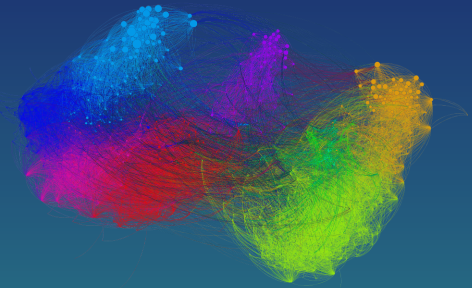
Graphs in Machine Learning

Michal Valko

Inria Lille - Nord Europe, France

Partially based on material by: Tomáš Kocák



Last Lecture

- Examples of applications of online SSL
- Analysis of online SSL
- SSL Learnability
- When does graph-based SSL provably help?
- Scaling harmonic functions to millions of samples

Previous Lab Session

- 16. 11. 2015 by Daniele Calandriello
- Content
  - Semi-supervised learning
  - Graph quantization
  - Online face recognizer
- Short written report
- Questions to piazza
- *Deadline:* 30. 11. 2015
- http://researchers.lille.inria.fr/~calandri/teaching.html

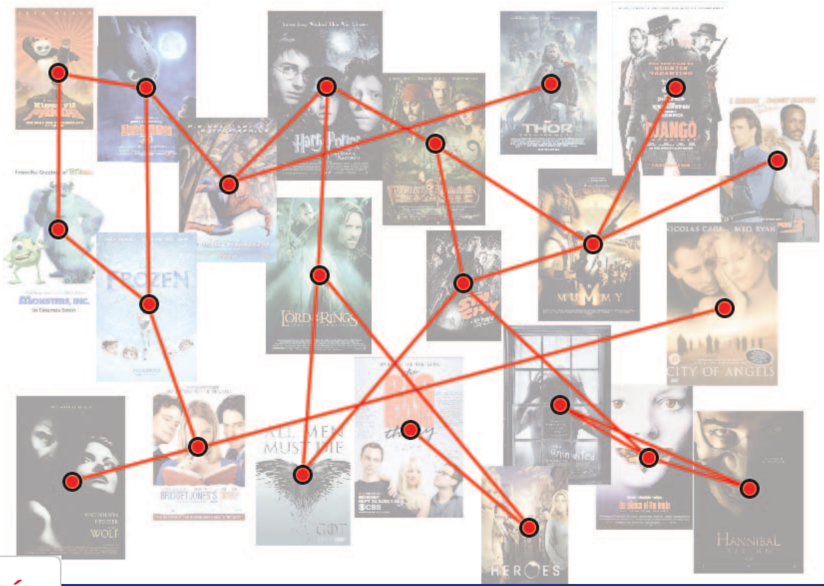
This Lecture

- Online decision-making on graphs
- Graph bandits
- Smoothness of rewards (preferences) on a given graph
- Observability graphs
- Exploiting side information

Final Class projects

- detailed description on the class website
- preferred option: you come up with the topic
- theory/implementation/review or a combination
- one or two people per project (exceptionally three)
- grade 60%: report + short presentation of the **team**
- deadlines
  - 23. 11. 2015 - strongly recommended DL for taking projects
  - 30. 11. 2015 - hard DL for taking projects
  - 06. 01. 2015 - submission of the project report
  - 11. 01. 2016 (or later) - project presentation
- list of suggested topics on piazza

Online Decision Making on Graphs



Online Decision Making on Graphs: Smoothness

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- *First application:* Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes

Online Decision Making on Graphs

Movie recommendation: (in each time step)

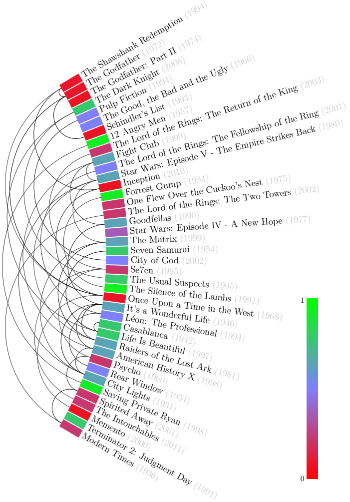
- Recommend movies to a **single user**.
- Good prediction after a few steps ($T \ll N$).

Goal:

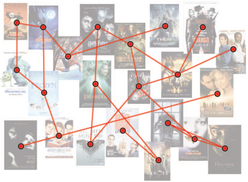
- Maximize overall reward (sum of ratings).

