# Graphs in Machine Learning

Michal Valko

*DeepMind Paris and Inria Lille*

TA: Omar Darwiche Domingues with the help of Pierre Perrault

Partially based on material by: Andreas Krause,
Branislav Kveton, Michael Kearns

# Piazza for Q&A's

**Purpose**

- registration for the class
- register with your **school** email and **full name**
- online course discussions and announcements
- questions and answers about the material and logistics
- **students encouraged to answer each others' questions**
- homework assignments
- virtual machine link and instructions
- **draft of the slides before the class**

`https://piazza.com/ens_cachan/fall2019/mvagraphsml` **NO EMAILS!**

class code given during the class

# Scribes 2019/2020

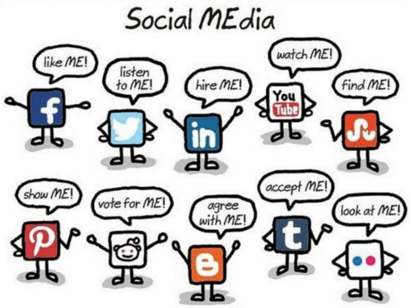
### Details

- number of people that can volunteer is limited (to 2x number of lectures)
- first-come-first serve by writing your name below (also in the resource section on piazza)
- use `https://www.overleaf.com/articles/tml-scribe-l08/wqhhzhgdprbdasthetemplate`
- grade bonus: -0.5-2 points depending on the quality
- important: you work in pairs to proofread what the other is typing
- example:
  `http://imagine.enpc.fr/~obozinsg/teaching/mva_gm/fall2017/`
- deadline: 1 month after the lecture

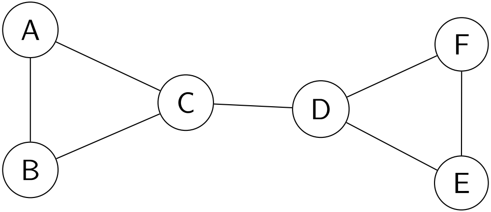
`https://piazza.com/ens_cachan/fall2019/mvagraphsml` **live now**

# Graphs from social networks

- people and their interactions
- directed (Twitter) and undirected (Facebook)
- structure is rather a *phenomena*
- typical ML tasks
  - advertising
  - product placement
  - link prediction (PYMK)

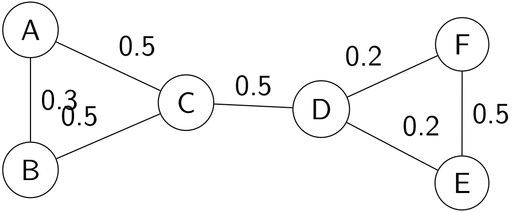


# Success story #1 Product placement - problem




Maximizing the Spread of Influence through a Social Network
http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf

