0,0,1,0,1,1,0,0,1,0,0,0,0,1)
- d = 18

Learning
- Likelihood of joint realizations

$$p_\theta(x) = \exp(\langle \theta, \phi(x) \rangle) - A(x)$$

- Maximum a posteriori prediction of an unobserved node H, given observed nodes O

$$x_H^* = \operatorname{argmax} p_\theta(x_H | x_O)$$



Graph G=(V,E)

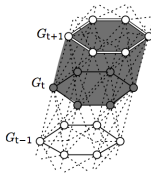
$\boldsymbol{\phi}(\mathbf{x})$: (/* Nodes*/
1. **on**, /*dom(angry)*/
2. off,
3. on, /*dom(rain)*/
4. **off**,
5. on, /*dom(map)*/
6. **off**,

/*Edges*/
1. **on, off**, /* edge angry-rain*/
2. on,on,
3. off,off,
4. off,on,
5. **on,off**, /*edge angry-map*/
6. on, on,
7. off, off,
8. off,on,
9. on, off, /*edge rain-map*/
10. on, on,
11. **off, off**
12. off,on)

)

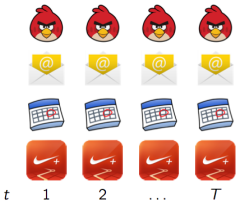
Spatio-temporal Random Fields

- The spatio-temporal graph is trained to predict each node's maximum a posteriori probability with the marginal probabilities.
- Generative model predicting all nodes.
  Maximum a posteriori prediction of an unobserved node H, given observed nodes O.
- The learned model answers diverse questions – you only need to choose H.



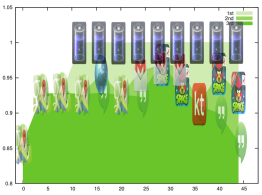
Spatio-temporal Models for App Usage Mining

- Spatial graph with Apps as nodes and edges based on covariance matrix.
- Windows of 30 minutes arranged for a day.
- Likelihood of an app at a particular time of day.
- Data from 8 users, public: http://sfb876.tu-dortmund.de/auto?self=$e675o3goow



STRF modeling usage of apps on Android phones

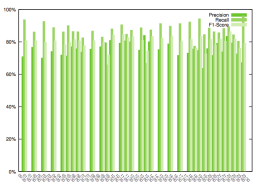
- Probability of an app being switched on
- Ranking according probability
- Show the first three ranks with their probability.

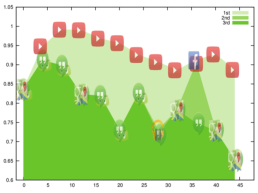


User G keeps battery on, plays Angry Bird, and looks up the map and communicates using hangout.

The Google addict

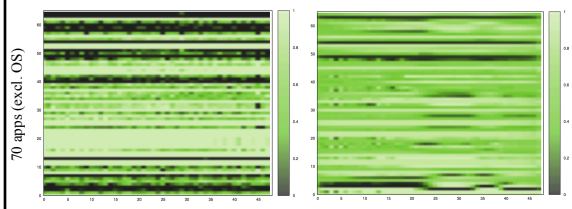
YouTube, hangout, music, maps





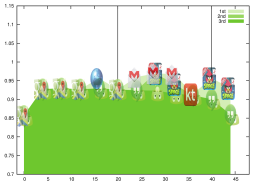
Facebook in the evening

Data record and STRF model of user G



Warn the user based on the prediction

- Given the likelihood of apps, the energy consumption of these apps,
- Calculate when the battery will be empty.
- Here:
  battery was full, when the picture starts,
  already at lunch time, we predict the battery being empty at dinner time,
  if not loaded.



Flood: City and Country

- The city of Dublin suffers from flood due to rainfall and tide. Wanted:
  - Optimal traffic regulations based on traffic predictions.
- Big data analytics for real-time survey of the situation.
- European project INSIGHT
- European project VaVel
  - Both coordinated by Dimitrios Gunopoulos

Smart trip modeling for Dublin

- Open Street Map → graph topology
- Open Trip Planner: user query (v,w), route planning based on traffic costs.
- Traffic costs learned:
  - Spatio-temporal random field based on sensor data stream;
  - Gaussian process estimates values for non-sensor locations.
- Framework for real-time processing of data streams, XML configuration of data flow, connecting data, traffic model and planner.

Constructing the spatio-temporal graph of Dublin

OpenStreetMap streets segmented according to junctions.
966 sensors transmit traffic flow every 6 minutes (www.dublinked.ie).
Aggregate sensor readings for 30 minutes, aggregate sensor nodes by 7NN.
Traffic flow discretized into 6 intervals of density.
48 time layers for each day (48* 30=1440 minutes make a day).
Training for every weekday.
Predicting density of each node and edge – interpreted as costs.

