Topic IV.2: Tensor Applications

Discrete Topics in Data Mining
Universität des Saarlandes, Saarbrücken
Winter Semester 2012/13

Topic IV.2: Tensor Applications

1. Tucker2 Decompositions and RESCAL
   1.1. Tucker2 and equivalent factors
   1.2. The RESCAL algorithm
2. Recap of the course
3. Feedback from Essay III

Tucker2 Decompositions and RESCAL

- Recall: Tucker3 decomposition decomposes a 3-way tensor into smaller core tensor and three factor matrices
- Tucker2 decomposition decomposes a 3-way tensor into a core tensor and **two** factor matrices
  - If the original tensor was of size N-by-M-by-K, the core is of size I-by-J-by-K (or M-by-I-by-J or I-by-M-by-J)
  - Equivalently, Tucker2 is Tucker3 with one factor matrix replaced with an identity matrix

More on Tucker2

- Tucker2 can be presented slice-wise:

  $\mathbf{X}_k = \mathbf{A}\mathbf{G}_k\mathbf{B}^T$ for each k

  - $\mathbf{X}_k$ is the kth (frontal) slice of $\mathcal{X}$
  - $\mathbf{G}_k$ is the kth (frontal) slice of the core tensor $\mathcal{G}$
  - $\mathbf{A}$ and $\mathbf{B}$ are the factor matrices
- In matricized form

  $\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}(\mathbf{I}_K \otimes \mathbf{B})^T$

  - $\mathcal{X}$ is N-by-M-by-K

What if B = A?

- Assume our tensor's two modes represent same entities
  - E.g. tensor is subject–relation–object, with subjects and objects from the same set of entities (e.g. humans)
  - Sender–topic–receiver with senders and receivers in the same set of people
- We can model this by restricting the two factor matrices to be the same
  - "Flow of information"
  - If we assign a dimension into a factor in one mode, that assignment holds also in the other mode

The RESCAL Problem

- Given an N-by-N-by-M tensor $\mathcal{X}$ and rank R, find an N-by-R factor matrix $\mathbf{A}$ and R-by-R-by-M core tensor $\mathcal{G}$ such that they minimize

$$\underbrace{\frac{1}{2}}_{\text{Notational convenience}}\underbrace{\left(\sum_{m=1}^{M}\left\|\mathbf{X}_m - \mathbf{A}\mathbf{G}_m\mathbf{A}^T\right\|_F^2\right)}_{\text{Squared error}} + \underbrace{\frac{1}{2}}_{\text{Notational convenience}}\underbrace{\lambda}_{\text{Regularization coefficient}}\underbrace{\left(\|\mathbf{A}\|_F^2 + \sum_{m=1}^{M}\|\mathbf{G}_m\|_F^2\right)}_{\text{Regularizer}}$$

Nickel, Tresp & Kriegel 2011, 2012

The RESCAL Algorithm

- Iterative updates
  - In updating $\mathbf{A}$ for $\mathbf{AG}_m\mathbf{A}^T$, we temporarily consider $\mathbf{A}$ and $\mathbf{A}^T$ different matrices, and only update $\mathbf{A}$
- **Updating A**: We stack the frontal slices of the data side-by-side and solve the resulting matrix problem
  - $\mathbf{Y} \approx \mathbf{AH}(\mathbf{I}_{2M} \otimes \mathbf{A}^T)$
    - $\mathbf{Y} = (\mathbf{X}_1, \mathbf{X}_1^T, \mathbf{X}_2, \mathbf{X}_2^T, \ldots, \mathbf{X}_M, \mathbf{X}_M^T)$
    - $\mathbf{H} = (\mathbf{G}_1, \mathbf{G}_1^T, \mathbf{G}_2, \mathbf{G}_2^T, \ldots, \mathbf{G}_M, \mathbf{G}_M^T)$
  - The gradient of this is

$$\mathbf{H}\Big((\mathbf{I} \otimes \mathbf{A}'^T\mathbf{A}')\mathbf{H}^T\mathbf{A}^T - (\mathbf{I} \otimes \mathbf{A}'^T)\mathbf{Y}^T\Big) + \lambda\mathbf{A}^T$$

    - Here **A**' is the version of A kept constant

Update Rules Continued

- Setting the gradient to zero, we get update rule

$$\mathbf{A} \leftarrow \left(\sum_{m=1}^{M} \mathbf{X}_m \mathbf{A} \mathbf{G}_m^T + \mathbf{X}_m^T \mathbf{A} \mathbf{G}_m \right) \left(\sum_{m=1}^{M} \mathbf{B}_m + \mathbf{C}_m + \lambda \mathbf{I} \right)^{-1}$$

  – Here $\mathbf{B}_m = \mathbf{G}_m \mathbf{A}^T \mathbf{A} \mathbf{G}_m{}^T$ and $\mathbf{C}_m = \mathbf{G}_m{}^T \mathbf{A}^T \mathbf{A} \mathbf{G}_m$

- **Updating G**: Writing $\mathbf{X}_m$ and $\mathbf{G}_m$ as vectors, we get optimization task

$$\|\mathrm{vec}(\mathbf{X}_m) - (\mathbf{A} \otimes \mathbf{A})\mathrm{vec}(\mathbf{G}_m)\| + \lambda \|\mathrm{vec}(\mathbf{G}_m)\|$$

  – Regularized linear regression

$$\mathbf{G}_m \leftarrow (\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{I})^{-1} \mathbf{Z} \mathtt{vec}(\mathbf{X}_m)$$

    - $\mathbf{Z} = \mathbf{A} \otimes \mathbf{A}$

A Bit on Complexity

- $\mathbf{Z} = \mathbf{A} \otimes \mathbf{A}$ can be huge
  - The most expensive computation is $(\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I}) - 1$
  - The same computation works for every frontal slice of $\mathcal{X}$
  - If there's no regularization at $\mathcal{G}$, then this becomes $(\mathbf{Z}^T\mathbf{Z})^{-1} = ((\mathbf{A} \otimes \mathbf{A})^T(\mathbf{A} \otimes \mathbf{A}))^{-1} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A} \otimes (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}$
    - Only needs the inverse of R-by-R matrix $\mathbf{A}^T\mathbf{A}$
- We can use the QR matrix decomposition
  - $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular
  - We get $\mathbf{X}_m - \mathbf{A}\mathbf{G}_m\mathbf{A}^T = \mathbf{X}_m - \mathbf{QR}\mathbf{G}_m\mathbf{R}^T\mathbf{Q}^T$ $= \mathbf{Q}^T\mathbf{X}_m\mathbf{Q} - \mathbf{R}\mathbf{G}_m\mathbf{R}^T$
    - Now $\mathbf{R}$ is only R-by-R