# Success story #1 Product placement - problem

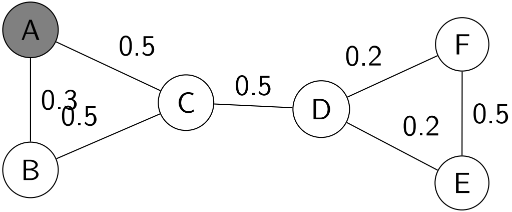



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

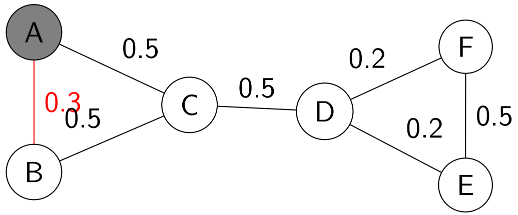



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf

# Success story #1 Product placement - problem

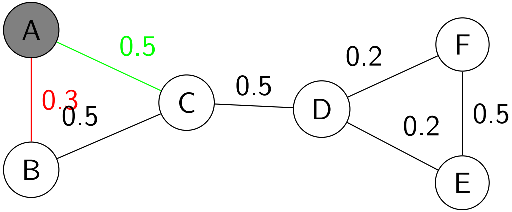



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

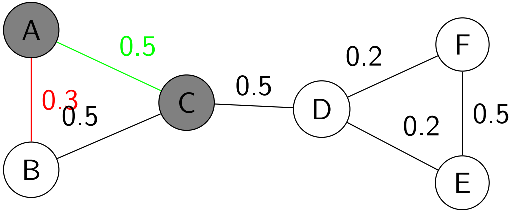



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

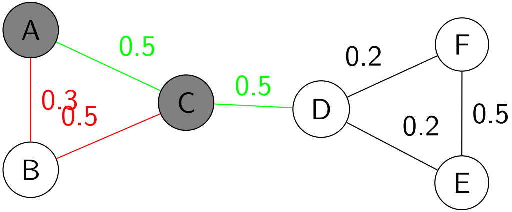



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

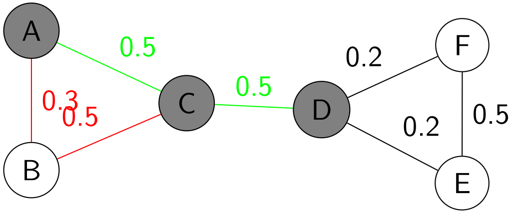



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

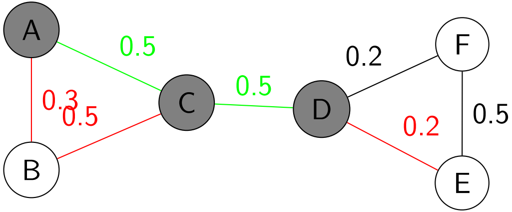



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem

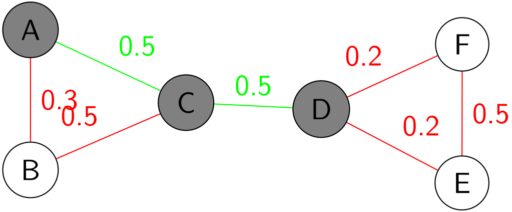



**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

Maximizing the Spread of Influence through a Social Network
`http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf`

# Success story #1 Product placement - problem




**Who should get free cell phones?**

$V = \{$**A**lice,**B**ob,**C**harlie,**D**orothy,**E**ric,**F**iona$\}$

$F(S) =$ Expected number of people influenced when targeting $S \subseteq V$ under some propagation model - e.g., cascades

How would you choose the target customers?

highest degree, close to the center, . . .

Maximizing the Spread of Influence through a Social Network
http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf

# Submodularity: Definition

A **set function** on a discrete set $A$ is **submodular** if for any $S \subseteq T \subseteq A$ and for any $e \in A \setminus T$

$$f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$$

Example: $S = \{\text{stuff}\} = \{\text{bread, apple, tomato, ...}\}$

$f(V) =$ cost of getting products $V$

$f(\{\text{bread}\}) = c(\text{bakery}) + c(\text{bread})$

$f(\{\text{bread}, \text{apple}\}) = c(\text{bakery}) + c(\text{bread}) + c(\text{market}) + c(\text{apple}\})$

$f(\{\text{bread}, \text{tomato}\}) = c(\text{bakery}) + c(\text{bread}) + c(\text{market}) + c(\text{tomato})$

$f(\{\text{bread}, \text{tomato}, \text{apple}\}) = c(\text{bakery}) + c(\text{bread}) + c(\text{market}) + c(\text{tomato}) + c(\text{apple})$

Adding an apple to the smaller set costs more!

$$\{\text{bread}\} \subseteq \{\text{bread}, \text{tomato}\}$$

$$f(\{\text{bread}, \text{apple}\}) - f(\{\text{bread}\}) > f(\{\text{bread}, \text{tomato}, \text{apple}\}) - f(\{\text{tomato}, \text{bread}\})$$

Diminishing returns: Buying in bulk is cheaper!

# Submodularity: Application

*Objective:* Find $\arg\max_{S \subseteq A, |S| \leq k} f(S)$

*Property:* NP-hard in general

*Special case:* $f$ is also **nonnegative** and **monotone**.

**Other examples:** information, graph cuts, covering, ...

Link to our **product placement** problem on a **social network graph?**

submodular?, nonnegative?, monotone?, $k$?

http://thibaut.horel.org/submodularity/papers/nemhauser1978.pdf

Let $S^\star = \arg\max_{S \subseteq A, |S| \leq k} f(S)$ where $f$ is monotonic and submodular set function and let $S_{\texttt{Greedy}}$ be a **greedy solution**.

$$\text{Then} \quad f(S_{\texttt{Greedy}}) \geq \left(1 - \frac{1}{e}\right) \cdot f(S^\star).$$

# Submodularity: `Greedy` algorithm

**Input:**
  $k$: the maximum allowed cardinality of the output
  $V$: a ground set
  $f$: a monotone, non-negative, and submodular function
**Run:**
$S_0 = \emptyset$
**for** $i = 1$ **to** $k$ **do**
  $S_i \leftarrow S_{i-1} \cup \left\{ \arg\max_{a \in V \setminus S_{i-1}} \left[ f\left(\{a\} \cup S_{i-1}\right) - f\left(S_{i-1}\right)\right] \right\}$
**end for**
**Output:**
  Return $S_{\texttt{Greedy}} = S_k$

