UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Efficient Anti-community Detection in Complex Networks

Sebastian Lackner[1], Andreas Spitz[1], Matthias Weidemüller[2], and Michael Gertz[1]

30th International Conference on Scientific and Statistical Database Management (SSDBM)
July 9 - 11, 2018, Bolzano-Bozen, Italy

[1]Database Systems Research Group, Heidelberg University, Germany
{lackner,spitz,gertz}@informatik.uni-heidelberg.de

[2]Quantum Dynamics of Atomic and Molecular Systems Group, Heidelberg University, Germany
weidemueller@uni-heidelberg.de

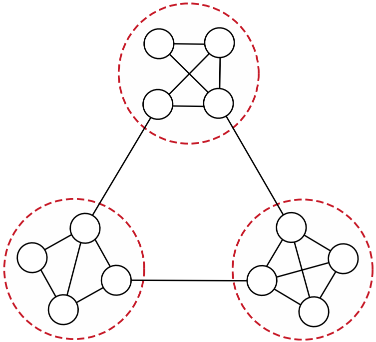
Community Structure

Many networks contain community structures.

Communities are characterized by

- many internal edges
- few external edges

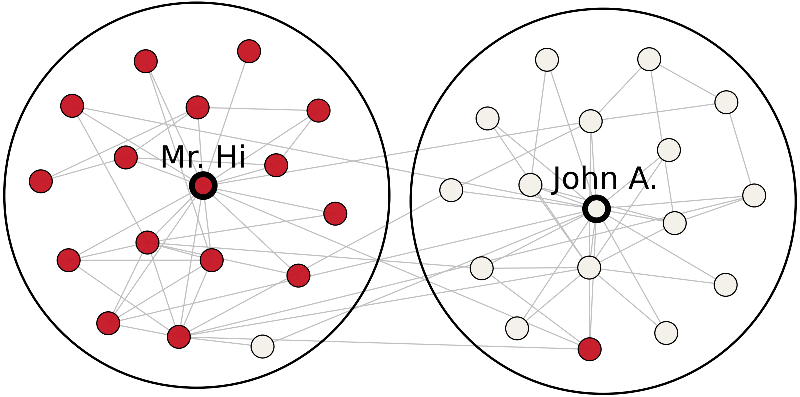
(*generalization of cliques*)



Applications in sociology, computer science, physics, biology, . . . [For10]

Zachary's Karate Club Network

$|V| = 34,$
$|E| = 156$



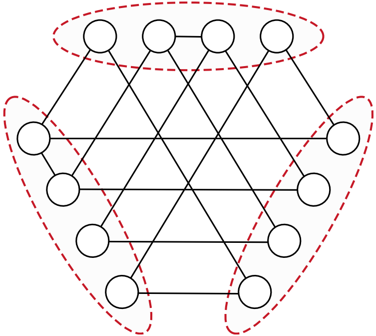
Communities in *Zachary's karate club* network [Zac77].
Colors denote membership after the fission of the club.

Anti-community Structure

Anti-Communities are characterized by

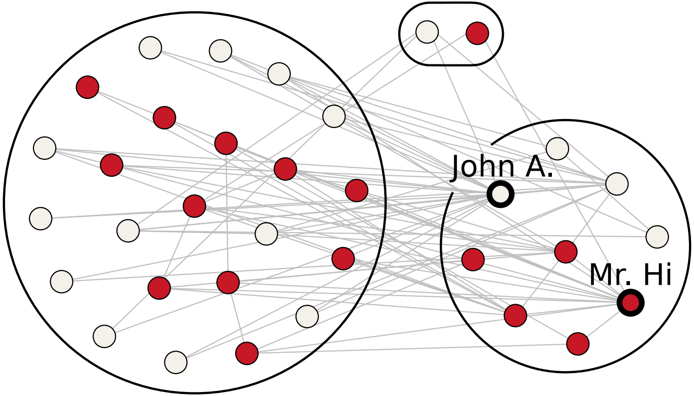
- few internal edges
- many external edges

  (*generalization of multipartite graphs*)



Zachary's Karate Club Network

$|V| = 34,$
$|E| = 156$



Anti-communities in *Zachary's karate club* network [Zac77].
Colors denote membership after the fission of the club.