More on Computational Complexity

$$\mathbf{A} \leftarrow \left(\sum_{m=1}^{M} \mathbf{X}_m \mathbf{A} \mathbf{G}_m^T + \mathbf{X}_m^T \mathbf{A} \mathbf{G}_m \right) \left(\sum_{m=1}^{M} \mathbf{B}_m + \mathbf{C}_m + \lambda \mathbf{I} \right)^{-1}$$

O(R^3)

p = number of non-zeros in $\mathcal{X}$

O(pNR)

O(NR^2)

$\lambda = 0$: O(p^3)
$\lambda \neq 0$: O(p^5)

$$\mathbf{G}_m \leftarrow (\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{I})^{-1} \mathbf{Z} \,\mathrm{vec}(\mathbf{X}_m)$$

O(pR^3)

QR of **A**: O(NR^2)
Projection $\mathbf{Q}^T \mathbf{X}_m \mathbf{Q}$: O($pNR^2$)

Application of RESCAL

- Tensor factorizations like RESCAL can be used for link prediction
  - Non-zero elements mean observed links
  - Zero elements mean unobserved
- The factorization will give us a representation of the original tensor where some of the zero elements will be represented with values above some threshold t
  - These elements are predicted as missing links
  - This can be evaluated using training data
- Problem: Multiplying the factors back is very expensive operation

Recap of the Course

- Discrete topics in data mining
  - A.k.a. “What Pauli likes to talk about in DM”
  - The modules of the course are not strongly connected
    - But some connections exist…
- Aim: high-level view of the ideas
  - Not too much details (too little details?)
- Few selected papers on each topic
  - Not necessarily the “best” papers
  - Very subjective selections process
- Essays instead of home works
  - Good (?) training for reading and writing

Intro

- Data mining, in a broad sense, is the set of techniques for analyzing and understanding data. (Zaki & Meira)
  - Is data mining voodoo science?
- Data mining is also a methodological science
  - The development of the tools to do data mining
  - C.f. statistics

Topic I: Pattern Set Mining

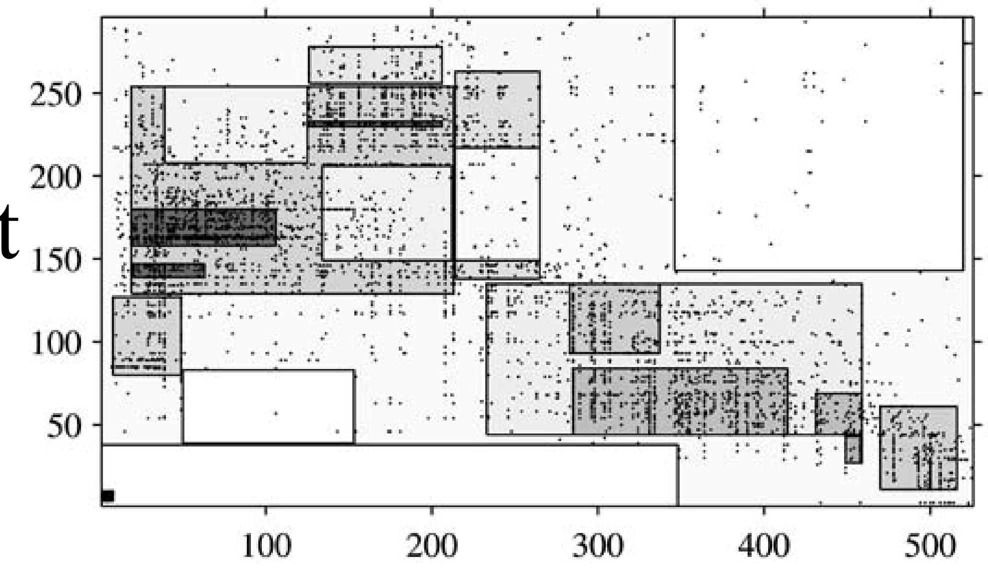
- What are patterns?
  - Frequent itemsets? Others?
- The flood of itemsets
  - Closed itemsets
    - No item can be added without changing the support
  - Maximal itemsets
    - No item can be added without becoming infrequent
  - Non-derivable itemsets
    - The support can’t be computed from subsets support

Tiling problems

- **Minimum tiling.** Given $\boldsymbol{X}$, find the least number of tiles $(\boldsymbol{r},\boldsymbol{c})$ such that
  - For all (i,j) s.t. $x_{ij} = 1$, there exists at least one pair $(\boldsymbol{r},\boldsymbol{c})$ such that $i \in \boldsymbol{r}$ and $j \in \boldsymbol{c}$ (i.e. $x_{ij} \in \boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$)
    - $i \in \boldsymbol{r}$ if exists j s.t. $r_j = i$
- **Maximum *k*-tiling.** Given $\boldsymbol{X}$ and integer k, find k tiles $(\boldsymbol{r}, \boldsymbol{c})$ such that
  - The number of elements $x_{ij} = 1$ that do belong in at least one $\boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$ is maximized