Let $S^\star = \arg\max_{S \subseteq A, |S| \leq k} f(S)$ where $f$ is monotonic and submodular set function and let $S_{\texttt{Greedy}}$ be a **greedy solution**.

$$\text{Then} \quad f(S_{\texttt{Greedy}}) \geq \left(1 - \frac{1}{e}\right) \cdot f(S^\star).$$

# Submodularity: Approximation guarantee of `Greedy`

Let $S_i$ be the $i$-th set selected by `Greedy`, $S_{\texttt{Greedy}} = S_k$. We show

$$f(S^\star) - f(S_i) \le \left(1 - \tfrac{1}{k}\right)^i \cdot f(S^\star).$$

Difference from the optimum before the $i$-th step ...

$$\begin{aligned} f\left(S^\star\right) - f\left(S_{i-1}\right) &\le f\left(S^\star \cup S_{i-1}\right) - f\left(S_{i-1}\right) \\ &\le \sum_{a \in S^\star \setminus S_{i-1}} \left(f\left(\{a\} \cup S_{i-1}\right) - f\left(S_{i-1}\right)\right) \\ &\le \sum_{a \in S^\star \setminus S_{i-1}} \left(f\left(S_i\right) - f\left(S_{i-1}\right)\right) \\ &\le k\left(f\left(S_i\right) - f\left(S_{i-1}\right)\right) \end{aligned}$$

Difference from the optimum after the $i$-th step ...

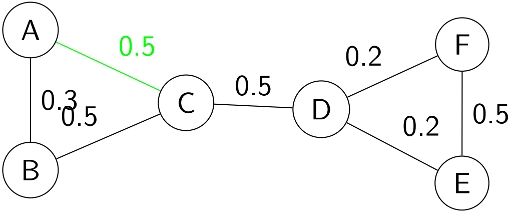
$$\begin{aligned} f(S^\star) - f(S_i) &= f(S^\star) - f(S_{i-1}) - \left(f\left(S_i\right) - f\left(S_{i-1}\right)\right) \\ &\le f(S^\star) - f(S_{i-1}) - \frac{f(S^\star) - f(S_{i-1})}{k} \end{aligned}$$

# Submodularity: Graph-related examples

- Influence maximization on networks (current example)
- Maximum-weight spanning trees
- Graph cuts
- Structure learning in graphical models (PGM course)
- More examples http://people.math.gatech.edu/~tetali/LINKS/IWATA/SFGT.pdf
- Deep Submodular Functions (2017) https://arxiv.org/pdf/1701.08939.pdf

back to the influence-maximization example ...

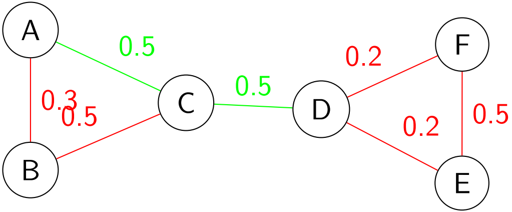
# Success story #1 Product placement - solution




**Key idea:** Flip coins $c$ in advance $\rightarrow$ "live" edges

MIIA: http://hanj.cs.illinois.edu/pdf/dmkd12_cwang.pdf/
Tutorial: cf. Andreas Krause http://submodularity.org/
Course: Jeff Billmes at UW

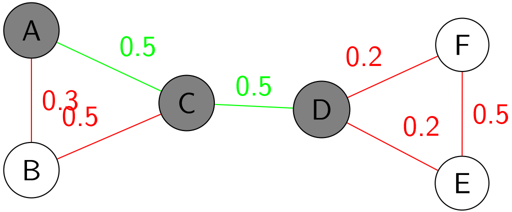
# Success story #1 Product placement - solution




**Key idea:** Flip coins $c$ in advance → “live” edges

MIIA: http://hanj.cs.illinois.edu/pdf/dmkd12_cwang.pdf/
Tutorial: cf. Andreas Krause http://submodularity.org/
Course: Jeff Billmes at UW

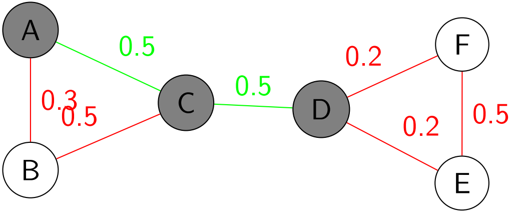
# Success story #1 Product placement - solution




**Key idea:** Flip coins $c$ in advance $\rightarrow$ "live" edges
$F_c(V)$ = People influenced under outcome $c$ (set cover!)

