Computational Bottlenecks in Graph Mining

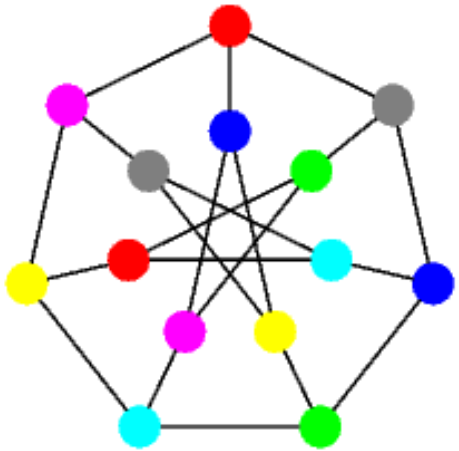
Karsten Borgwardt

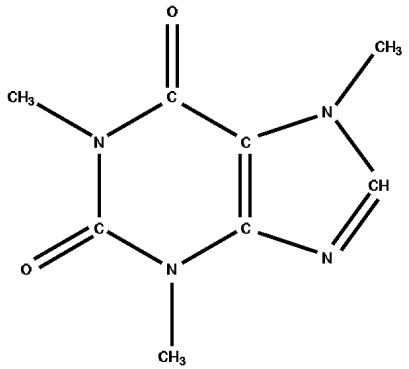
Machine Learning and Computational Biology Research Group
Max Planck Institute for Intelligent Systems & Max Planck Institute for Developmental Biology, Tübingen, Germany

MLG, San Diego
August 21, 2011

Graph mining is everywhere

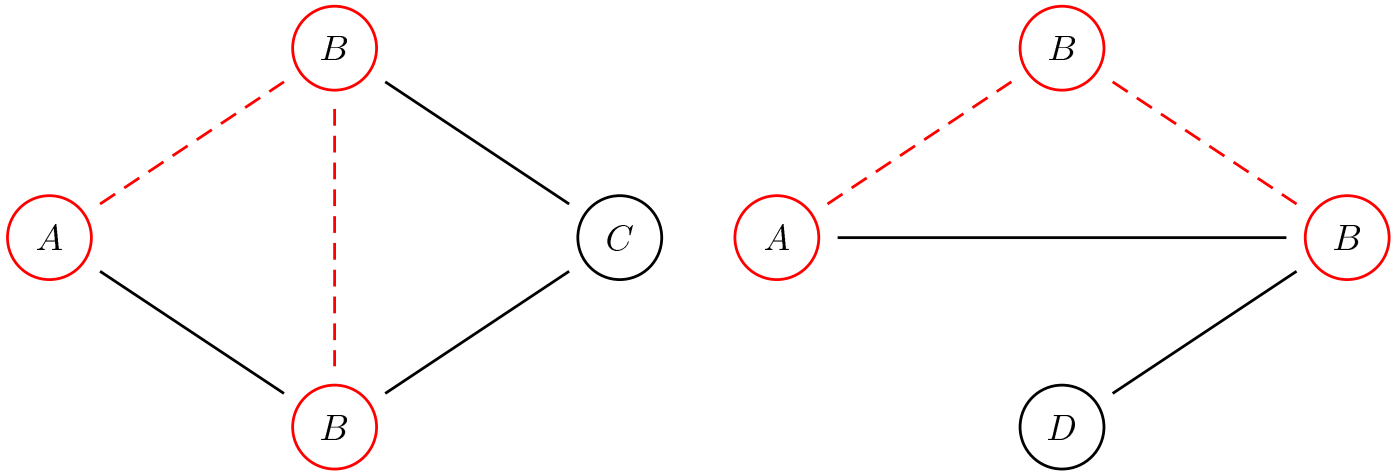
- Graphs are everywhere
  - Bioinformatics
  - Social Network Analysis
  - Natural Language Processing
- Hot topics in databases/data mining
  - Frequent subgraph mining
  - Dense subgraph mining
  - Graph indexing and search
- Recent trends
  - Data: Growing size of graphs is a challenge for classic approaches
  - Methods: Kernel Machine Learning approaches to graph mining





Problem 1: Measure the similarity of two graphs

- How similar are two graphs?
  - How similar is their structure?
  - How similar are their node labels and edge labels?



- Applications
  - Function prediction for molecules and proteins
  - Change detection in networks of friendship
  - Comparison of semantic structures in NLP

Graph comparison

1. Graph isomorphism and subgraph isomorphism checking
   - Exact match
   - Exponential runtime
2. Graph edit distances
   - Involves definition of a cost function
   - Typically subgraph isomorphism as intermediate step
3. Topological descriptors
   - Lose some of the structural information represented by the graph **or**
   - Exponential runtime effort
4. Graph kernels (Gärtner et al, 2003; Kashima et al. 2003)
   - Goal 1: Polynomial runtime
   - Goal 2: Applicable to large graphs
   - Goal 3: Applicable to graphs with attributes