Using STRF for smart trip modeling -- Evaluation

- Confusion matrix of predicting the number of vehicles (6 intervals) for all sensors and all half hours following 1 pm on Fridays, tested on March 1.,8.,15.,22.,29.
- given the traffic at 1 p.m. (bold is true).

Predicted → / True ↓	**0**	**1-5**	**6-20**	**21-30**	**31-60**	**61-**	**Prec**
0	**840**	32	10	6	3	0	**0.943**
1-5	2	**632**	498	3	0	1	0.556
6-20	91	156	**12169**	2006	83	25	**0.838**
21-30	32	0	1223	**5637**	717	14	0.739
31-60	43	0	60	893	**1945**	29	0.655
61-	0	0	16	3	12	**35**	0.530
Recall	**0.833**	0.771	**0.871**	0.659	0.705	0.34	

Smart traffic for smart cities

- Spatio-temporal modeling is natural for traffic prediction.
- Several questions can be answered using the same learned model.
- The answers come along with their probabilities. This might be helpful for decision makers.
- Integration of Spatio-temporal random fields into the Open Trip Planner and Gaussian information completion resulted in an excellent navigation system.

Piatkowski, Lee, Morik (2013) Spatio-temporal random fields: compressible representation and distributed estimation, Machine Learning Journal 93:1, 115 – 140.
Liebig, Piatkowski, Bockermann, Morik (2014) Predictive Trip Planning – Smart Routing in Smart Cities, Mining Urban Data Workshop at 17th Intern. Conf. on Extending Database Technology.

Graphical models **on** resource-restricted processors

- Spatio-Temporal Random Fields on the phone. but
- Power consumption is restricted!
- "The most obvious technique to conserve power is to reduce the number of cycles it takes to complete a workload."
  (Intel 64, IA-32 architectures optimization reference manual, guidelines for extending battery life)
- Restrict the parameter space of the Markov Random Field

$$\theta \in \{0,1,\ldots,K\} \subset \mathbb{N}$$

	Sandy Bridge		**ARM 11**	
	Real	Int	Real	Int
+	3	**1**	8	**1**
*	5	**3**	8	4-5
/	14	13-15	19	-
Bit shift	-	3	-	2

Clock cycles for arithmetics on different processors: Real vs. integer.

Parameter space transformation

- Graph model tree-structured
- Transform the parameter space:
  $\eta_i(\theta) = \theta_i \ln 2$

MRF :

$$p(\vec{x}) = \frac{1}{Z(\theta)} \exp\left(\sum_i \theta_i \phi_i(\vec{x})\right)$$

$$= \exp\left[\langle \theta, \phi(\vec{x}) \rangle - A(\theta)\right]$$

IntegerMRF :

$$p(\vec{x}) = \exp\left[\langle \eta(\theta), \phi(\vec{x}) \rangle\right]$$

$$= 2^{\left[\langle \theta, \phi(x) \rangle - A(\eta(\theta))\right]}$$

$$= \frac{2^{\langle \theta, \phi(x) \rangle}}{\sum_{y \in \mathcal{X}} 2^{\langle \theta, \phi(y) \rangle}}$$

Integer belief propagation

- Simply replacing the exp(.) by $2^{(\cdot)}$ is not sufficient
  - Overflows are normally avoided by normalization.
  - Normalization is impossible in integer division.
- Magnitude of messages corresponds to probability
  - Use the length of each message
  - Bit-length is similar to log

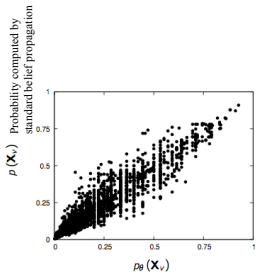
$$m_{vu}(y) = \sum_{x \in \mathcal{X}_v} \exp\left(\theta_{vu=xy} + \theta_{v=x}\right) \prod_{w \in N_v - \{u\}} m_{wu}(x)$$

$$\tilde{m}_{vu}(y) = \sum_{x \in \mathcal{X}_v} 2^{\left(\theta_{vu=xy} + \theta_{v=x}\right)} \prod_{w \in N_v - \{u\}} \tilde{m}_{wu}(x)$$

$$\beta_{vu}(y) = \max_{x \in \mathcal{X}_v} \theta_{vu=xy} + \theta_{v=x} + \sum_{w \in N_v - \{u\}} \beta_{wu}(x)$$

Discretized probability space

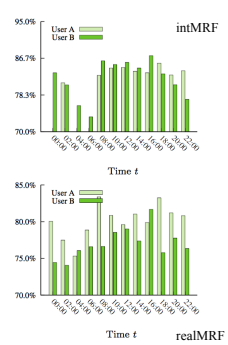
- Belief propagation is now bit-length propagation, i.e. the MAP and marginals are computed using the bit-length.
- The approximation error depends on the number of neighboring nodes and the space of states.
- Some true probabilities (y axis) cannot be expressed by the integer approximation (x axis).