MIIA: http://hanj.cs.illinois.edu/pdf/dmkd12_cwang.pdf/
Tutorial: cf. Andreas Krause http://submodularity.org/
Course: Jeff Billmes at UW

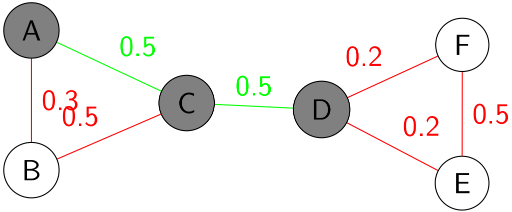
# Success story #1 Product placement - solution




**Key idea:** Flip coins $c$ in advance → "live" edges
$F_c(V)$ = People influenced under outcome $c$ (set cover!)
$F(V) = \sum_c P(c)F_c(V)$ is submodular as well!

MIIA: `http://hanj.cs.illinois.edu/pdf/dmkd12_cwang.pdf/`
Tutorial: cf. Andreas Krause `http://submodularity.org/`
Course: Jeff Billmes at UW

# Success story #1 Product placement - solution

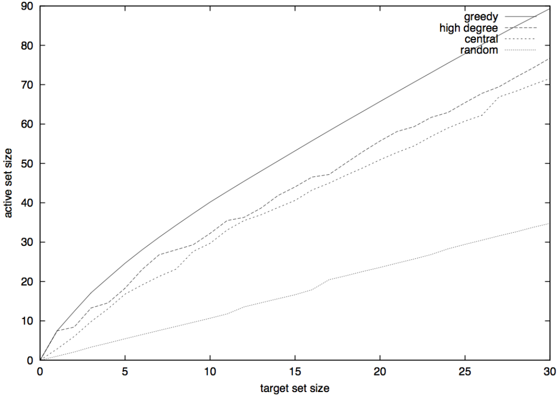



**Key idea:** Flip coins $c$ in advance $\rightarrow$ "live" edges
$F_c(V) =$ People influenced under outcome $c$ (set cover!)
$F(V) = \sum_c P(c)F_c(V)$ is submodular as well!
Computational issues?

MIIA: `http://hanj.cs.illinois.edu/pdf/dmkd12_cwang.pdf/`
Tutorial: cf. Andreas Krause `http://submodularity.org/`
Course: Jeff Billmes at UW

# Success story #1 Product placement - comparison

**influence** on the ArXiv/Physics co-authorship graph




greedy approximation does better than the centrality measures

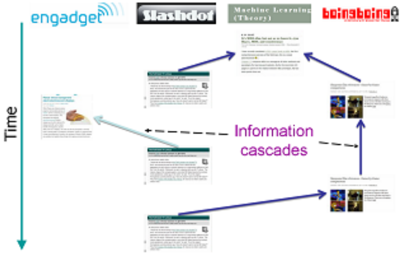
# Graphs from utility and technology networks

- link services
- power grids, roads, transportation networks, Internet, sensor networks, water distribution networks
- structure is either *hand designed* or not
- typical ML tasks
  - best routing under unknown or variable costs
  - identify the node of interest

Berkeley's Floating Sensor Network

# Graphs from information networks

- web
- blogs
- wikipedia
- typical ML tasks
  - find influential sources
  - search (`PageRank`)



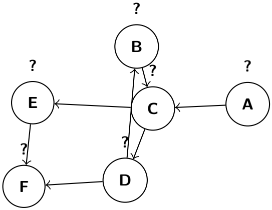

Blog cascades (ETH) - *submodularity*

# Success story #2 Google PageRank

*Objective:* **Rank** all web pages (nodes on the graph) by how **many** other pages link to them and how **important** they are.

basic PageRank is independent of query and the page content

Internet $\rightarrow$ graph $\rightarrow$ matrix $\rightarrow$ stochastic matrix $\mathbf{M}$ $\left(\sum_j \mathbf{M}_{ij} = 1\right)$



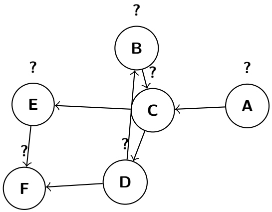
# Success story #2 Goog… …eRank

*Objective:* **Rank** … …ph) by how **many** other pages li… …e.

basic PageRan… …e content

Internet → graph → … …trix **M** $\left(\sum_j \mathbf{M}_{ij} = 1\right)$

**Random Surfer** Process



# Success story #2 Goog… …eRank

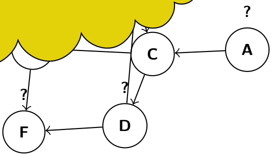
*Objective:* **Rank** … …ph) by how **many** other pages li… …e.

basic PageRan… …ge content

Internet → graph → … …trix **M** $\left(\sum_j \mathbf{M}_{ij} = 1\right)$

**Random Surfer** Process

What is wrong with it?