Challenges and Objectives

- **Definition**

  How to define anti-communities?

- **Models and Algorithms**

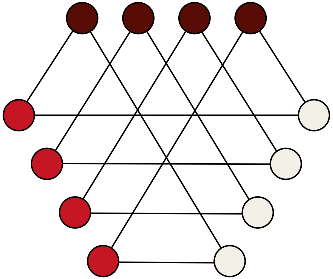
  Which algorithms can be used?

- **Exploratory Analysis**

  Are anti-communities also present in other networks?

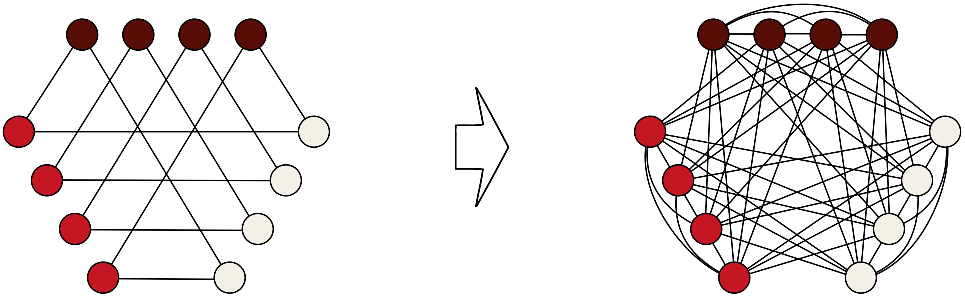
Definition

Graph Complement



Original network
with 3 anti-communities

Graph Complement



Original network
with 3 anti-communities

Graph complement
with 3 communities

Definition

Definition

Vertices $C \subseteq V$ of graph $G = (V, E)$ form an anti-community iff C forms a community in the graph complement $\hat{G} = (V, \hat{E})$ with $\hat{E} := (V \times V) \setminus E$.

Definition

Definition

Vertices $C \subseteq V$ of graph $G = (V, E)$ form an anti-community iff C forms a community in the graph complement $\hat{G} = (V, \hat{E})$ with $\hat{E} := (V \times V) \setminus E$.

Conclusions:

- Not really unique (many definitions for communities)
- Many existing algorithms and methods can be reused

Models and Algorithms

Proposed Methods

Existing methods either slow or poor quality.

Greedy algorithms

- using *Modularity* measure [NG04]
- using *Anti-Modularity* measure [CYC14]

Vertex similarity

- Adjacency mapping
- Distance mapping

Proposed Methods

Existing methods either slow or poor quality.

Greedy algorithms

- using *Modularity* measure [NG04]
- using *Anti-Modularity* measure [CYC14]

} Optimization problem

Vertex similarity

- Adjacency mapping
- Distance mapping

} Clustering problem

Modularity Measure

Intuition: Number of internal edges in $\mathcal{G} = (V, E)$ minus number of edges in a random graph with same degree-distribution.

Modularity of a graph

$$M := \frac{1}{2m} \sum_{ij} \left[a_{ij} - \frac{d_i d_j}{2m} \right] \delta(g_i, g_j)$$

m:	Total number of edges
$\mathbf{A} = [a_{ij}]$:	Adjacency matrix of $\mathcal{G}$
$d = [d_i]$:	Vertex degrees
$\delta(g_i, g_j)$:	1 iff v_i and v_j are both in same group

Greedy Algorithms

Make locally optimal choice at each step.

1. **Initialization**

   Assign each vertex to a separate group

Greedy Algorithms

Make locally optimal choice at each step.