Geometric Tiles

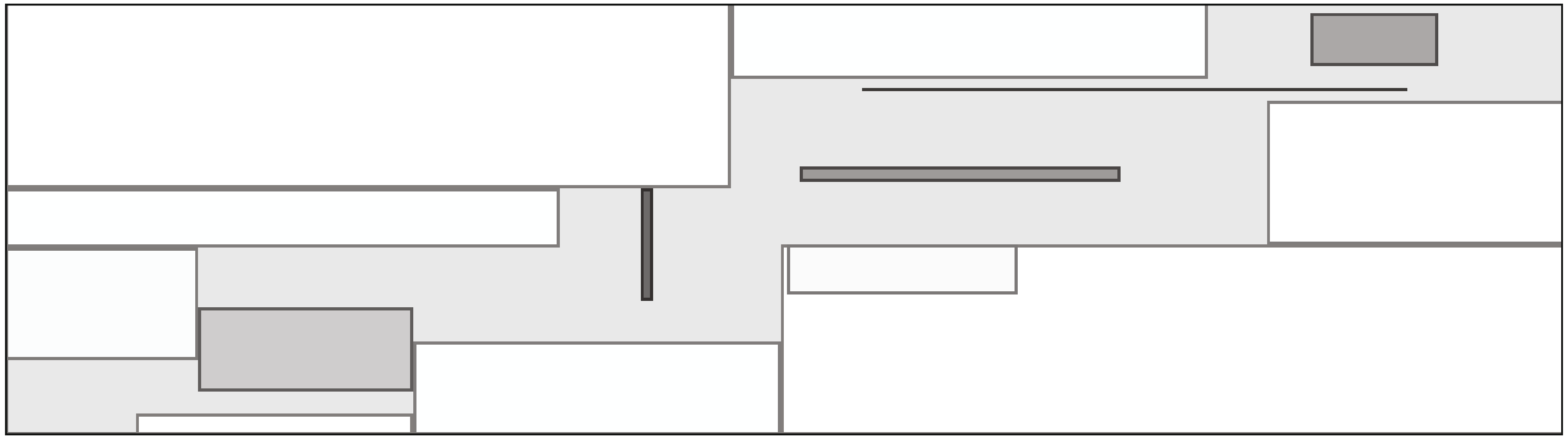
- There are $2^n 2^m$ possible combinatorial submatrices in an n-by-m matrix
  - If we look for density, we cannot look just monochromatic areas
- A **geometric (density) tile** is a tile with continuous row and column indices Teksti
  - It can be described given two corners
    - Or specific corner plus width and height
  - Only $n^2 m^2$ possible



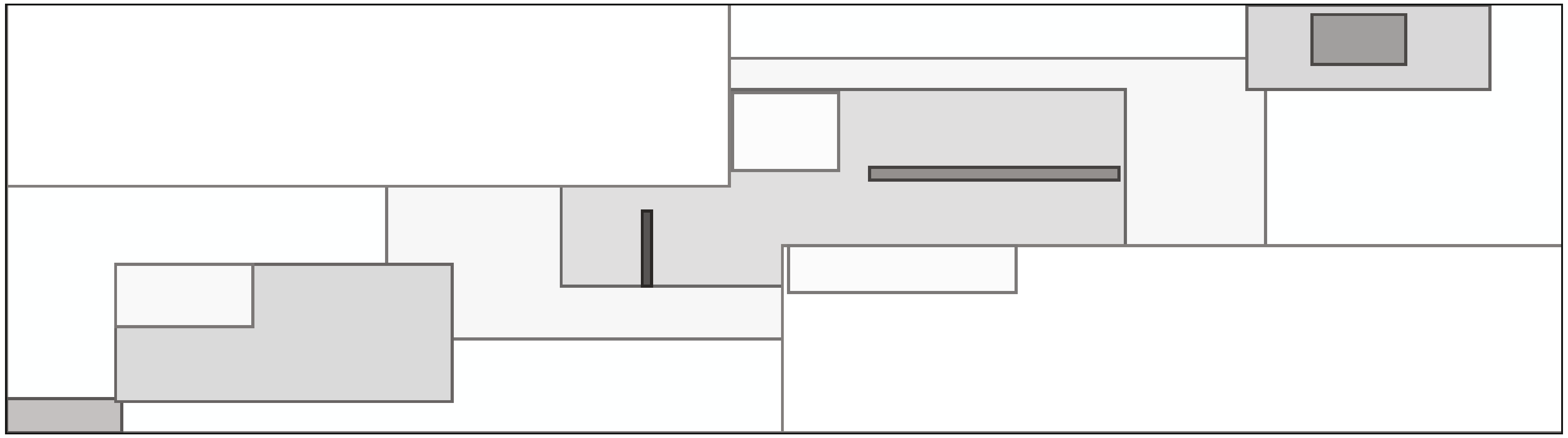
- We also allow a hierarchy of tiles
  - A sub-tile must be completely within its parent

Tiles That Overlap Within Parents

No overlap



Overlap within parent



Tatti & Vreeken 2012

The MDL Principle and Data Mining

- The MDL principle can be used to combat *overfitting*
  - Overfitting: model explains the training data too well and doesn't generalize to unseen data
  - MDL presents a natural penalty to too complex models
- The MDL principle can be used to *select* the output
  - Among many possible sets of results (models), select the one that compresses the data best
  - Note: we must explain the *whole* data
    - E.g. MDL does not allow lossy compression
    - But we can circumvent this by having a lossy model and a correction term (error)

Example of a Final Code Table

Code Table

Itemset	*Code*	*Usage*
A C		3
B D		3
C E		2
A		1
B		2
C	–	0
D		1
E		1

Topic II: Graph Mining

- Graphs are everywhere
  - Analysing them is important
- Measures of centrality
  - Degree centrality
  - Eccentricity centrality
  - Closeness centrality
  - Betweenness centrality
  - Prestige
  - PageRank
- Random graph models
  - Erdős–Renyi
  - Watts–Strogats
  - Barabási–Albert

Frequent Subgraph Mining

- Given a set D of n graphs and a minimum support parameter *minsup*, find all connected graphs that are subgraph isomorphic to at least *minsup* graphs in D
  - Enormously complex problem
  - For graphs that have m vertices there are
    - $2^{O(m^2)}$ subgraphs (not all are connected)
  - If we have s labels for vertices and edges we have
    - $O\left((2s)^{O(m^2)}\right)$ labelings of the different graphs
  - Counting the support means solving multiple NP-hard problems