# Success story #2 Google `PageRank`

`http://infolab.stanford.edu/~backrub/google.html`:

*`PageRank` can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page.*

- page is **important** if **important** pages link **to** it
  - circular definition
- importance of a page is distributed **evenly**
- probability of being bored is 15%

# Success story #2 Google PageRank

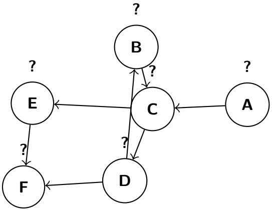
**Google matrix:** $\mathbf{G} = (1-p)\mathbf{M} + p \cdot \frac{1}{N}\mathbb{1}_{N \times N}$, where $p = 0.15$

# Success story #2 Google PageRank

**Google matrix:** $\mathbf{G} = (1-p)\mathbf{M} + p \cdot \frac{1}{N}\mathbb{1}_{N\times N}$, where $p = 0.15$

**G** is **stochastic** why? What is **Ga** for any **a**? We look for $\mathbf{G}\mathbf{v} = 1 \times \mathbf{v}$, steady-state vector, a right eigenvector with eigenvalue 1. why?

**Perron's theorem:** Such $v$ exists and it is **unique**
if the entries of **G** are positive.

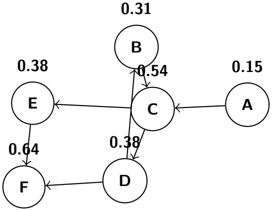


# Success story #2 Google PageRank

**Google matrix:** $\mathbf{G} = (1-p)\mathbf{M} + p \cdot \frac{1}{N}\mathbb{1}_{N\times N}$, where $p = 0.15$

**G** is **stochastic** why? What is **Ga** for any **a**? We look for $\mathbf{Gv} = 1 \times \mathbf{v}$,
steady-state vector, a right eigenvector with eigenvalue 1. why?

**Perron's theorem:** Such $v$ exists and it is **unique**
if the entries of **G** are positive.

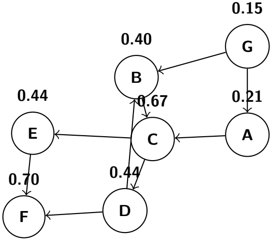


# Success story #2 Google PageRank

**Google matrix:** $\mathbf{G} = (1-p)\mathbf{M} + p \cdot \frac{1}{N}\mathbb{1}_{N\times N}$, where $p = 0.15$

$\mathbf{G}$ is **stochastic** why? What is $\mathbf{Ga}$ for any $\mathbf{a}$? We look for $\mathbf{Gv} = 1 \times \mathbf{v}$, steady-state vector, a right eigenvector with eigenvalue 1. why?

**Perron's theorem:** Such $v$ exists and it is **unique**
if the entries of $\mathbf{G}$ are positive.



# Success story #2 Google `PageRank`

**History:** [Desikan, 2006]

- The anatomy of a large-scale hypertextual web search engine [Brin & Page 1998]
- US patent for PageRank granted in 2001
- Google indexes 10's of billions of web pages (1 billion $= 10^9$)
- Google serves $\geq$ 200 million queries per day
- Each query processed by $\geq$ 1000 machines
- All search engines combined process more than 500 million queries per day

# Success story #2 Google PageRank

*Problem:* Find an eigenvector of a stochastic matrix.

- $n = 10^9$ !!!
- luckily: **sparse** (average outdegree: 7)
- better than a simple centrality measure (e.g., degree)
- power method

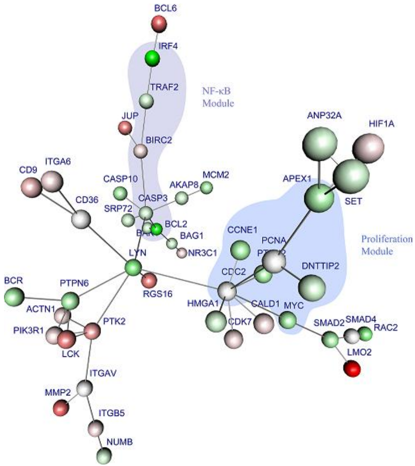
$$\mathbf{v}_0 = (1_A \quad 0_B \quad 0_C \quad 0_D \quad 0_E \quad 0_F)^\mathsf{T}$$
$$\mathbf{v}_1 = \mathbf{G}\mathbf{v}_0$$
$$\mathbf{v}_{t+1} = \mathbf{G}\mathbf{v}_t = \mathbf{G}^{t+1}\mathbf{v}$$

$\mathbf{v}_{t+1} = \mathbf{v}_t \implies \mathbf{G}\mathbf{v}_t = \mathbf{v}_t$ and we found the steady vector

But wait, **M** is sparse, but **G** is dense! What to do?