1. **Initialization**

   Assign each vertex to a separate group

2. **Merge**

   Merge two groups, s.t. the *Modularity* is minimized
   (or the *Anti-Modularity* is maximized)

Greedy Algorithms

Make locally optimal choice at each step.

1. **Initialization**

   Assign each vertex to a separate group

2. **Merge**

   Merge two groups, s.t. the *Modularity* is minimized
   (or the *Anti-Modularity* is maximized)

3. **Repeat**

   If more than one group is left, go to step 2.
   Otherwise, return groups with best *(Anti-)Modularity*.

Based on the concept of structural equivalence.

1. **Mapping**

   Map vertices to feature vector representation

   - **Adjacency mapping:** $M(v_i) := [a_{ij}]_j$
   - **Distance mapping:** $M(v_i) := [d(v_i, v_1), \ldots, d(v_i, v_n)]$

Vertex Similarity

Based on the concept of structural equivalence.

1. **Mapping**

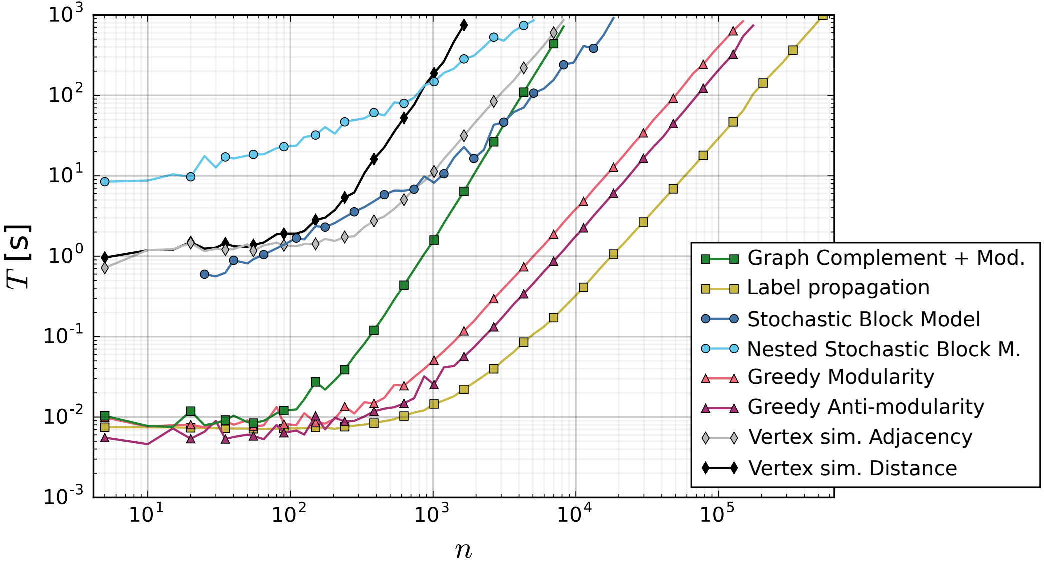
   Map vertices to feature vector representation

   - **Adjacency mapping:** $M(v_i) := [a_{ij}]_j$
   - **Distance mapping:** $M(v_i) := [d(v_i, v_1), \ldots, d(v_i, v_n)]$

2. **Clustering**

   Compute clustering of feature vectors (*k-Means*, . . .)

Runtime Evaluation

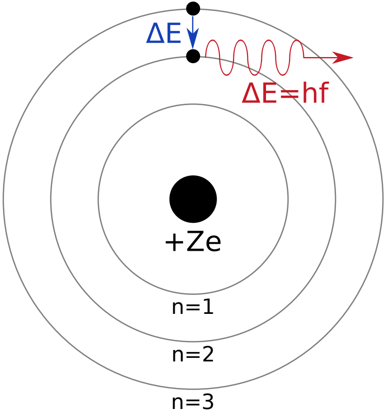


Evaluation with Erdős-Rényi random graphs (sparse)