Assumptions:

- Unknown reward function $f : V(G) \to \mathbb{R}$.
- Function f is **smooth** on a graph.
- Neighboring movies $\Rightarrow$ similar preferences.
- Similar preferences $\not\Rightarrow$ neighboring movies.



Let's be lazy: Ignore the structure!



→

This is an multi-armed bandit problem!

The performance depends on the number of movies (N arms).

Worst case regret (to the best fixed strategy) $R_T = \mathcal{O}\left(\sqrt{NT}\right)$

What is N for imdb.com? 3,538,545 `http://www.imdb.com/stats`

Let's be lazy: Ignore the structure!

Another problem of the typical bandits strategies for recommendation?

If there is no information shared, we need to try all of the options!

UCB/MOSS and likely TS start with pulling each of the arms once

This is a problem both algorithmically and theoretically ....

Watch all the movies and then I tell you which one you like ....

What do we need for movie recommendation?

An algorithm useful in the case $T \ll N$!

Exploiting the structure is a must!

Recap: Smooth graph functions

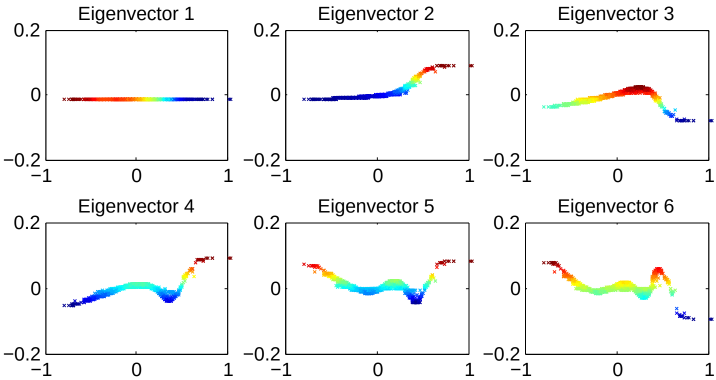
- $\mathbf{f} = (f_1, \ldots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathbf{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathbf{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}^*$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha}^* = \mathbf{f}$ Note: $\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^*$

$$S_G(\mathbf{f}) = \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^{*\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\alpha}^* = \|\boldsymbol{\alpha}^*\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N} \lambda_i (\alpha_i^*)^2$$

Smoothness and regularization: Small value of

(a) $S_G(\mathbf{f})$ **(b)** $\boldsymbol{\Lambda}$ norm of $\boldsymbol{\alpha}^*$ **(c)** α_i^* for large λ_i

Smooth graph functions: Flixster eigenvectors



Eigenvectors from the Flixster data corresponding to the smallest few eigenvalues of the graph Laplacian projected onto the first principal component of data. Colors indicate the values.

Online Learning Setting - Bandit Problem

Learning setting for a bandit algorithm π

- In each time t step choose a node $\pi(t)$.
- the $\pi(t)$-th row $\mathbf{x}_{\pi(t)}$ of the matrix $\mathbf{Q}$ corresponds to the arm $\pi(t)$.
- Obtain noisy reward $r_t = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$. Note: $\mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* = f_{\pi(t)}$
  - ε_t is R-sub-Gaussian noise. $\forall\xi \in \mathbb{R}, \mathbb{E}[e^{\xi\varepsilon_t}] \leq \exp\left(\xi^2 R^2/2\right)$
- Minimize cumulative regret

$$R_T = T\max_a\left(\mathbf{x}_a^{\mathsf{T}}\boldsymbol{\alpha}^*\right) - \sum_{t=1}^{T}\mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^*.$$

What is a good result?

Can't we just use *linear bandits*?

Online Decision Making on Graphs: Smoothness

- **Linear bandit algorithms**
  - **LinUCB** (Li et al., 2010)
    - Regret bound $\approx D\sqrt{T \ln T}$
  - **LinearTS** (Agrawal and Goyal, 2013)
    - Regret bound $\approx D\sqrt{T \ln N}$

Note: D is ambient dimension, in our case N, length of x_i.
Number of actions, e.g., all possible movies $\rightarrow$ **HUGE!**

- **Spectral bandit algorithms**
  - **SpectralUCB** (Valko et al., ICML 2014)
    - Regret bound $\approx d\sqrt{T \ln T}$
    - Operations per step: $D^2 N$
  - **SpectralTS** (Kocák et al., AAAI 2014)
    - Regret bound $\approx d\sqrt{T \ln N}$
    - Operations per step: $D^2 + DN$

Note: d is effective dimension, usually much smaller than D.