# Graphs from biological networks

- protein-protein interactions
- gene regulatory networks
- typical ML tasks
  - discover unexplored interactions
  - learn or reconstruct the structure



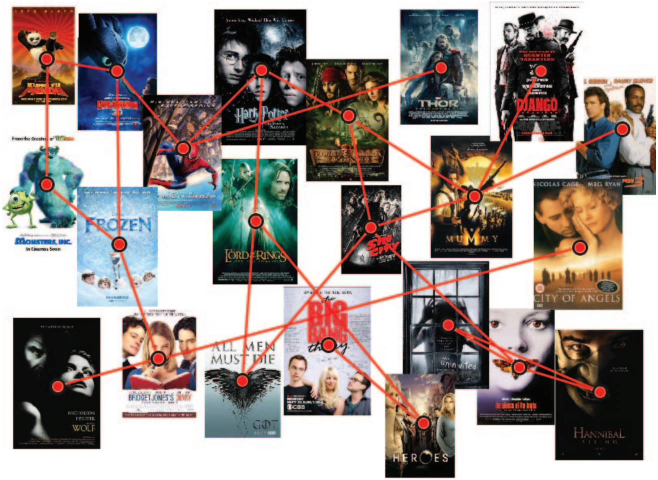

Diffuse large B-cell lymphomas - Dittrich et al. (2008)

# Graphs from similarity networks

**graph is not naturally given**

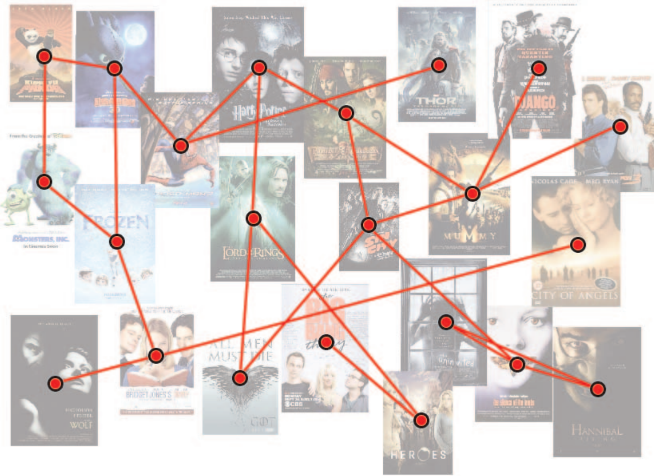
# Graphs from similarity networks

**but we can construct it**



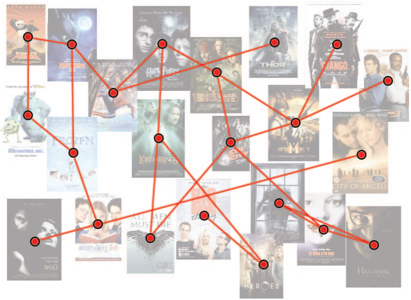
# Graphs from similarity networks

**and use it as an abstraction**



# Graphs from similarity networks

- vision
- audio
- text
- typical ML tasks
  - **semi-supervised learning**
  - **spectral clustering**
  - **manifold learning**



movie similarity

# Two sources of graphs in ML

**Graph as models for networks**

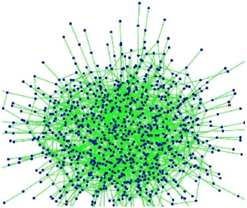
- given as an input
- discover interesting properties of the structure
- represent useful information (viral marketing)
- be the object of study (anomaly detection)

**Graph as nonparametric basis**

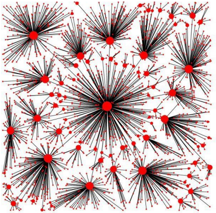
- we create (learn) the structure
- flat vectorial data $\rightarrow$ similarity graph
- nonparametric regularizer
- encode structural properties: smoothness, independence, ...

# Random Graph Models

**Erdős-Rényi**
independent edges



**Barabási-Albert**
preferential attachment



**Stochastic Blocks**
modeling communities

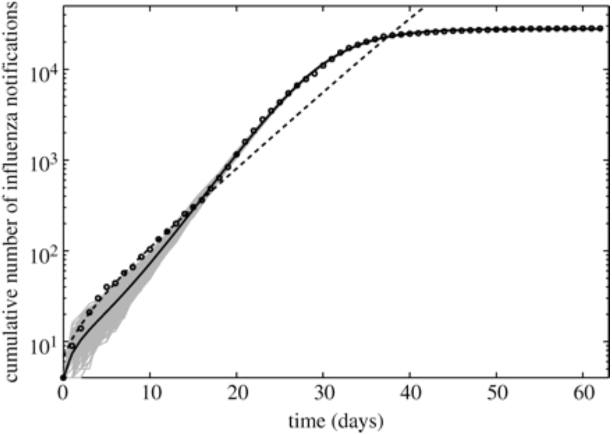
Watts-Strogatz, Chung-Lu, Fiedler, ....