Probability computed by bit-length approximation

Experiments

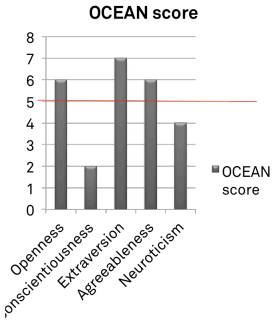
- User A
  - 2064 vertices
    (app at time = {on, off})
  - 23368 spatio-temporal edges
  - 38 training instances,
    32 test instances
- User B
  - 2064 vertices
  - 14998 spatio-temporal edges
  - 125 training instances
    119 test instances
- Comparing integer MRF and real MRF
  - Average of 50 trainings
  - Runtime: intMRF 156 (136) s
    realMRF 466 (419) s
  - Predicting apps at t+1, given apps at t



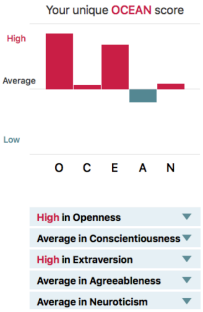
Smartphones – the Human Sensor

- Rayid Ghani: chief scientist at the 2012 Obama campaign: fundraising, voter mobilization, volunteer (invited talk ECML PKDD 2013 Prague)
- Personality testing on smart phones: B. Rammstedt and O. P. John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in English and German. Journal of Research in Personality, 2007
- MyPersonality (started 2008 at Cambridge, UK, by Michael Kosinski)
  publicly available data for unregistered users:
  Facebook status upates of 250 users + personality + Facebook properties; 3120 hand-labeled status updates
  for registered users: 3,100 000 Big 5 scores; Facebook friendship graphs, activity records, topics, dictionaries, demographics
  https://applymagicsauce.com
- Cambridge Analytica (Alexander J. A. Nix) claims to have won the election for Trump using 5000 data points on 220 000 000 Americans, predictive models with 100 variables
  https://ocean.cambridgeanalytica.org

Big Five (OCEAN) from Questionnaire



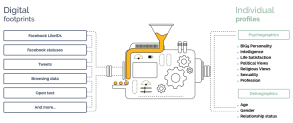
Psychomeda L.Satow (German)

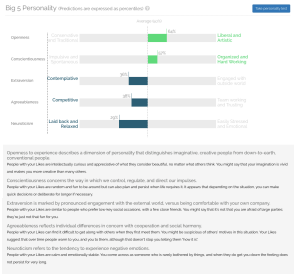


Cambridge Analytica online test

Big 5 (OCEAN) from Facebook Likes

- University of Cambridge, Psychometrics Centre Method: ApplyMagicSauce
- Michael Kosinski (now Stanford)





Using 12 Fb LIKES

App Usage Mining

- Smartphones produce big data.
- Each user generates about 60 gigabytes of data per year. This is big data!
- Data curation is time-consuming.
- Basic Data representation:
  - Did $user_i$ start app_j ?

Binary	app_1	...	app_m
usr_1	1		0
usr_n	0		1

App Usage Mining

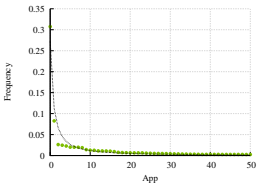
- Smartphones produce big data.
- Each user generates about 60 gigabytes of data per year. This is big data!
- Data curation is time-consuming.
- Basic Data representation:
  - Did $user_i$ start app_j ?
  - How often did $user_i$ start app_j ?
- Probability distributions deliver an overview of the data.

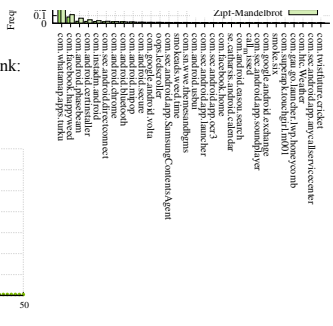
Frequent	app_1	...	app_m
usr_1	#starts		
usr_n			#starts(n,m)

Probability Distributions of Apps

App usage follows the Zipf or Mandelbrot distribution:

- The probability of an app is inverse proportional to its rank: $p(n) = 1/n$





App Usage Mining

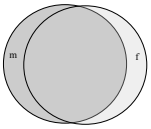
- Smartphones produce big data.
- Each user generates about 60 gigabytes of data per year. This is big data!
- Data curation is time-consuming.
- Basic Data representation:
  - Did $user_i$ start app_j ?
  - How often did $user_i$ start app_j ?
- Probability distributions deliver an overview of the data.
- Additional attributes of users can be predicted.