Effective dimension

- **Effective dimension:** Largest d such that

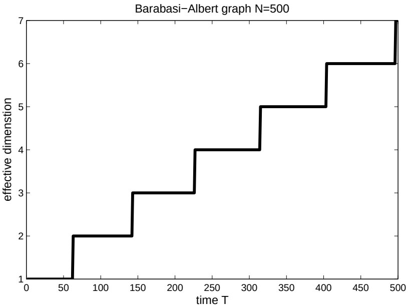
$$(d-1)\lambda_d \leq \frac{T}{\log(1+T/\lambda)}.$$

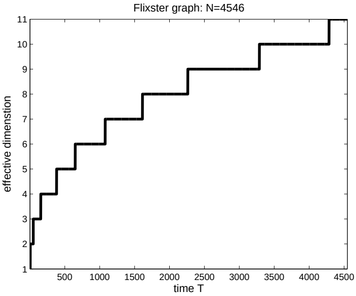
- Function of time horizon and graph properties
- λ_i : i-th smallest eigenvalue of $\boldsymbol{\Lambda}$.
- λ: Regularization parameter of the algorithm.

Properties:

- d is small when the coefficients λ_i grow rapidly above time.
- d is related to the number of "non-negligible" dimensions.
- Usually d is much smaller than D in real world graphs.
- Can be computed beforehand.

Effective dimension vs. Ambient dimension

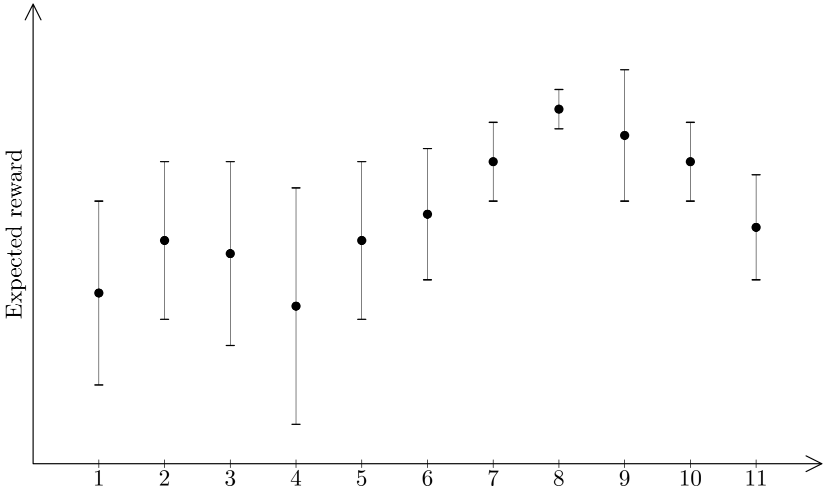




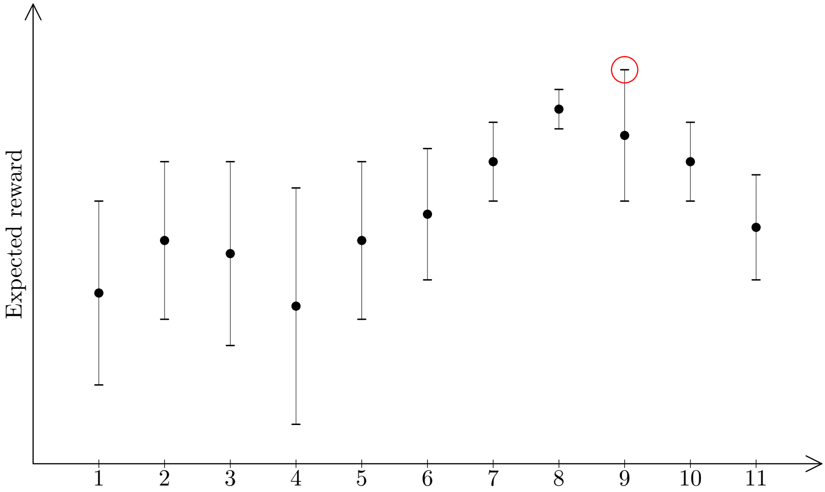
$$d \ll D$$

Note: In our setting $T < N = D$.