Speeding up Graph kernels

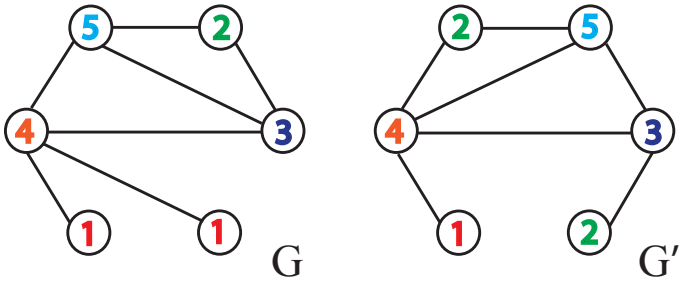
- Walks (NIPS 2006c, JMLR 2010)
  - Defined by Gärtner et al. and Kashima et al. in 2003
  - Slow: $O(n^6)$ where n is the number of nodes in G and G'
  - We use Sylvester equations and Kronecker products to compute the same kernel in $O(n^3)$
- Shortest paths (ICDM 2005)
  - Literature claimed there was no obvious way to define a graph kernel based on shortest paths.
  - We defined a graph kernel comparing the lengths of shortest paths in two graphs.
  - Wiener Index from chemoinformatics is an instance of this kernel.

Speeding up Graph kernels

- Subgraphs of limited size k (AISTATS 2009)
  - Suggested as 'graphlets' (k=4) by Przulj (Bioinformatics, 2007)
  - Corresponding graph kernel scales as $O(n^8)$
  - We turn this into a fast kernel on unlabeled graphs $O(nd^{k-1})$.
- Results from group theory (ICML 2008, 2009)
  - Use concepts from group theory to derive feature vector representation of graphs
  - computable in $O(n^3)$
- Unresolved question:
  - How to compute kernels efficiently on large, labeled graphs?

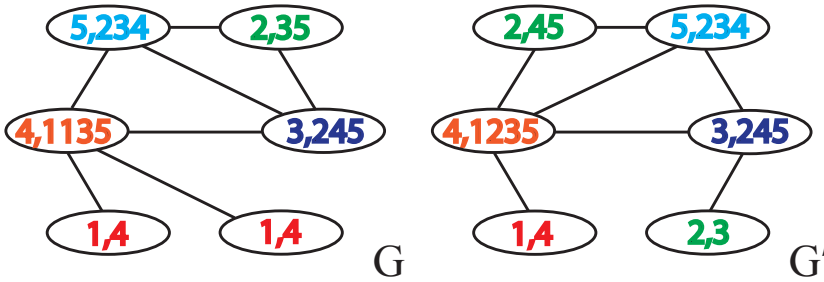
Weisfeiler-Lehman kernel (Shervashidze and Borgwardt, NIPS 2009)

Given labeled graphs G and G′



a

1st iteration
Result of steps 1 and 2: multiset-label determination and sorting



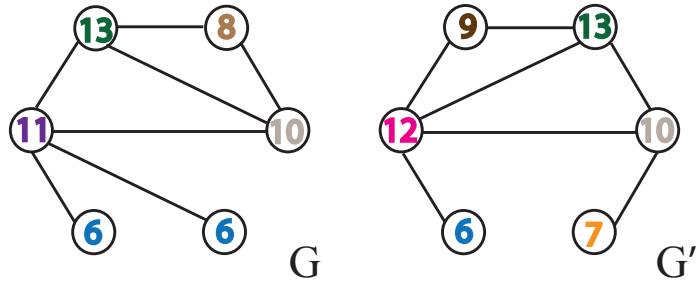
b

1st iteration
Result of step 3: label compression

1,4 → 6		3,245 → 10	
2,3 → 7		4,1135 → 11	
2,35 → 8		4,1235 → 12	
2,45 → 9		5,234 → 13	

c

1st iteration
Result of step 4: relabeling



d

Weisfeiler-Lehman kernel (Shervashidze and Borgwardt, NIPS 2009)