Exploratory Analysis

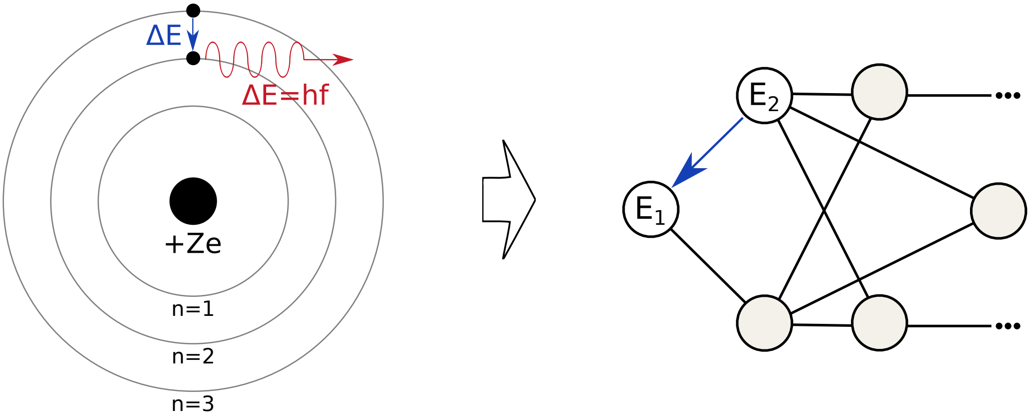
Spectral Line Networks

Goal: Encode energy states of a physical system (and their relation) in a network.



Spectral Line Networks

Goal: Encode energy states of a physical system (and their relation) in a network.



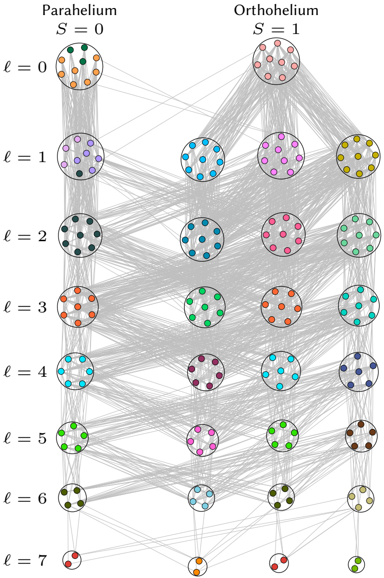
Example: *Spectral Line Network of Helium*

Spectral line network network of *Helium* [KRRN15] with $|V| = 183$, $|E| = 2282$.

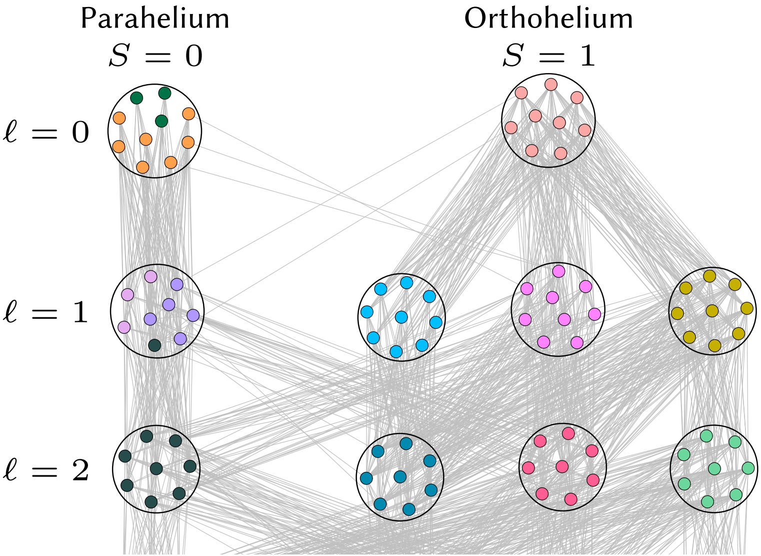
Colors show the anti-communities obtained with a vertex similarity method.