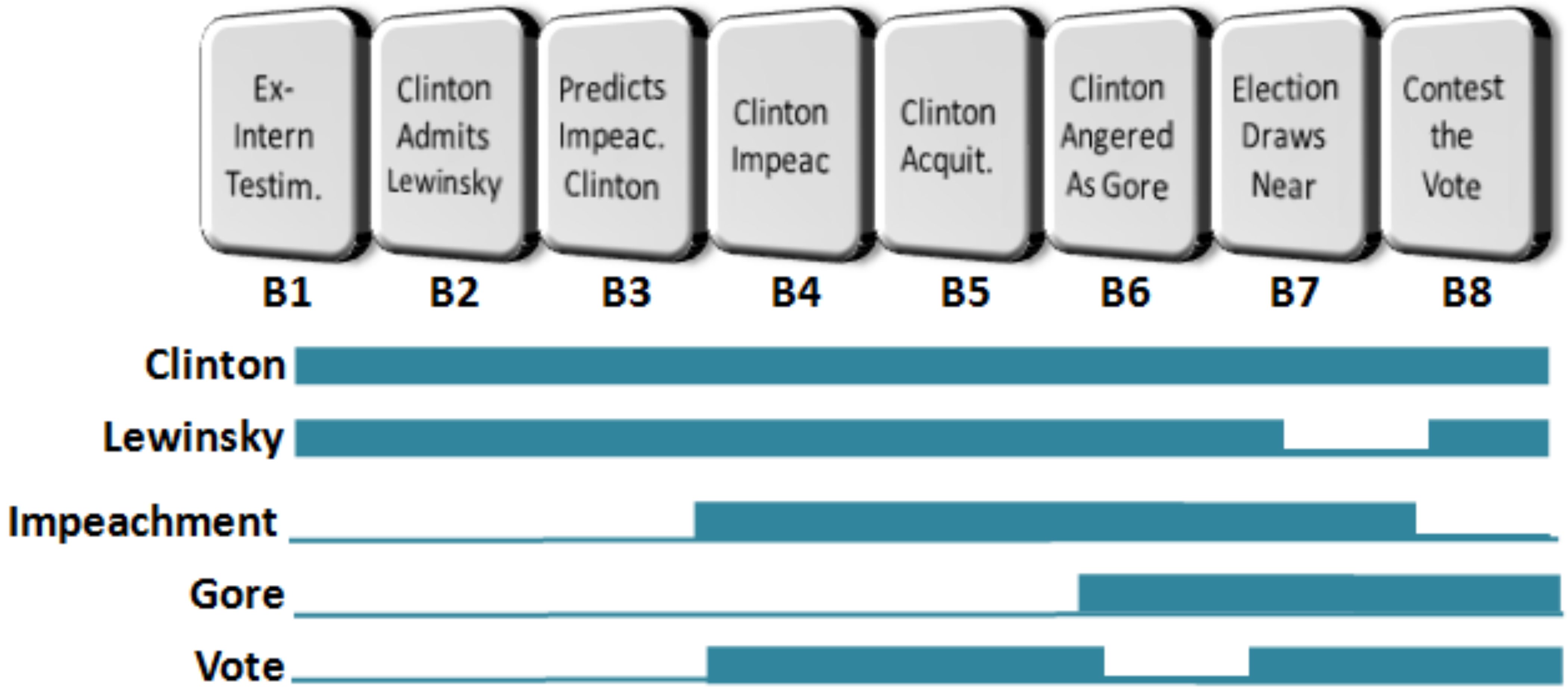
The AGM Algorithm

- Start with frequent graphs of 1 vertex
- **while** there are frequent graphs left
  - Join two frequent (k–1)-vertex graphs
  - Check the resulting graphs subgraphs are frequent
    - If not, **continue**
  - Compute the canonical form of the graph
    - If this canonical form has already been studied, **continue**
  - Compare the canonical form with the canonical forms of the k-vertex subgraphs of the graphs in D
    - If the graph is frequent, keep, otherwise discard
- **return** all frequent subgraphs

The gSpan Algorithm

- `gSpan`:
  - **for each** frequent 1-edge graphs
    - **call** `subgrm` to grow all nodes in the code tree rooted in this 1-edge graph
    - remove this edge from the graph
- `subgrm`
  - **if** the code is not canonical, return
  - Add this graph to the set of frequent graphs
  - Create each super-graph with one more edge and compute its frequency
  - **call** `subgrm` with each frequent super-graph