# Erdős number project

- `http://www.oakland.edu/enp/` **try it!**
- an example of a real-world graph
- 401 000 authors, 676 000 edges ($\ll 401000^2 \rightarrow$ sparse)
- average degree 3.36
- average distance for the largest component: 7.64
- 6 degrees of separation [Travers & Milgram, 1967]
- heavy tail

# Spanish flu in San Francisco 1918–1919

Small-world phenomenon and diseases

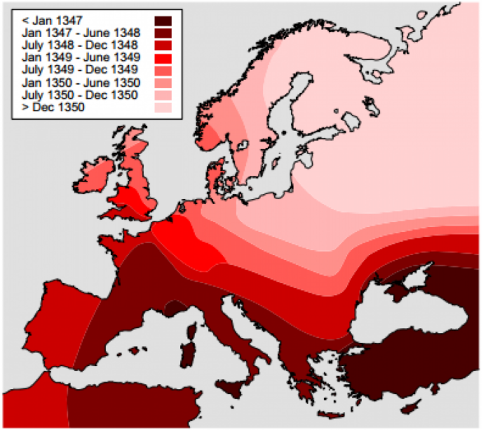



http://rsif.royalsocietypublishing.org/content/4/12/155

Small world: Obvious?

# Black death!

# Black death: spread



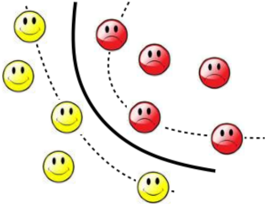

source: catholic.org

https://www.youtube.com/watch?v=EEK6c9Bh5CQ

# Links to the other courses

- **Introduction to statistical learning**
  - links to the learning theory on graphs: label propagation, learnability, generalization
- **Reinforcement learning**
  - link to the online learning (bandit) lecture at the end of the semester
- **Advanced learning for text and graph data**
  - data-mining graph course on the topics not covered in this course
  - details on the next slide

# Statistical Machine Learning in Paris!



https://sites.google.com/site/smileinparis/home

**Speakers:** ML researches and Ph.D. students - former MVA students

**Topic:** Francis Bach: Double descent

**Date:** Thursday October 17th

**Time:** 15:00 - 16:30

**Place:** Inria Paris https://forms.gle/H7VTjNfnLGFKQpYf7

# Parisian Deep and Sequential Seminar - New in 2019

**Organizers:** Brain, DeepMind, Criteo, FB, P6, Dauphine, P11

**Next session:** soon be to announced

**Link above:** soon be to active

# Administrivia

**Time:** Tuesday afternoons, next week at 13:30

**Place:** ENS Cachan, next week at Salle Condorcet

**7 lectures + 3 recitations (TDs)**

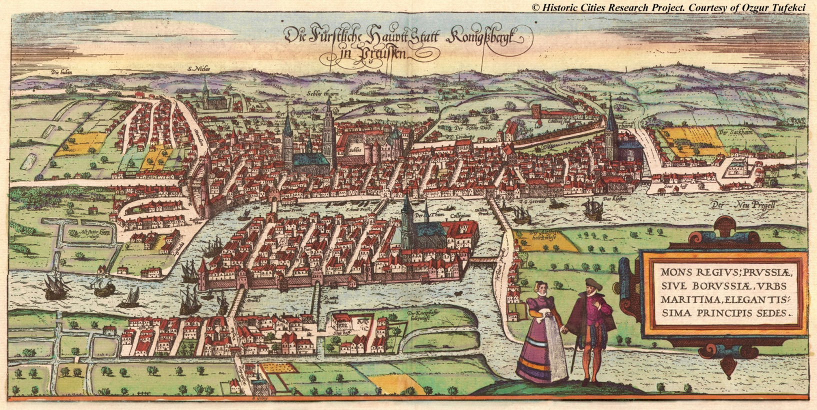
**Validation:** grades from TDs (40%) + class project (60%)

**Research:** contact me for *internships*, *Ph.D. theses*, *projects*, etc.

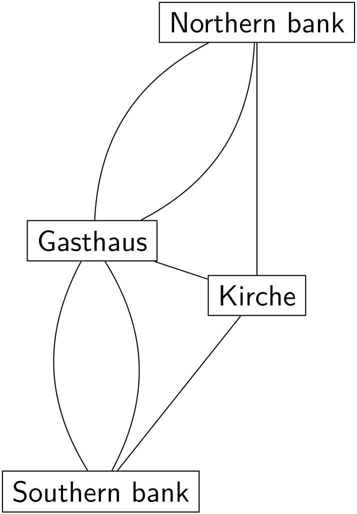
**Course website:**
`http://researchers.lille.inria.fr/~valko/hp/mva-ml-graphs`