Circles show the ground-truth partition

- orbital angular momentum (ℓ),
- total angular momentum (j), and
- spin (s)

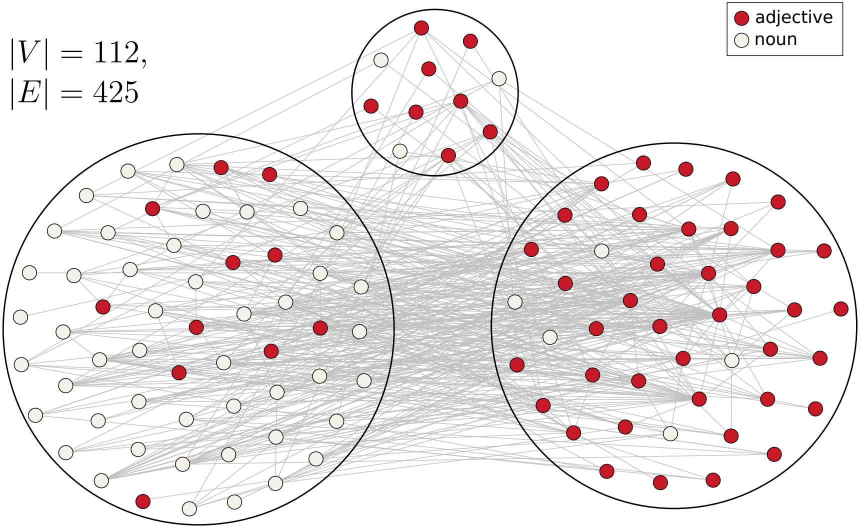


Example: *Spectral Line Network of Helium*



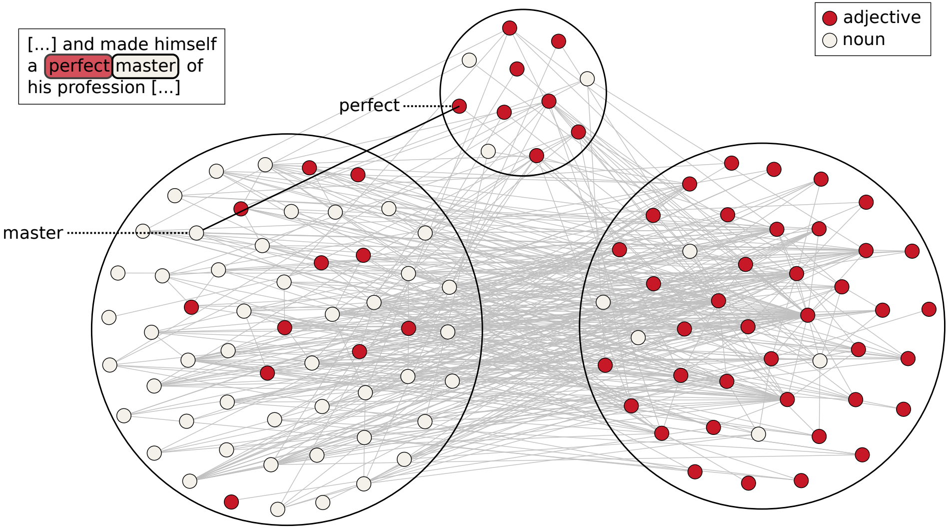
Example: *Adjectives and Nouns* Network

$|V| = 112,$
$|E| = 425$



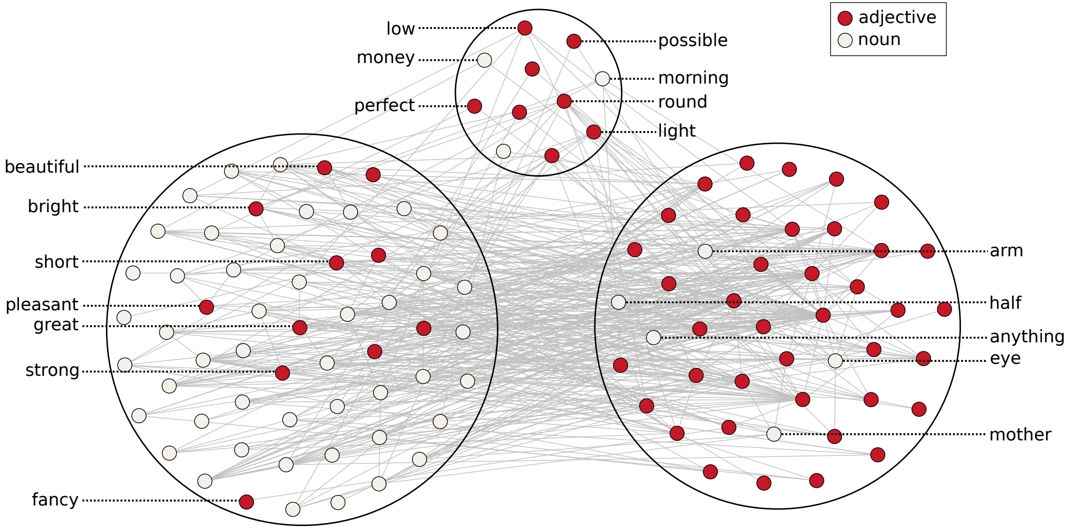
Adjectives and Nouns network [New06]. Circles correspond to the anti-communities found by the greedy modularity minimization algorithm.

Example: *Adjectives and Nouns* Network



Adjectives and Nouns network [New06]. Circles correspond to the anti-communities found by the greedy modularity minimization algorithm.

Example: *Adjectives and Nouns* Network



Adjectives and Nouns network [New06]. Circles correspond to the anti-communities found by the greedy modularity minimization algorithm.

Summary

Summary

- Anti-community structures are present in many networks, including
  - networks of spectral line transitions
  - Zachary's karate club network
  - . . . and many more

- Many concepts of traditional community detection can be reused by computing the graph complement

- Specialized algorithms and measures are required if performance is important

Further Reading

- **Evaluation measures:**

  Adaption of the adjusted Rand index and normalized mutual information measures for anti-communities.

- **Random graphs:**

  Algorithms to generate Erdős-Rényi and Barabási-Albert random graph model for graphs with (anti-)community structure.