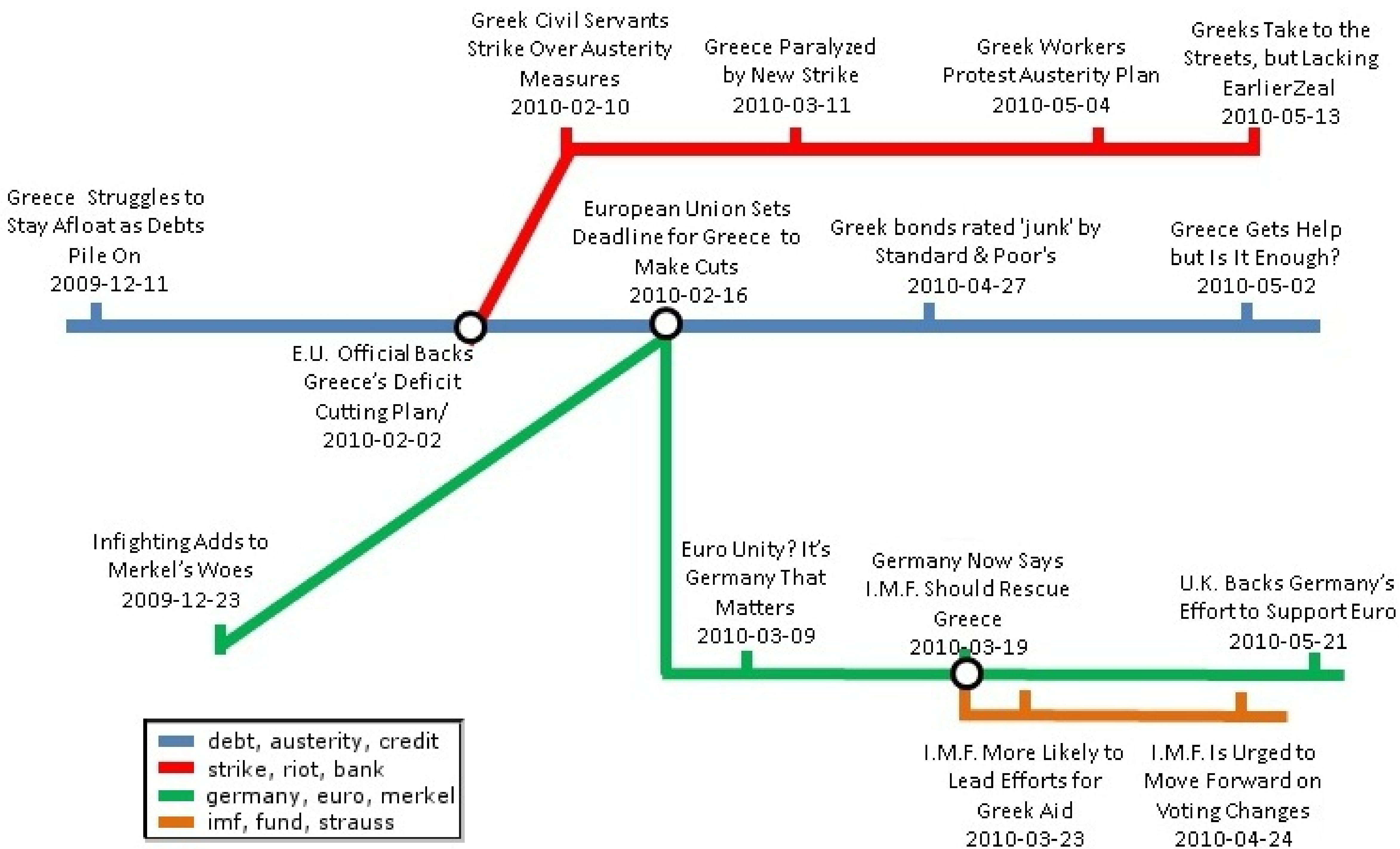
More Coherent Story



Topic consistent over transitions

Shahaf & Guestrin 2010

More Detailed Example



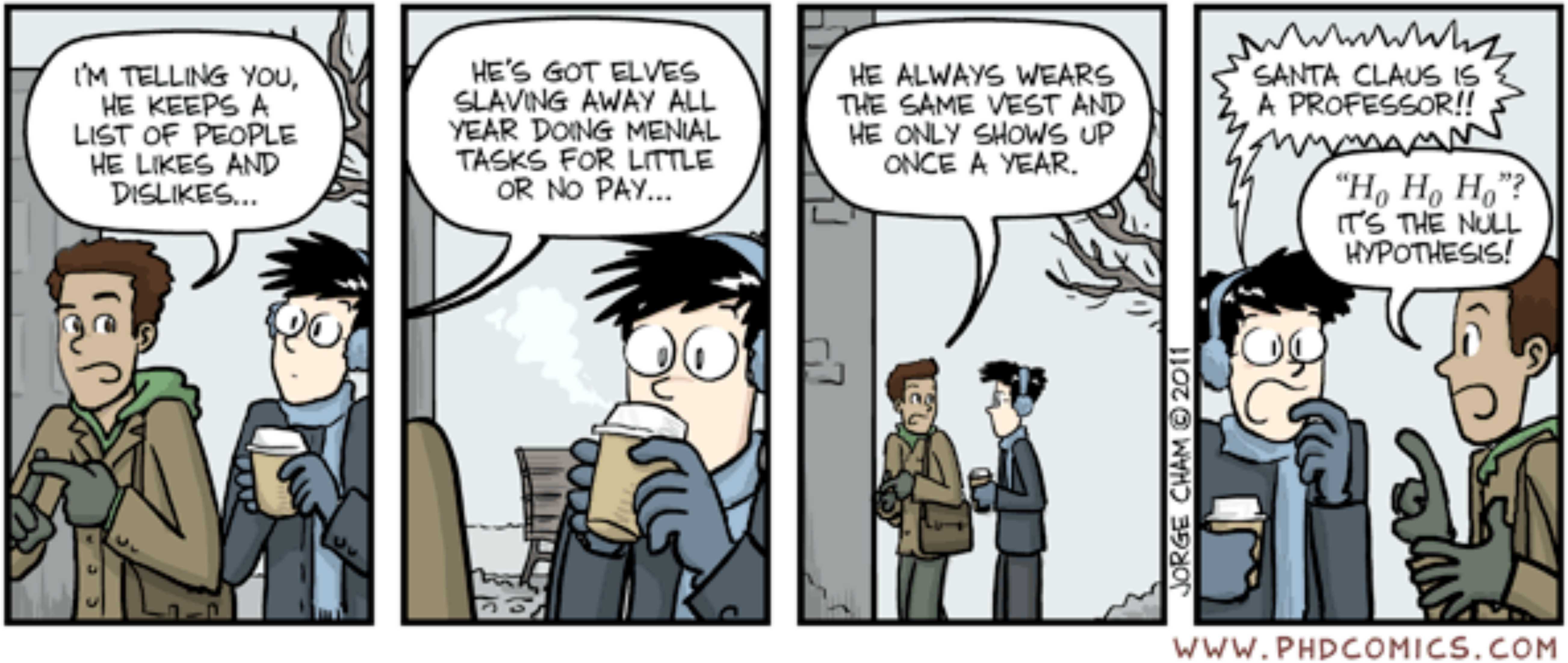
Shahaf, Guestrin & Horvitz 2012a

Topic III: Significance Testing

- The bread-and-butter of statistics
- Are my finding significant?
  - How to test this in data mining?

The Main Idea

- Let $O_{k,s}$ be the number of observed k-itemsets of support at least s
  - Let $\hat{O}_{k,s}$ be the random variable corresponding to that in a random dataset
- **Theorem.** There exists a level $s_{\min}$ such that if $s \geq s_{\min}$, $\hat{O}_{k,s}$ is approximated well by Poisson distribution
  - With this, we can compute the p-values easily
    - No need for data samples (almost…)
  - Only works with large-enough support levels
    - Rare events