**Contact, online class discussions, and announcements:**
`https://piazza.com/ens_cachan/fall2019/mvagraphsml`
class code given during the class

# Graph theory refresher



# Graph theory refresher



# Graph theory refresher

- 250 years of graph theory
- Seven Bridges of Königsberg (Leonhard Euler, 1735)
- necessary for Eulerian circuit: 0 or 2 nodes of odd degree
- after bombing and rebuilding there are now 5 bridges in Kaliningrad for the nodes with degrees $[2, 2, 3, 3]$
- the original problem is solved but not practical
  `http://people.engr.ncsu.edu/mfms/SevenBridges/`

# Similarity Graphs

Input: $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N$

- raw data
- flat data
- vectorial data

# Similarity Graphs

Similarity graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ — **(un)weighted**

*Task 1:* For each pair $i$, $j$: define a **similarity function** $s_{ij}$

*Task 2:* Decide which edges to include

$\varepsilon$-neighborhood graphs – connect the points with the distances smaller than $\varepsilon$

$k$-NN neighborhood graphs – take $k$ nearest neighbors

fully connected graphs - consider everything

*This is art (not much theory exists).*

`http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/publications/Luxburg07_tutorial.pdf`

# Similarity Graphs: $\varepsilon$-neighborhood graphs

Edges connect the points with the distances smaller than $\varepsilon$.

- distances are roughly on the same scale ($\varepsilon$)
- weights may not bring additional info $\rightarrow$ unweighted
- equivalent to: similarity function is at least $\varepsilon$
- theory [Penrose, 1999]: $\varepsilon = ((\log N)/N)^{1/d}$ to guarantee connectivity $N$ nodes, $d$ dimension, https://projecteuclid.org/euclid.aop/1022677261
- practice: choose $\varepsilon$ as the length of the longest edge in the MST - minimum spanning tree

What could be the problem with this MST approach?

# Similarity Graphs: $k$-nearest neighbors graphs

Edges connect each node to its $k$-nearest neighbors.

- asymmetric (or directed graph)
  - option OR: ignore the direction
  - option AND: include if we have both direction (mutual $k$-NN)
- how to choose $k$?
- $k \approx \log N$ - suggested by asymptotics (practice: up to $\sqrt{N}$)
- for mutual $k$-NN we need to take larger $k$
- mutual $k$-NN does not connect regions with different density
- why don't we take $k = N - 1$?

# Similarity Graphs: Fully connected graphs

Edges connect everything.

- choose a "meaningful" similarity function $s$
- default choice:

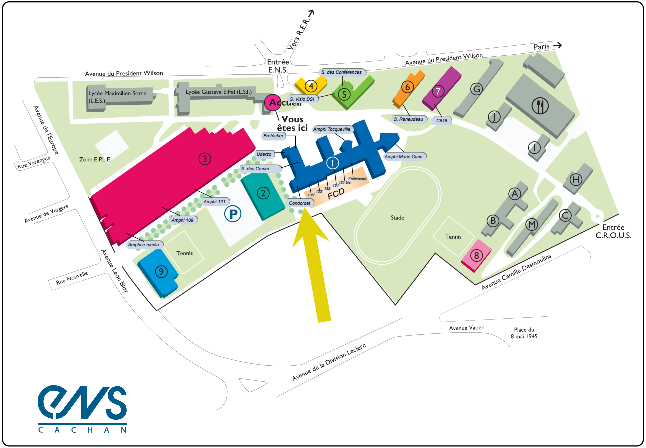
$$s_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- why the exponential decay with the distance?
- $\sigma$ controls the width of the neighborhoods
  - similar role as $\varepsilon$
  - **a** practical rule of thumb: 10% of the average empirical std
  - possibility: learn $\sigma_i$ for each feature independently
- metric learning (a whole field of ML)

# Similarity Graphs: Important considerations

- *calculate all $s_{ij}$ and threshold* has its limits ($N \approx 10000$)
- graph construction step can be a huge bottleneck
- want to go higher? (we often have to)
  - down-sample
  - approximate NN
    - **LSH** - Locally Sensitive Hashing
    - **CoverTrees**
    - **Spectral sparsifiers**
  - sometime we may not need the graph (just the final results)
  - yet another story: when we start with a large graph and want to make it sparse (later in the course)
- these rules have little theoretical underpinning
- similarity is very data-dependent

# Next class on Tuesday, October 8th at 13:30!



*Michal Valko*
contact via Piazza