End of the 1st iteration
Feature vector representations of G and G′

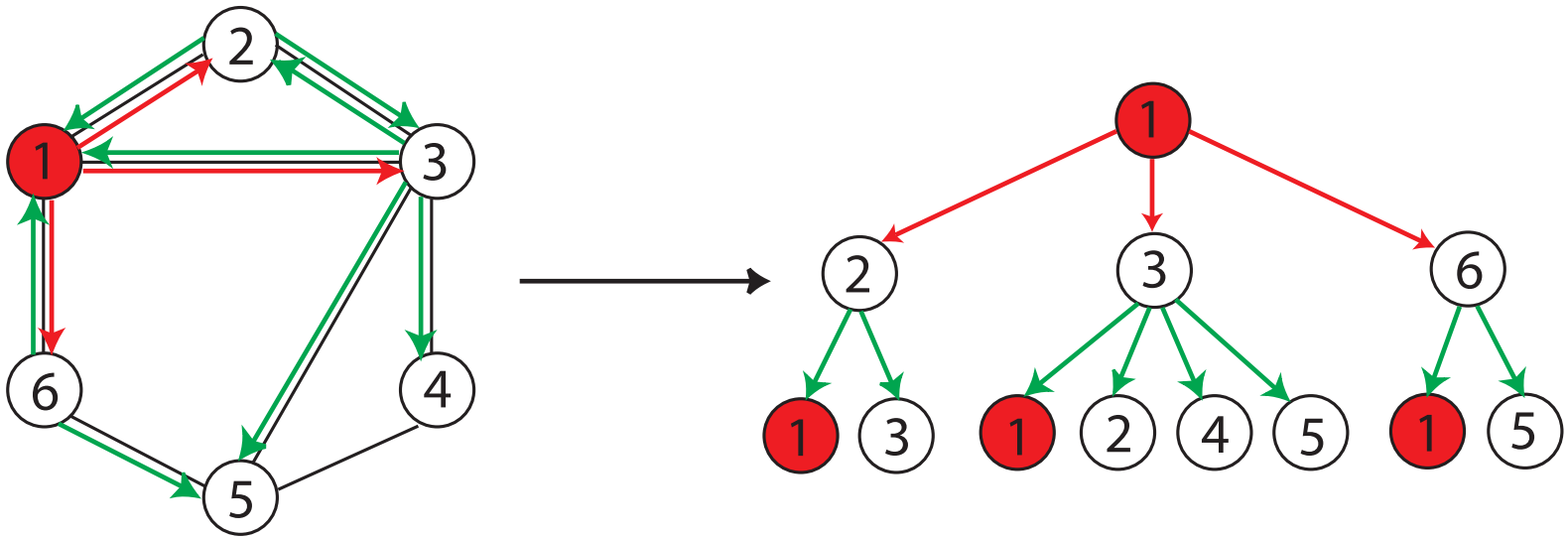
$$\varphi^{(1)}_{WLsubtree}(G) = (2, 1, 1, 1, 1, 2, 0, 1, 0, 1, 1, 0, 1)$$

$$\varphi^{(1)}_{WLsubtree}(G') = (\underbrace{1, 2, 1, 1, 1}_{\text{Counts of original node labels}}, \underbrace{1, 1, 0, 1, 1, 0, 1, 1}_{\text{Counts of compressed node labels}})$$

$$k^{(1)}_{WLsubtree}(G,G') = \langle \varphi^{(1)}_{WLsubtree}(G), \varphi^{(1)}_{WLsubtree}(G') \rangle = 11.$$