- **Performance evaluation:**

  Quality comparison for graphs with known community structure.

Resources

Implementations and datasets available at:

`http://dbs.ifi.uni-heidelberg.de/resources/anticommunity`

Thank you!

Bibliography

Bibliography i

[CYC14] L. Chen, Q. Yu, and B. Chen. "Anti-modularity and anti-community detecting in complex networks". In: *Inf. Sci.* 275 (2014), pp. 293–313.

[For10] S. Fortunato. "Community detection in graphs". In: *Phys. Rep.* 486.3 (2010), pp. 75–174.

[Hol04] J. M. Hollas. *Modern spectroscopy.* John Wiley & Sons, 2004.

[KRRN15] A. Kramida, Y. Ralchenko, J. Reader, and NIST ASD Team. NIST Atomic Spectra Database (ver. 5.3), [Online]. Available: `http://physics.nist.gov/asd` [2017, July 4]. National Institute of Standards and Technology, Gaithersburg, MD. 2015.

Bibliography ii

[New06] M. E. J. Newman. "Finding community structure in networks using the eigenvectors of matrices". In: *Phys. Rev. E* 74.3 (2006).

[NG04] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Phys. Rev. E* 69.2 (2004).

[Pei14] T. P. Peixoto. "Hierarchical block structures and high-resolution model selection in large networks". In: *Phys. Rev. X* 4 (1 2014).

[Pei17] T. P. Peixoto. "Bayesian stochastic blockmodeling". In: (2017). URL: `https://arxiv.org/abs/1705.10225`.

[Zac77] W. W. Zachary. "An information flow model for conflict and fission in small groups". In: *J. Anthropol. Res.* 33.4 (1977), pp. 452–473.

