Generalized Matrix Factorizations

as a

Unifying Framework

for

Pattern Set Mining

Complexity Beyond Blocks

Pauli Miettinen

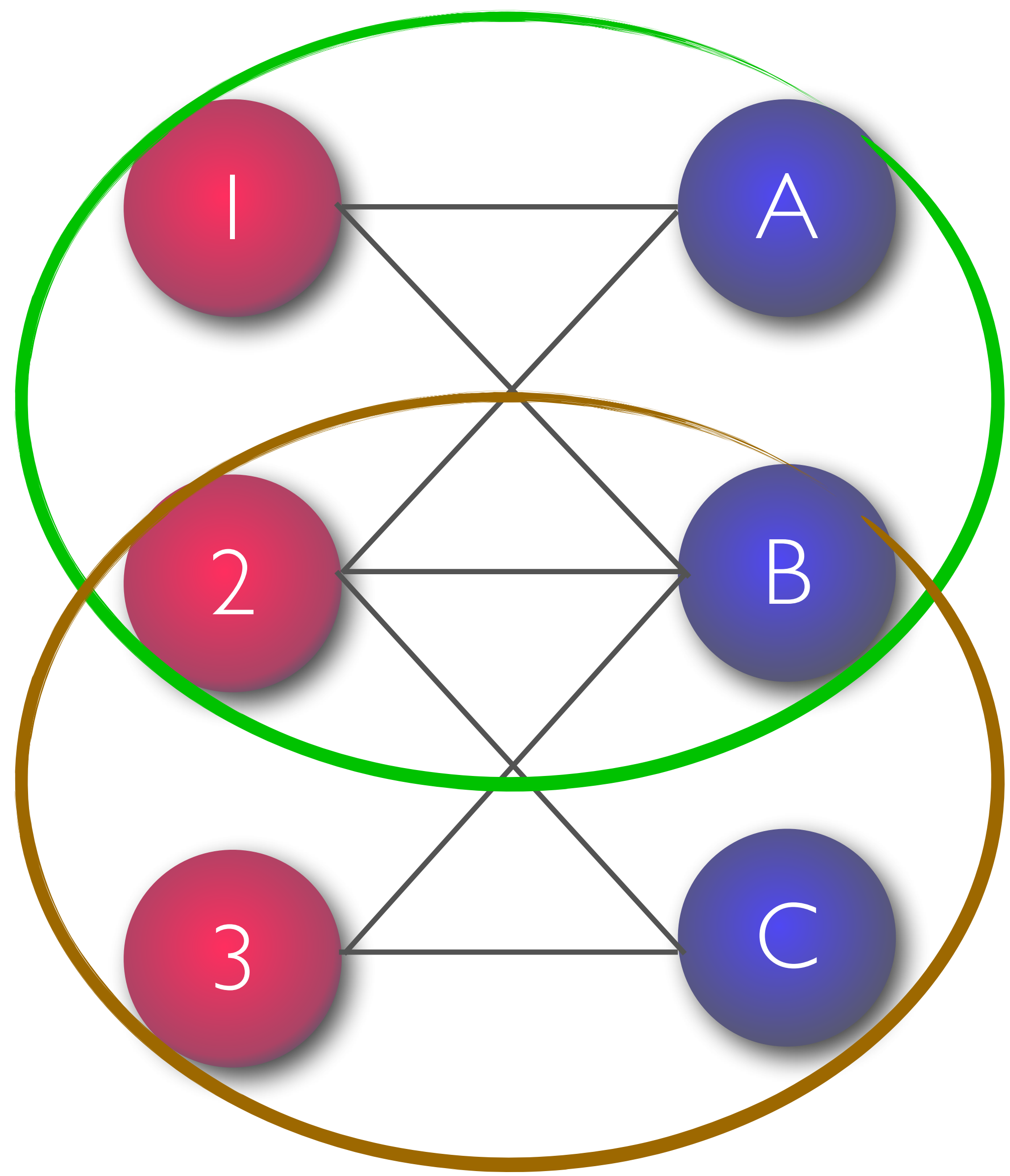
10 September 2015

max planck institut
informatik

Community detection

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

$$= \begin{array}{c} 1 \\ 2 \\ 3 \end{array}\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \circ \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