e

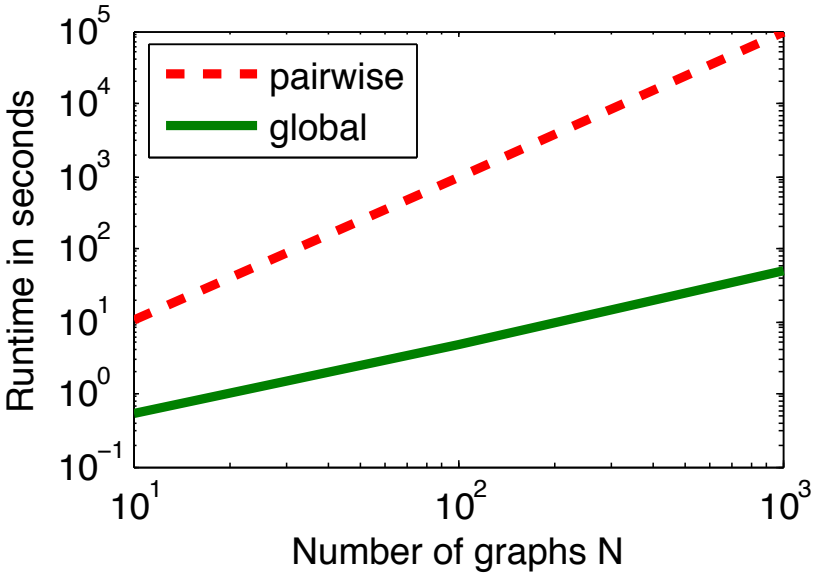
Subtree-like patterns

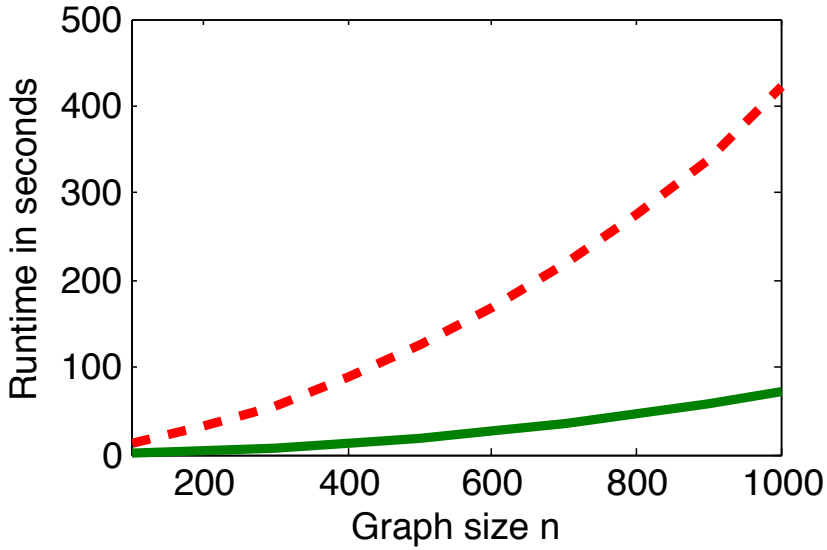


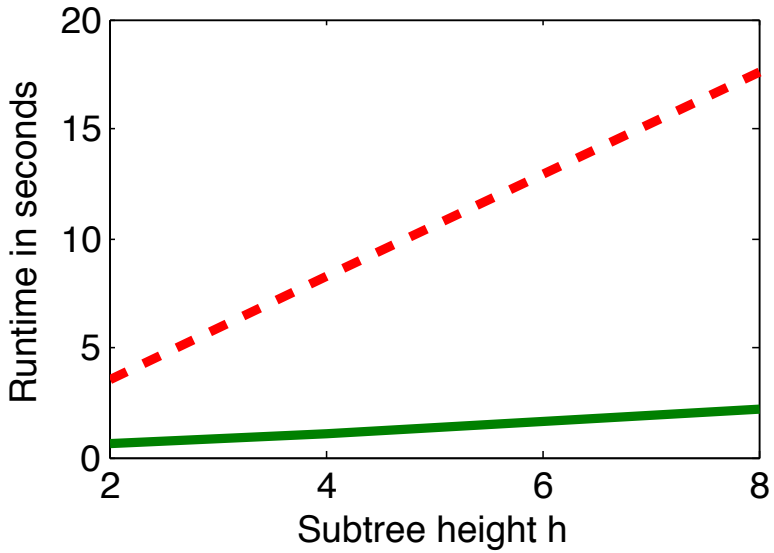
Weisfeiler-Lehman kernel: Theoretical runtime properties

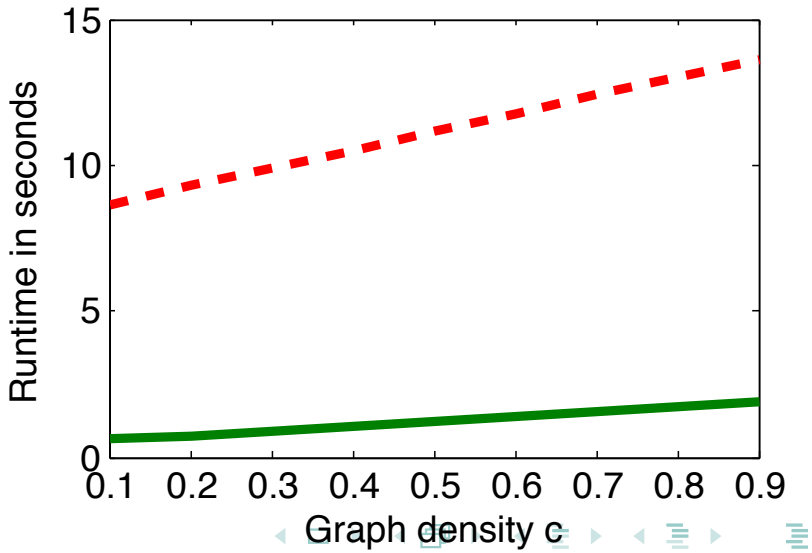
- Fast Weisfeiler-Lehman kernel (NIPS 2009 and JMLR 2011)
  - **Algorithm**: Repeat the following steps h times
    1. Sort: Represent each node v as sorted list L_v of its neighbors ($O(m)$)
    2. Compress: Compress this list into a hash value $h(L_v)$ ($O(m)$)
    3. Relabel: Relabel v by the hash value $h(L_v)$ ($O(n)$)
- Runtime analysis
  - per graph pair: Runtime $O(m\ h)$
  - for N graphs: Runtime $O(N\ m\ h + N^2\ n\ h)$ (naively $O(N^2\ m\ h)$)