Backup Slides

Baseline Methods

- **Graph complement + X**

  Allows to reuse existing methods, but high memory usage / slow.

- **Label propagation algorithm for anti-communities** [CYC14]

  Fast, but poor quality

- **Generic methods**

  e.g., Stochastic block models [Pei14; Pei17]

Complexity of Greedy Algorithms

- **Community detection:**

Naive method	$\mathcal{O}(n^3)$
Skip unconnected edges	$\mathcal{O}(n(n+m))$
Use max-heap data structure	$\mathcal{O}(n \log^2 n)$[1]

[1] for graphs with strong hierarchical structure

Complexity of Greedy Algorithms

- **Community detection:**

Naive method	$\mathcal{O}(n^3)$
Skip unconnected edges	$\mathcal{O}(n(n+m))$
Use max-heap data structure	$\mathcal{O}(n \log^2 n)$[1]

- **Anti-community detection:**

Graph complement	$\mathcal{O}(n^3)$
Our method	$\mathcal{O}(n(n+m))$

[1] for graphs with strong hierarchical structure

Complexity of Greedy Algorithms

- **Community detection:**

Naive method	$\mathcal{O}(n^3)$
Skip unconnected edges	$\mathcal{O}(n(n+m))$
Use max-heap data structure	$\mathcal{O}(n \log^2 n)$[1]

- **Anti-community detection:**

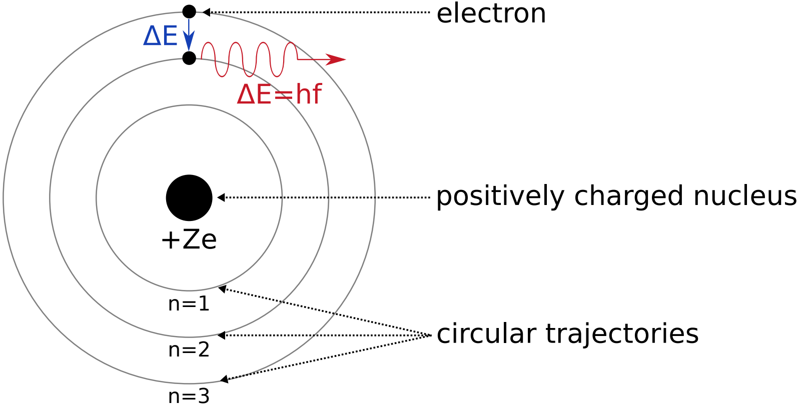
Graph complement	$\mathcal{O}(n^3)$
Our method	$\mathcal{O}(n(n+m))$

Result can also be used to improve community detection!

[1] for graphs with strong hierarchical structure

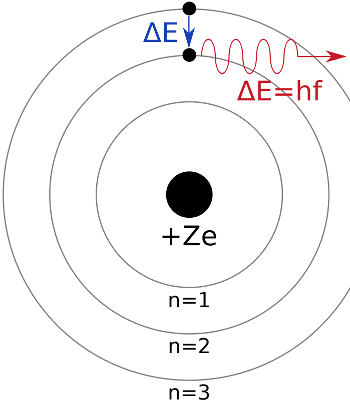
Basics of the *Bohr Model*

Goal: Encode energy states of a physical system (and their relation) in a network.



Basics of the *Bohr Model*

Goal: Encode energy states of a physical system (and their relation) in a network.



- **Energy states**

  defined by possible orbits of electrons

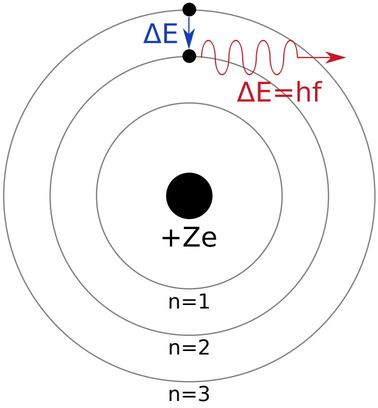
- **State transitions**

  requires / releases energy ΔE

  $\rightarrow$ emission or absorption line

Basics of the *Bohr Model*

Goal: Encode energy states of a physical system (and their relation) in a network.