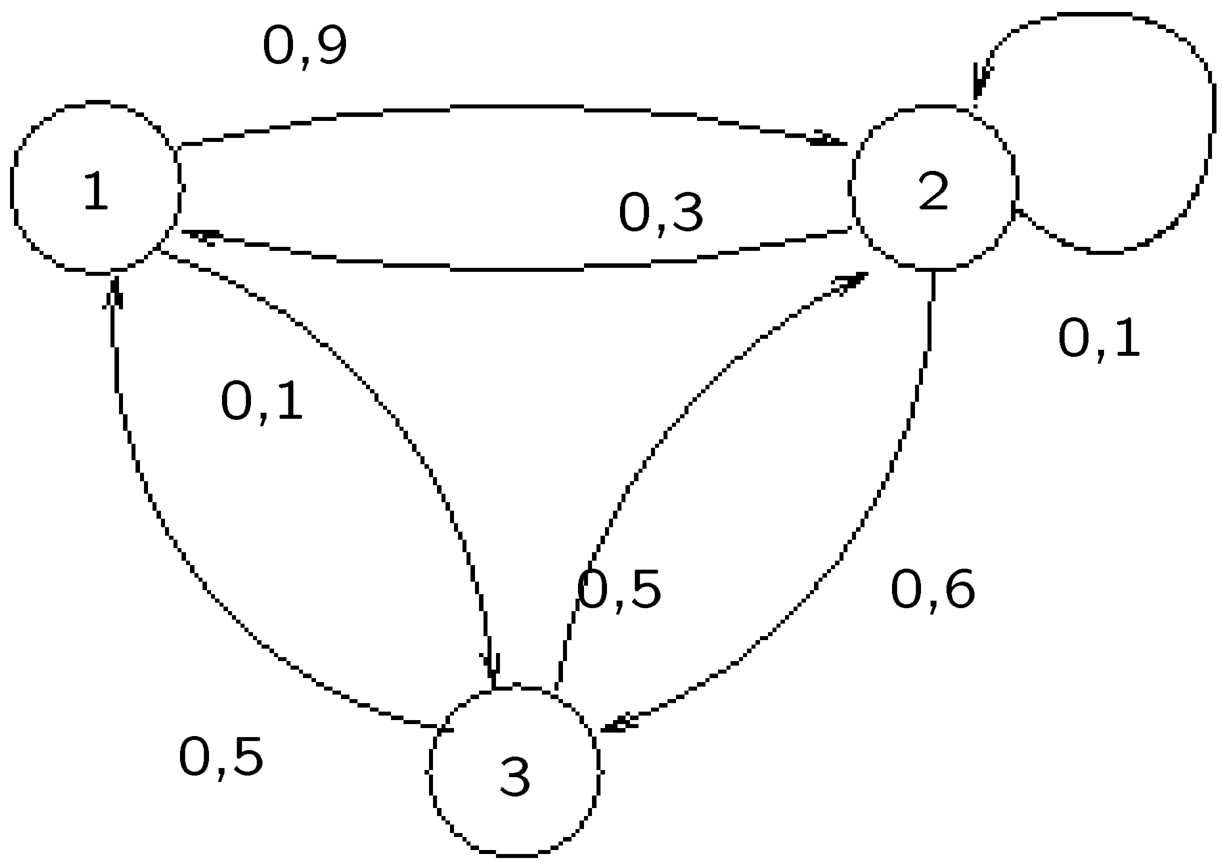

I'M TELLING YOU, HE KEEPS A LIST OF PEOPLE HE LIKES AND DISLIKES...
HE'S GOT ELVES SLAVING AWAY ALL YEAR DOING MENIAL TASKS FOR LITTLE OR NO PAY...
HE ALWAYS WEARS THE SAME VEST AND HE ONLY SHOWS UP ONCE A YEAR.
JORGE CHAM © 2011
SANTA CLAUS IS A PROFESSOR!!
"H_0 H_0 H_0"? IT'S THE NULL HYPOTHESIS!
WWW.PHDCOMICS.COM

Swaps

$$\begin{array}{c|ccccc|} & & A & & B & \\ \cline{2-6} & & \vdots & & \vdots & \\ u & \cdots & 1 & \cdots & 0 & \cdots \\ & & \vdots & & \vdots & \\ v & \cdots & 0 & \cdots & 1 & \cdots \\ & & \vdots & & \vdots & \\ \cline{2-6} \end{array} \quad \Longleftrightarrow \quad \begin{array}{c|ccccc|} & & A & & B & \\ \cline{2-6} & & \vdots & & \vdots & \\ u & \cdots & 0 & \cdots & 1 & \cdots \\ & & \vdots & & \vdots & \\ v & \cdots & 1 & \cdots & 0 & \cdots \\ & & \vdots & & \vdots & \\ \cline{2-6} \end{array}$$

- A **swap box** of $\boldsymbol{D}$ is a 2-by-2 combinatorial sub-matrix that is either *diagonal* or *anti-diagonal*
- A **swap** turns diagonal swap box into anti-diagonal, or vice versa
- **Theorem** [Ryser '57]. If $\boldsymbol{A}, \boldsymbol{B} \in M(\boldsymbol{r}, \boldsymbol{c})$, then $\boldsymbol{A}$ is reachable from $\boldsymbol{B}$ with a finite number of swaps

Example Markov chain



$$P = \begin{pmatrix} 0 & 9/10 & 1/10 \\ 3/10 & 1/10 & 6/10 \\ 1/2 & 1/2 & 0 \end{pmatrix}$$

The Metropolis Algorithm

- The **Metropolis algorithm** is a general technique to transform any irreducible Markov chain into a time-reversible chain with a required stationary distribution
  - A Markov chain is *time-reversible* if $\pi_i \boldsymbol{P}_{ij} = \pi_j \boldsymbol{P}_{ji}$
- Let $N(x)$, N, and M be as in previous slide, and let $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ be the desired stationary distribution.
  - Let

$$\mathbf{P}_{xy} = \begin{cases} 1/M \min\{1, \pi_y/\pi_x\} & \text{if } x \neq y \text{ and } y \in N(x), \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x), \\ 1 - \sum_{y \neq x} \mathbf{P}_{xy} & \text{if } x = y. \end{cases}$$

  - If the chain is aperiodic and irreducible, the stationary distribution is the desired one

Local Changes

- One-element changes
  - Replace a value
  - Add another value
- Four-element changes
  - Rotate
    - If a = a’ and b = b’, equals to swap
  - Mask
    - Preserves row and column sums

$$
\begin{array}{c|ccccc|}
 & & j_1 & & j_2 & \\
\hline
 & & \vdots & & \vdots & \\
i_1 & \dots & a & \dots & b & \dots \\
 & & \vdots & & \vdots & \\
i_2 & \dots & b' & \dots & a' & \dots \\
 & & \vdots & & \vdots & \\
\hline
\end{array}
\quad \Longrightarrow \quad
\begin{array}{c|ccccc|}
 & & j_1 & & j_2 & \\
\hline
 & & \vdots & & \vdots & \\
i_1 & \dots & b' & \dots & a & \dots \\
 & & \vdots & & \vdots & \\
i_2 & \dots & a' & \dots & b & \dots \\
 & & \vdots & & \vdots & \\
\hline
\end{array}
$$