Weisfeiler-Lehman kernel: Empirical Runtime properties

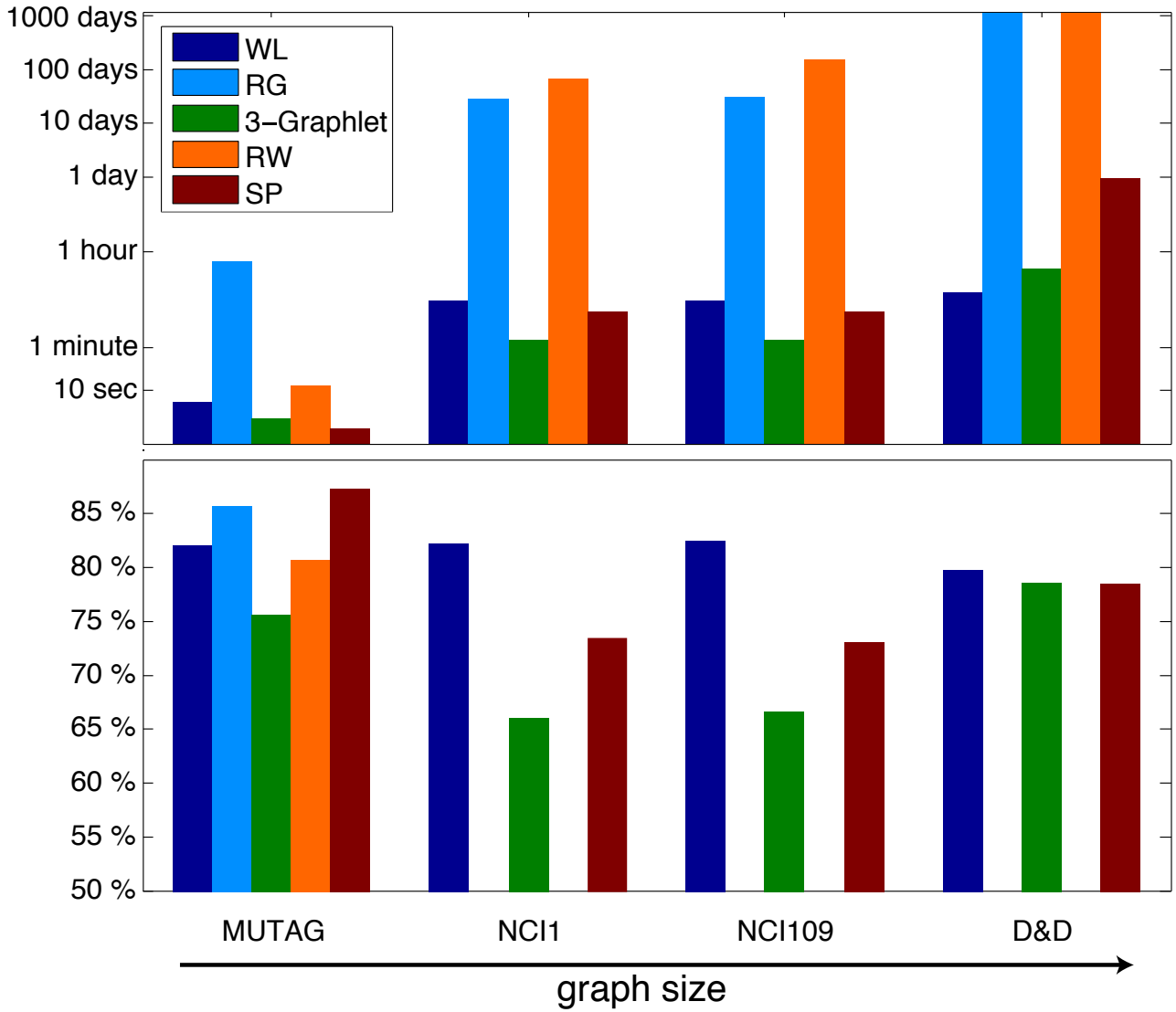




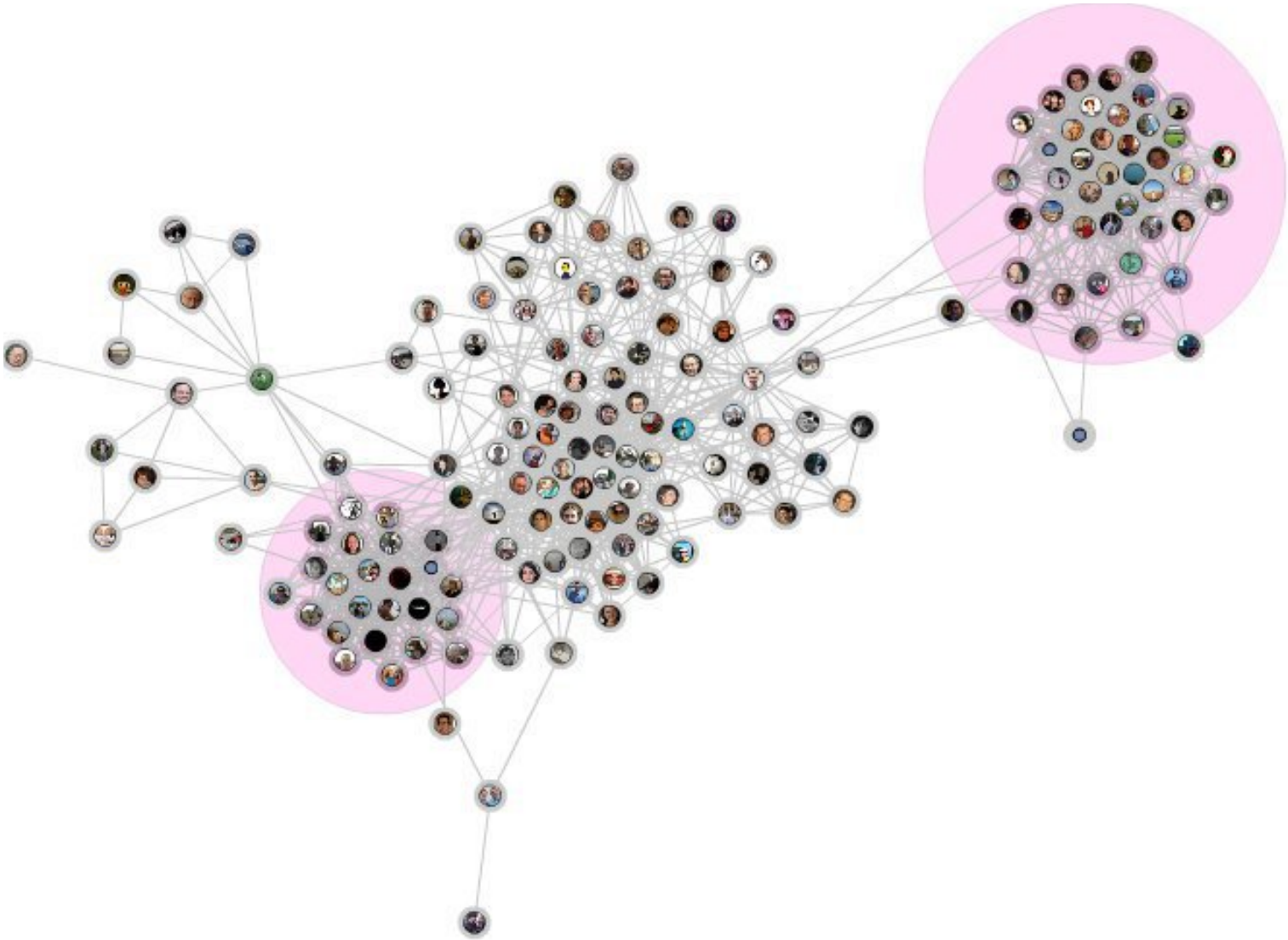




Weisfeiler-Lehman kernel: Runtime and accuracy



Problem 2: Find the most similar nodes in a graph



created by Social Graph

The lightbulb approach (Paturi et al., COLT 1989)

Maximum correlation

- The lightbulb algorithm tackles the maximum correlation problem on an $m \times n$ matrix A with binary entries:

$$\operatorname*{argmax}_{i,j} |\rho_A(x_i, x_j)|. \tag{1}$$

Quadratic runtime algorithm

- The problem can be solved by naive enumeration of all n^2 possible solutions.

The lightbulb approach

Lightbulb algorithm

1. Given a binary matrix A with m rows and n columns.
2. Repeat l times:
   - Sample k rows
   - Increase a counter for all pairs of columns that match on these k rows.
3. The counters divided by l give an estimate of the correlation $P(x_i = x_j)$.

Subquadratic runtime