- **Energy states**

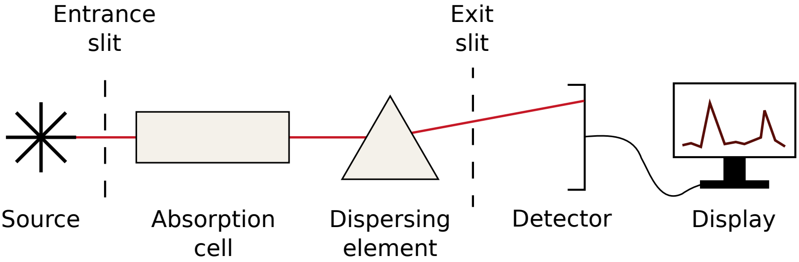
  defined by possible orbits of electrons

- **State transitions**

  requires / releases energy ΔE

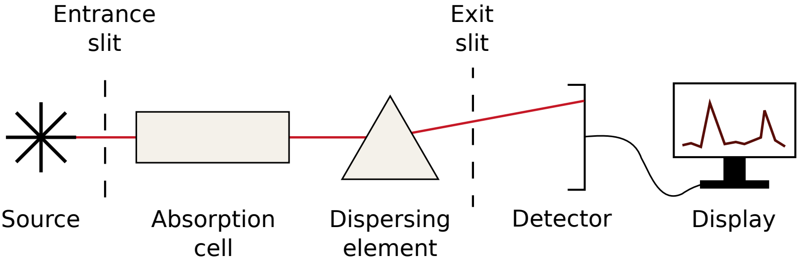
  $\rightarrow$ emission or absorption line

⚠ Simplified model!



Overview of an absorption experiment. Visualization based on *Modern Spectroscopy* by Hollas [Hol04].

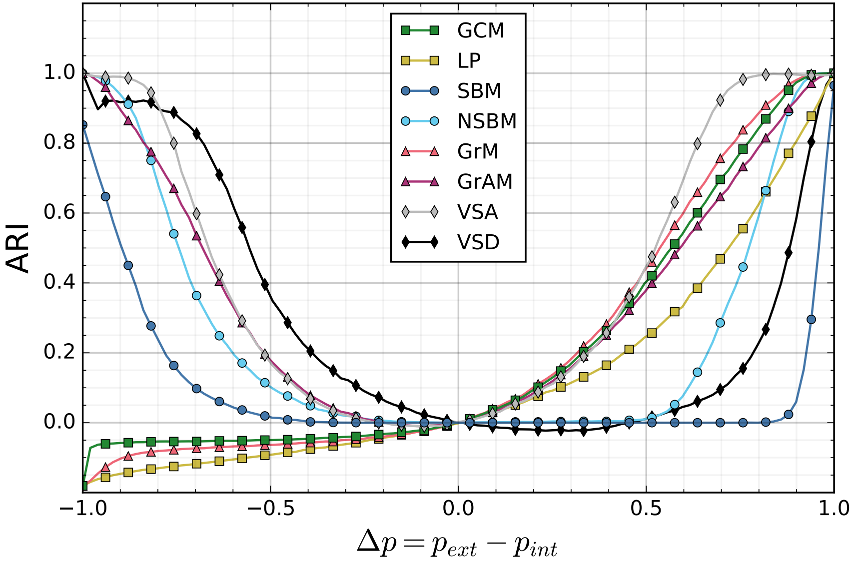
Spectral Line Networks



Overview of an absorption experiment. Visualization based on *Modern Spectroscopy* by Hollas [Hol04].

Spectral lines ▶ **State transitions** ▶ **Energy states** ▶ **Network**

Performance evaluation



Evaluation with Erdős-Rényi random graphs ($k = 5$)