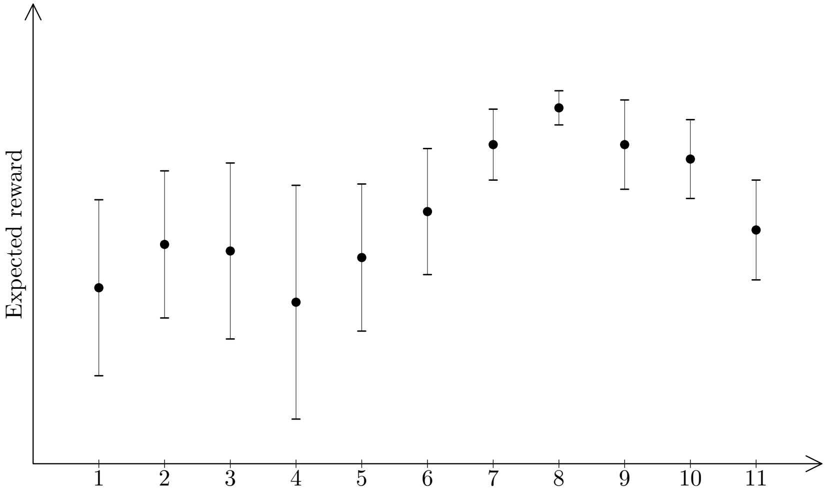
UCB-style algorithms: Estimate



UCB-style algorithms: Sample



UCB-style algorithms: Estimate ...



SpectralUCB

Given a vector of weights $\boldsymbol{\alpha}$, we define its $\boldsymbol{\Lambda}$ norm as

$$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} = \sqrt{\sum_{k=1}^{N} \lambda_k \alpha_k^2} = \sqrt{\boldsymbol{\alpha}^\mathsf{T} \boldsymbol{\Lambda} \boldsymbol{\alpha}},$$

and fit the ratings r_v with a (regularized) least-squares estimate

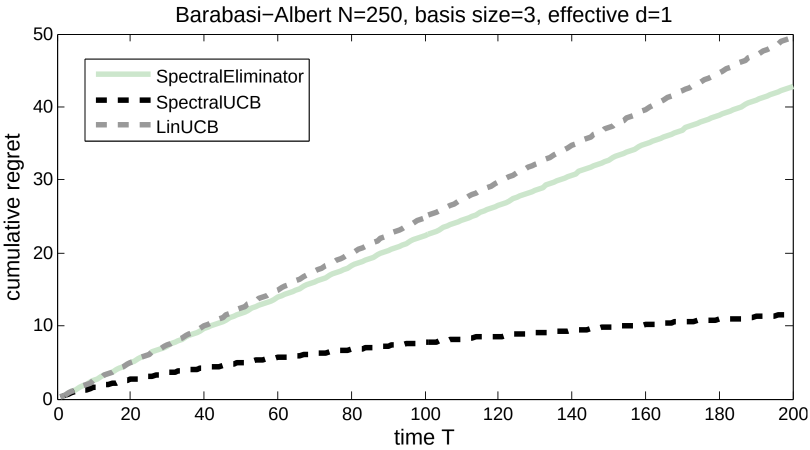
$$\widehat{\boldsymbol{\alpha}}_t = \arg\min_{\boldsymbol{\alpha}} \left(\sum_{v=1}^{t} \left[\langle \mathbf{x}_v, \boldsymbol{\alpha} \rangle - r_v\right]^2 + \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 \right).$$

$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$ is a penalty for non-smooth combinations of eigenvectors.