- With probability $1 - n^{-\alpha}$, the lightbulb algorithm retrieves the most correlated pair in $O(\alpha\ n^{1+\frac{\ln p_1}{\ln q_2}}\ \ln^2 n) = O(n(\alpha\ n^{\frac{\ln p_1}{\ln q_2}}\ \ln^2 n))$.

Difference between the node-pair and the lightbulb setting

Discrepancies

- Node attributes are non-binary in general
- Pearson's correlation coefficient

Locality sensitive hashing (Charikar, 2002)

Given a collection of vectors in $\mathbb{R}^m$ we choose a random vector $\vec{r}$ from the m-dimensional Gaussian distribution. Corresponding to this vector $\vec{r}$, we define a hash function $h_{\vec{r}}$ as follows:

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{r} \cdot \vec{u} \geq 0 \\ 0 & \text{if } \vec{r} \cdot \vec{u} < 0 \end{cases} \tag{2}$$

Theorem

For vectors $\vec{v}, \vec{u}$, $Pr[h_{\vec{r}}(\vec{u}) = h_{\vec{r}}(\vec{v})] = 1 - \frac{\theta(\vec{u}, \vec{v})}{\pi}$, where θ is the angle between the two vectors.

Pearson's correlation coefficient

Link between correlation and cosine

Karl Pearson defined the correlation of 2 vectors $\vec{v}, \vec{u}$ in $\mathbb{R}^m$ as