Rotate

$$
\begin{array}{c|ccccc|}
 & & j_1 & & j_2 & \\
\hline
 & & \vdots & & \vdots & \\
i_1 & \dots & +\alpha & \dots & -\alpha & \dots \\
 & & \vdots & & \vdots & \\
i_2 & \dots & -\alpha & \dots & +\alpha & \dots \\
 & & \vdots & & \vdots & \\
\hline
\end{array}
$$

Mask

Ojala et al. 2009

Finding the MaxEnt Distribution

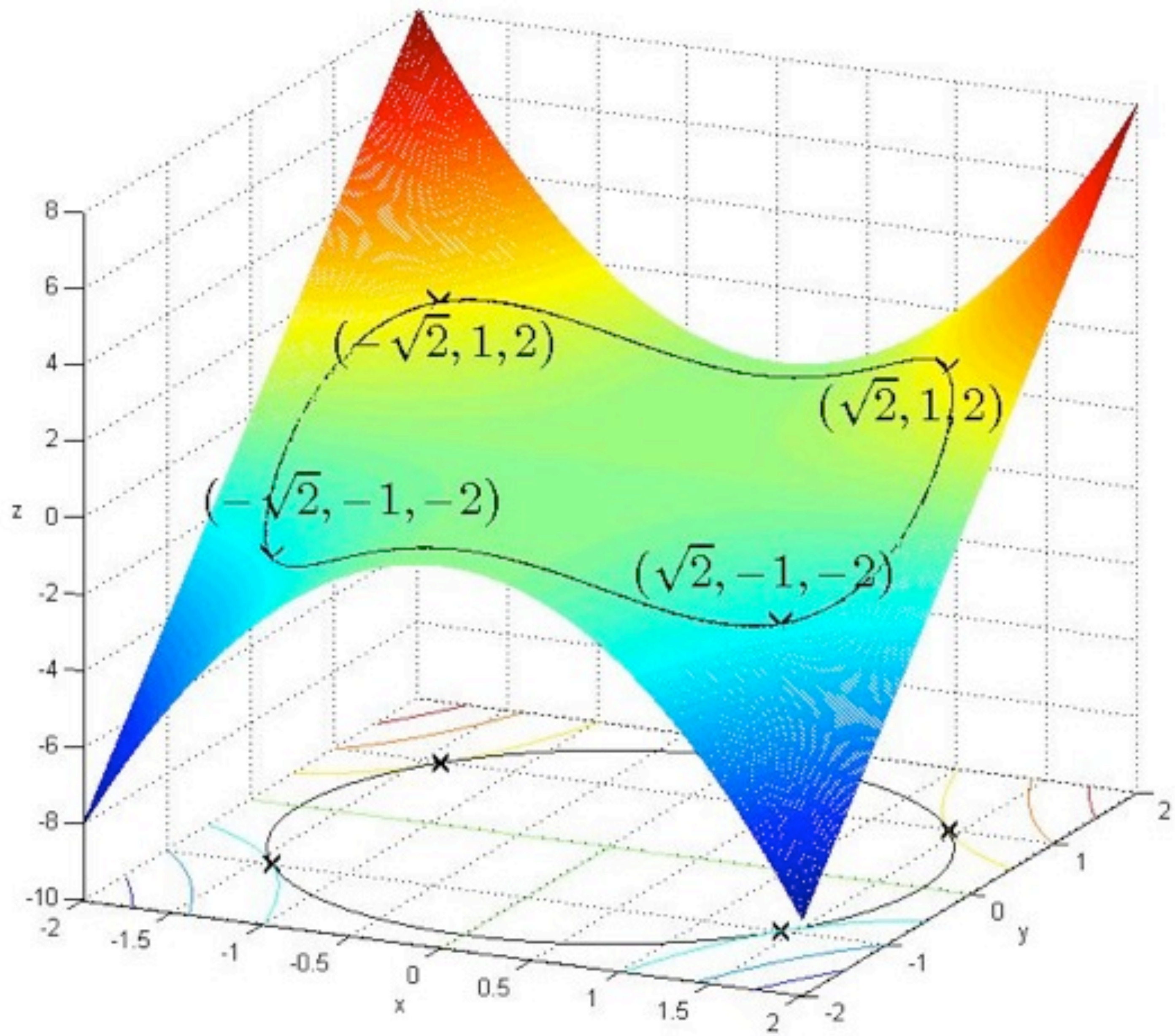
- Finding the MaxEnt distribution is a convex program with linear constraints

$$
\begin{aligned}
\max_{\Pr(\mathbf{x})} \quad & -\textstyle\sum_{\mathbf{x}} \Pr(\mathbf{x}) \log \Pr(\mathbf{x}) \\
\text{s.t.} \quad & \textstyle\sum_{\mathbf{x}} \Pr(\mathbf{x}) f_i(\mathbf{x}) = d_i \qquad \text{for all } i \\
& \textstyle\sum_{\mathbf{x}} \Pr(\mathbf{x}) = 1
\end{aligned}
$$

- Can be solved, e.g., using the Lagrange multipliers

Example

minimize $f(x,y) = x^2y$
subject to $g(x,y) = x^2 + y^2 = 3$



$$L(x,y,\lambda) = x^2y + \lambda(x^2 + y^2 - 3)$$

$$\frac{\partial L}{\partial x} = 2xy + 2\lambda x = 0$$

$$\frac{\partial L}{\partial y} = x^2 + 2\lambda y = 0$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 3 = 0$$

Solution: $x = \pm\sqrt{2}$, $y = -1$

MaxEnt Models for Tiling

- The Tiling problem
  - Binary data, aim to find fully monochromatic submatrices
- Constraints: the *expected* row and column margins

$$\sum_{\mathbf{D}\in\{0,1\}^{n\times m}} \Pr(\mathbf{D}) \left(\sum_{j=1}^{m} d_{ij} \right) = r_i$$

$$\sum_{\mathbf{D}\in\{0,1\}^{n\times m}} \Pr(\mathbf{D}) \left(\sum_{i=1}^{n} d_{ij} \right) = c_j$$