Binary	app_1	...	app_m	target
usr_1	0		1	
usr_n	1		0	

Predicting Attributes -- Gender

- Menthal data from Alexander Markowetzhttps://menthal.org
- 42,482 men and 31,700 women were logged.
- Top 99 apps of women and top 99 apps of men overlap by 82 apps.
- We can predict the gender by app usage data using logistic regression and the L1 norm: accuracy 91.2%.

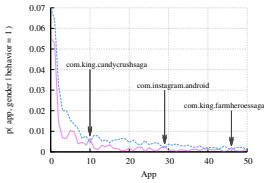
Binary	app_1	...	app_m	gender
usr_1	0		1	f
usr_n	1		0	m



Inspecting the Results -- Gender

- Menthal data from Alexander Markowetz https://menthal.org
- 42,482 men and 31,700 women were logged.
- Ranking apps according to their frequency, compare rankings of men and women.
- Largest differences:
  - ClashOfClans men 80, women 219
  - FarmHeroes men 249, women 78
  - GoogleDocs men 52, women 92

Frequent	app_1	...	app_m
usr_1	#starts		
usr_n			#starts(n,m)



App Usage Mining

- Smartphones produce big data.
- Each user generates about 60 gigabytes of data per year. This is big data!
- Data curation is time-consuming.
- Individual data:
  - Where did $user_i$ use app_j ? Include only places that occur more than t times.

Usr_n	app_1	...	app_m

Usr_i	app_1	...	app_m
$place_1$	#starts		

Usr_1	app_1	...	app_m
$place_1$	#starts		
$place_n$			#starts(n, m)

App Usage Mining

- Smartphones produce big data.
- Each user generates about 60 gigabytes of data per year. This is big data!
- Data curation is time-consuming.
- Individual data:
  - Where did $user_i$ use app_j ? Include only places that occur more than t times.
  - When did $user_i$ use app_j ? Discretized time of day: 30 minutes

Usr_n	app_1	...	app_m

Usr_i	app_1	...	app_m
0	#starts		

Usr_1	app_1	...	app_m
0	#starts		
1			
...			
48			#starts(n, m)

References

- Nico Piatkowski, Sangkyun Lee, Katharina Morik (2016) Integer Undirected Graphical Models for Resource-Constrained Systems. In: Neurocomputing, 173:1, 9 – 23
- Nico Piatkowski, Katharina Morik (2016) Stochastic Discrete Clenshaw-Curtis Quadrature. In: 33rd Int. Conference on Machine Learning, Proceedings JMLR
- Marco Stolpe, Thoms Liebig, Katharina Morik (2015): Communicaton-efficient learning of traffic flow in a network of wireless presence sensors. In: Workshop Parallel and Distributed Computing for KDD, CEUR-WS
- Thomas Liebig, Nico Piatkowsi, Christian Bockermann, Katharina Morik (2014): Predictive Trip Planning – Smart Routing in Smart Cities. In: EDBT/ICDT Workshops.CEUR-WS
- Francois Schnitzler, Thomas Liebig, Shie Mannor, Katharina Morik (2014): Combining a Gaussian Markov model and Gaussian process for traffic prediction in Dublin city center. In: EDBT/ICDT, CEUR-WS
- Jochen Streicher, Nico Piatkowski, Katharina Morik, Olaf Spinczyk (2013) Open Smartphone Data for Mobility and Utilization Analysis in Ubiquitous Environments. In: Atzmüller, Scholz (eds) 4th Int. Workshop on Mining Ubiquitous and Social Environments (MUSE)
- Nico Piatkowski, Sangkyun Lee, Katharina Morik (2013) Spatio-temporal Random Fields: Compressible Representation and Distributed Estimation. In: Machine Learning Journal, 93:1, 115 – 140.
- Thomas Liebig, Nico Piatkowski, Christian Bockermann, Katharina Morik (2014) Predictive Trip Planning – Smart Routing in Smart Cities. In: Mining Urban Data Workshop at 17th Intern. Conf. on Extending Database Technology

Code and local data available at
http://sfb876.tu-dortmund.de/index.htm