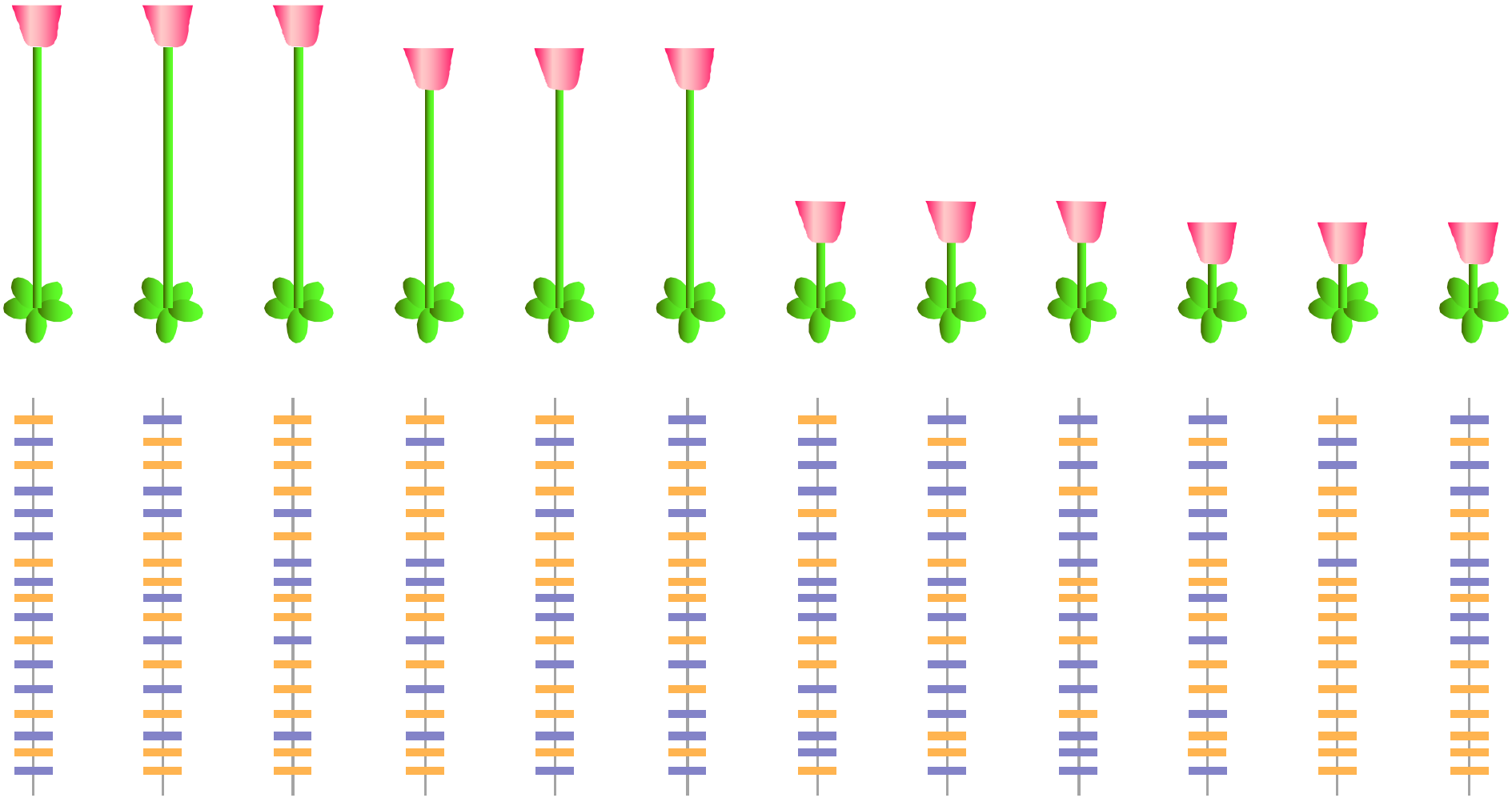
$$\rho = \frac{cov(\vec{v}, \vec{u})}{\sigma_v \sigma_u}, \tag{3}$$

that is the covariance of the two vectors divided by their standard deviations. An equivalent geometric way to define it is:

$$\rho = cos(\vec{v} - \bar{v}, \vec{u} - \bar{u}), \tag{4}$$

where $\bar{v}$ and $\bar{u}$ are the mean value of $\vec{u}$ and $\vec{v}$, respectively.

Genome-wide association mapping



by courtesy of D. Weigel

Challenges in two-locus mapping

Scale of the problem

- Typical datasets include order $10^5 - 10^7$ SNPs.
- Hence we have to consider order $10^{10} - 10^{14}$ SNP pairs.
- Enormous multiple hypothesis testing problem.
- Enormous computational runtime problem.

Our contribution

- We assume binary phenotypes (cases and controls).
- Genotypes may be homozygous or heterozygous.
- We assume m individuals with n SNPs each.
- We define an algorithm that rapidly detects epistatic interactions in a runtime subquadratic in n (Achlioptas et al., KDD 2011).