  - Note that these are in the correct form

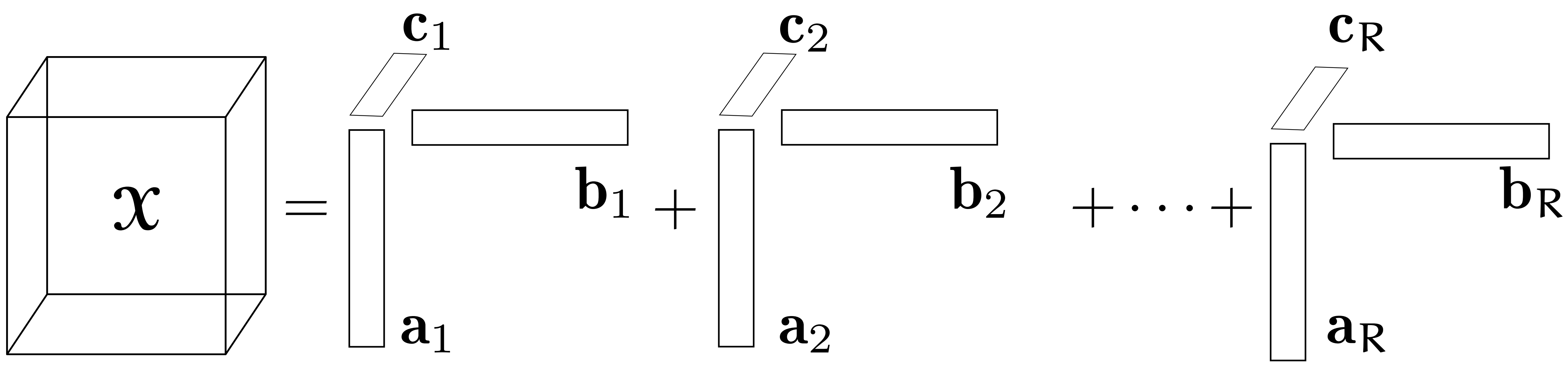
De Bie 2010

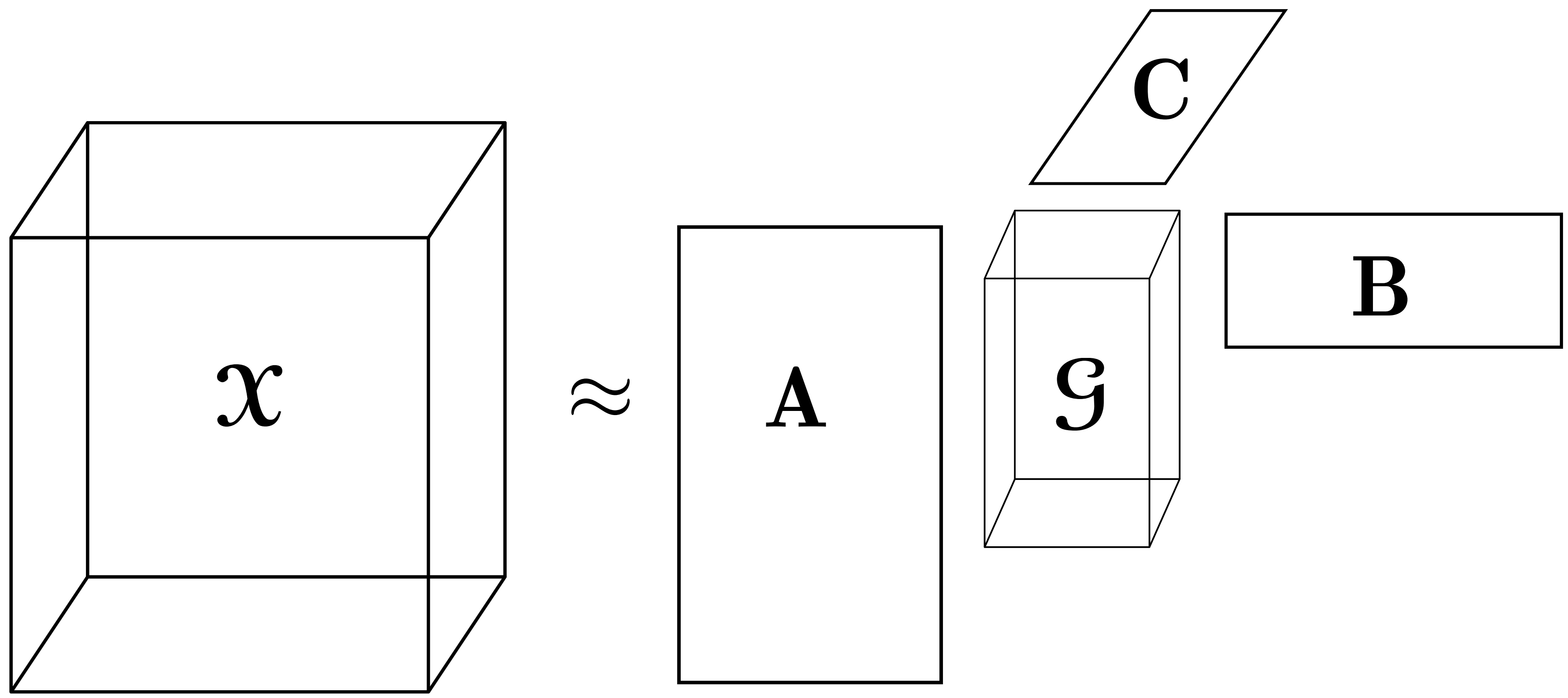
Preserving Means and Variances

- To preserve row and column means and variances, we need to constraint
  - Row and column sums
  - Row and column sums-of-squares
- After solving the MaxEnt equation, we again get that the MaxEnt distribution for **D** is a product of probabilities for *dij*
  - $\Pr(d_{ij}) \sim \mathcal{N}\left(-\frac{\lambda_i^r+\lambda_j^c}{2(\mu_i^r+\mu_j^c)}, \left(2(\mu_i^r+\mu_j^c)\right)^{-1/2}\right)$
    - λs are Lagrange multipliers associated with the constraints on sums
    - μs are Lagrange multipliers associated with the constraints on sums-of-squares

Topic IV: Tensors

- Tensors are cool.





Feedback on Topic III Essays

- Generally, quality's still high
- MaxEnt seemed to cause problems to you
  - Very briefly discussed
  - Sometimes mixed with other approaches using maximum entropy
- Both swap-based and MaxEnt-based methods can handle numerical data
  - Constraining row and column margins makes only sense if row and column margins make sense

Exam Information

- 19 February (Tuesday)
- Oral exam
- Room 021 at MPII building (E1.4)
- Time frame: 10 am – 6 pm
  - If you have constraints within this time frame, send me email
  - About 20 min per student
- I will ask questions on one or two topic areas
  - You can veto one proposed topic are—but only one