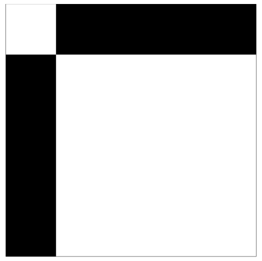
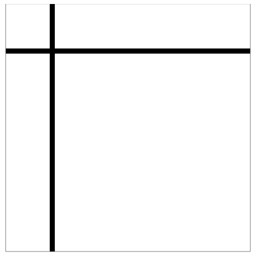
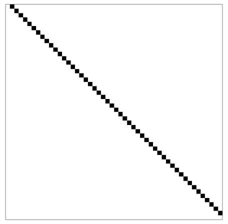


Rank-1 matrices

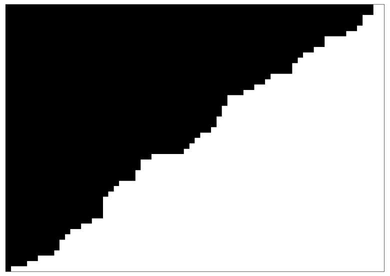
- (Bi-)cliques are rank-1 submatrices
  - Collection of rank-1 submatrices summarizes the graph using its cliques
- Matrix factorizations express the (complex) input as a sum of rank-1 matrices
  - $\boldsymbol{AB} = \boldsymbol{a}_1 \boldsymbol{b}_1^T + \boldsymbol{a}_2 \boldsymbol{b}_2^T + \cdots + \boldsymbol{a}_k \boldsymbol{b}_k^T$
- Matrix factorizations summarize complex data using simple patterns

Beyond blocks

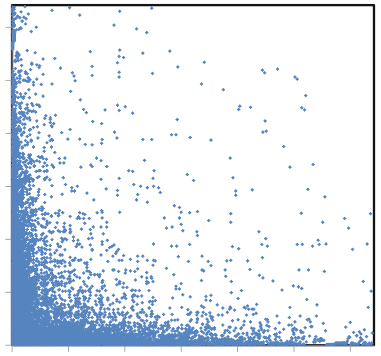
- Cliques are not the only (graph) patterns
- Biclique cores, stars, chains
  - Koutra et al., SDM ’14.

- Nested graphs
  - e.g. Junttila ’11, Kötter et al., WWW ’15



- Hyperbolic communities
  - Araujo et al., ECML PKDD ’14



Limitations of matrix factorization

- The matrix-factorization language is useful
  - Recycle ideas, approaches, and results
- But the other patterns are not rank-1 matrices
  - It is not easy to express a collection of nested matrices as a matrix factorization

Generalized outer products

- Rank-1 matrix = outer product of two vectors
  - $\boldsymbol{A} = \boldsymbol{x}\boldsymbol{y}^T$
- Define **generalized outer product**

$$o(\boldsymbol{x}, \boldsymbol{y}, \theta) \in \mathbb{R}^{n \times m}$$

Vectors Parameters

  - $o(\boldsymbol{x}, \boldsymbol{y}, \theta)_{ij} = x_i y_j$ or 0

Example: biclique core

$$o\left(\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, \begin{bmatrix}1 & 1 & 1 & 1 & 1\end{bmatrix}, \{1,2\}\right) = \begin{bmatrix}0 & 0 & 1 & 1 & 1\\0 & 0 & 1 & 1 & 1\\1 & 1 & 0 & 0 & 0\\1 & 1 & 0 & 0 & 0\\1 & 1 & 0 & 0 & 0\end{bmatrix}$$

Rows that belong to the pattern

Columns that belong to the pattern

The core

Example: nested matrix

Step function

$$o\left(\begin{bmatrix}1\\1\\1\\0\\1\\1\end{bmatrix},\ [\,1\ 1\ 1\ 1\ 1\,],\ [\,1\ 2\ 2\ 5\ 6\,]\right)=\begin{bmatrix}1&1&1&1&1\\0&1&1&1&1\\0&0&0&1&1\\0&0&0&0&0\\0&0&0&1&1\\0&0&0&0&1\end{bmatrix}$$