Common approaches in the literature

Exhaustive enumeration

- Only with special hardware such as GPU implementations: EPIBLASTER (Kam-Thong et al., EJHG 2010)

Filtering approaches

- Statistical criterion, e.g. SNPs with large main effect (Zhang et al., 2007)
- Biological criterion, e.g. underlying PPI (Emily et al., 2009)

Index structure approaches

- fastANOVA, branch-and-bound on SNPs (Zhang et al., 2008)
- TEAM, efficient updates of contingency tables (Zhang et al., 2010)

Difference in correlation for epistasis detection

- We phrase epistasis detection as a difference in correlation problem:

$$\underset{i,j}{\operatorname{argmax}} \, |\rho_{cases}(x_i, x_j) - \rho_{controls}(x_i, x_j)|. \tag{5}$$

- Different degree of linkage disequilibrium of two loci in cases and controls

Difference in correlation

Theorem

- Given a matrix of cases A and a matrix of controls B of identical size.
- Finding the maximally correlated pair on

$$\begin{pmatrix} A & A \\ B & 1-B \end{pmatrix} \tag{6}$$

- and on

$$\begin{pmatrix} A & 1-A \\ B & B \end{pmatrix} \tag{7}$$

- is identical to

$$\underset{i,j}{\operatorname{argmax}} \, |\rho_A(x_i, x_j) - \rho_B(x_i, x_j)|. \tag{8}$$

The lightbulb epistasis algorithm

Algorithm

1. Binarize original matrices A_0 and B_0 into A and B by locality sensitive hashing.
2. Compute maximally correlated pair P_1 on $\begin{pmatrix} A & A \\ B & 1-B \end{pmatrix}$ via lightbulb.
3. Compute maximally correlated pair P_2 on $\begin{pmatrix} A & 1-A \\ B & B \end{pmatrix}$ via lightbulb.
4. Report the maximum of P_1 and P_2.

Experiments: Nordborg lab SNP dataset

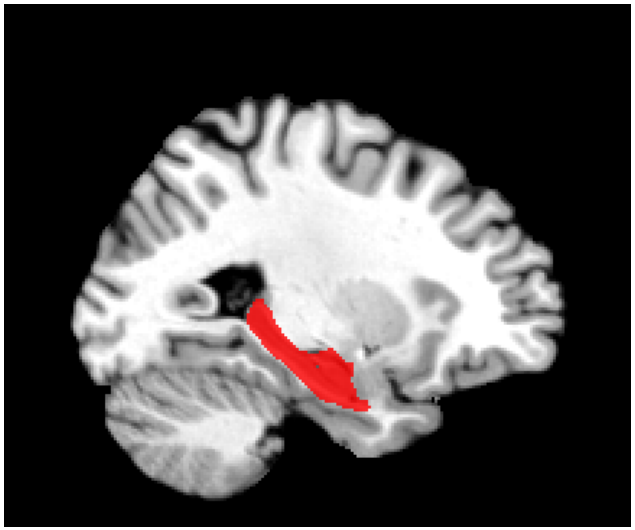
Results on Nordborg SNP dataset

# SNPs	Measurements	Pairs	Exponent	Speedup	Top 10	Top 100	Top 500	Top 1K
100,000	8,255,645	8,186,657	1.38	611	1.00	0.86	0.82	0.80
100,000	52,762,001	51,732,700	1.54	97	1.00	1.00	0.99	0.98

Runtime

- Runtime is empirically $O(n^{1.5})$.
- Epistasis detection on the human genome would require 1 day of computation on a typical desktop PC.

First finding



by P. Sämann

- 567 subjects
- 1,075,163 SNPs
- phenotype: Hippocampus volume
- genome-wide significant results ($p < 10^{-12}$)
- near genes involved in cell-cell signaling

Summary

Efficient graph comparison and node pair search

- We define kernels on graphs with discrete node labels, whose runtime is only linear in the number of edges m and the number of iterations h of the Weisfeiler-Lehman algorithm.
- We define a scheme to find the most correlated pair of nodes in a graph, which is subquadratic in the number of nodes n.
- A variant of this correlation search algorithm can be used to search for interacting genetic loci in subquadratic time.

Thank you

Group members:

- Nino Shervashidze
- Panagiotis Achlioptas
- Tony Kam-Thong
- Chloé-Agathe Azencott
- Barbara Rakitsch
- Limin Li
- Dominik Grimm
- Theofanis Karaletsos
- Christoph Lippert
- Oliver Stegle
- Hyokun Yun

Collaborators:

- F. Holsboer, MPI Psychiatry
- K. Mehlhorn, MPI Computer Science
- B. Müller-Myhsok, MPI Psychiatry
- B. Schölkopf, MPI-IS
- A. Smola, Yahoo! Research
- D. Weigel, MPI Dev. Biology

Sponsors:

- A.-v.-Humboldt (Chloé)
- DFG
- Microsoft Research Cambridge
- VW (Oliver)

Main references

- Nino Shervashidze and Karsten Borgwardt. **Fast subtree kernels on graphs**, NIPS 2009.
- Nino Shervashidze *et al.* **Weisfeiler-Lehman graph kernels**, JMLR 2011.
- Panagiotis Achlioptas *et al.* **Two-locus association mapping in subquadratic runtime**, KDD 2011.
- Tony Kam-Thong *et al.* **Epistasis detection on quantitative phenotypes by exhaustive enumeration using GPUs**, ISMB 2011.
- Tony Kam-Thong *et al.* **EPIBLASTER-Fast exhaustive two-locus epistasis detection strategy using graphical processing units**, European Journal of Human Genetics, 2011.