Generalized decompositions

- Recall, $\boldsymbol{X} \approx \boldsymbol{AB} = \boldsymbol{a}_1\boldsymbol{b}_1^T + \boldsymbol{a}_2\boldsymbol{b}_2^T + \cdots + \boldsymbol{a}_k\boldsymbol{b}_k^T$ is a decomposition of $\boldsymbol{X}$
- The **generalized decomposition** of $\boldsymbol{X}$ is

$$\boldsymbol{X} \approx \boldsymbol{F}_1 \boxplus \boldsymbol{F}_2 \boxplus \cdots \boxplus \boldsymbol{F}_k, \quad \boldsymbol{F}_i = o(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_i)$$

  - $\boxplus$ is the addition in the underlying algebra
    - sum, AND, OR, XOR, …

o-induced rank

- The smallest k s.t. $\boldsymbol{X} = \boldsymbol{F}_1 \boxplus \ldots \boxplus \boldsymbol{F}_k$ is the **o-induced rank** of $\boldsymbol{X}$
  - Analogous to the standard (Schein) rank
- Can be infinite if the matrix cannot be expressed (exactly) with that kind of outer products
  - If the outer product can generate a matrix that has exactly one nonzero at arbitrary position, it’s induced rank is always bounded

Decomposability

- Outer product o is **decomposable** (to f) if, for some f, $o(\boldsymbol{x}, \boldsymbol{y}, \theta)_{ij} = f(x_i, y_j, i, j, \theta)$
  - Then we have
$$x_{ij} = \bigboxplus_{l=1}^{k} f(x_{il}, y_{lj}, i, j, \theta)$$
  as in standard matrix multiplication

Nice work, but … why?

- So, we can express complex patterns using some weird functions
- What’s the advantage?
- Using the common language, it’s easy to see how some results (and techniques) can be generalized as well

How hard can it be…

- …to find the maximum-circumference pattern?
- I.e. given $\boldsymbol{A}$, find $\boldsymbol{x}$, $\boldsymbol{y}$, and θ s.t. $o(\boldsymbol{x}, \boldsymbol{y}, \theta) \in \boldsymbol{A}$ and you maximize $|\boldsymbol{x}| + |\boldsymbol{y}|$
  - If o is hereditary and the pattern can have infinitely many distinct rows and columns, NP-hard
  - If there's only fixed number of distinct rows or columns, the problem is in P
  - If $\boldsymbol{x} = \boldsymbol{y}$ is required, then it's almost always NP-hard

How hard can it be...

- ...to select the smallest subset that gives an exact summarization?
- I.e. given a set $S = \{\boldsymbol{F}_i : \text{rank}(\boldsymbol{F}_i) = 1\}$, $\boxplus_{\boldsymbol{F} \in S} \boldsymbol{F} = \boldsymbol{X}$, find the the smallest $C \subseteq S$ s.t. $\boxplus_{\boldsymbol{F} \in C} \boldsymbol{F} = \boldsymbol{X}$
  - NP-hard for $\boxplus \in$ {AND, OR, XOR}
  - hard to approximate within $\ln(n)$ for OR and within superpolylogarithmic for XOR

How hard can it be…

- …to compute the rank?
- Well, that depends… (on the underlying algebra)
- Doesn’t depend (only) on the outer product
  - E.g. normal outer product is NP-hard for OR but in P for XOR

How hard can it be…

- …to find the decomposition of fixed size that minimizes the error?
- NP-hard if computing the rank is
- NP-hard to approximate to within superpolylogarithmic factors for OR and XOR

Conclusions

- Matrix factorizations are sort-of mixture models
  - Present complex data as an aggregate of simpler parts
- Generalized outer products let us represent more than just cliques as ”rank-1” matrices
  - And allow to generalize many results from cliques

Future

- More work is needed to see what is the correct level of generality for the outer products
- Results for numerical data?
- Framework with no users isn't very useful…

Thank You!

Questions?