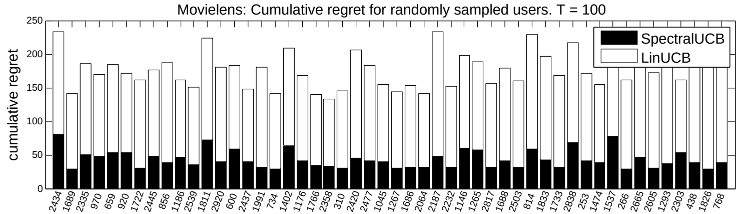
SpectralUCB

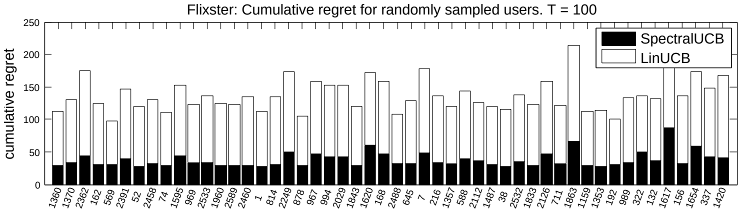
Input:
N, T, $\{\mathbf{\Lambda_L}, \mathbf{Q}\}$, λ, δ, R, C
Run:
$\mathbf{\Lambda} \leftarrow \mathbf{\Lambda_L} + \lambda \mathbf{I}$
$d \leftarrow \max\{d : (d-1)\lambda_d \leq T/\ln(1 + T/\lambda)\}$
for $t = 1$ **to** T **do**
Update the basis coefficients $\widehat{\boldsymbol{\alpha}}$:
$\mathbf{X}_t \leftarrow [\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(t-1)}]^\mathsf{T}$
$\mathbf{r} \leftarrow [r_1, \ldots, r_{t-1}]^\mathsf{T}$
$\mathbf{V}_t \leftarrow \mathbf{X}_t \mathbf{X}_t^\mathsf{T} + \mathbf{\Lambda}$
$\widehat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1} \mathbf{X}_t^\mathsf{T} \mathbf{r}$
$c_t \leftarrow 2R\sqrt{d \ln(1 + t/\lambda) + 2\ln(1/\delta)} + C$
$\pi(t) \leftarrow \arg\max_a \left(\mathbf{x}_a^\mathsf{T} \widehat{\boldsymbol{\alpha}} + c_t \|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}} \right)$
Observe the reward r_t
end for

SpectralUCB: Synthetic experiment



SpectralUCB: Movie data experiments





SpectralUCB: Regret Bound

- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

The **cumulative regret** R_T of **SpectralUCB** is with probability $1 - \delta$ bounded as

$$R_T \leq \left(8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4 \right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

$$R_T \approx d\sqrt{T \ln T}$$

SpectralUCB: Regret Bound

- Derivation of the confidence ellipsoid for $\widehat{\boldsymbol{\alpha}}$ with probability $1-\delta$.
  - Using analysis of OFUL (Abbasi-Yadkori et al., 2011)

$$|\mathbf{x}^{\mathsf{T}}(\widehat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}^*)| \leq \|\mathbf{x}\|_{\mathbf{V}_t^{-1}} \left(R\sqrt{2\ln\left(\frac{|\mathbf{V}_t|^{1/2}}{\delta|\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right)$$

- Regret in one time step: $r_t = \mathbf{x}_*^{\mathsf{T}}\boldsymbol{\alpha}^* - \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* \leq 2c_t\|\mathbf{x}_{\pi(t)}\|_{\mathbf{V}_t^{-1}}$
- Cumulative regret:

$$R_T = \sum_{t=1}^{T} r_t \leq \sqrt{T\sum_{t=1}^{T} r_t^2} \leq 2(c_T+1)\sqrt{2T\ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|}}$$

- Upperbound for $\ln(|\mathbf{V}_t|/|\boldsymbol{\Lambda}|)$

$$\ln\frac{|\mathbf{V}_t|}{|\boldsymbol{\Lambda}|} \leq \ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|} \leq 2d\ln\left(\frac{\lambda+T}{\lambda}\right)$$

SpectralUCB: Regret Bound

Sylvester's determinant theorem:

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}|(1 + \mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x})$$

Goal:

- Upperbound determinant $|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}|$ for $\|\mathbf{x}\|_2 \leq 1$
- Upperbound $\mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x}$

$$\mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x} = \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^\mathsf{T}\mathbf{x} = \mathbf{y}^\mathsf{T}\mathbf{\Lambda}^{-1}\mathbf{y} = \sum_{i=1}^{N} \lambda_i^{-1} y_i^2$$

- $\|\mathbf{y}\|_2 \leq 1.$
- $\mathbf{y}$ is a canonical vector.
- $\mathbf{x} = \mathbf{Q}\mathbf{y}$ is an eigenvector of $\mathbf{A}$.

SpectralUCB: Regret Bound

Corollary: Determinant $|\mathbf{V}_T|$ of $\mathbf{V}_T = \mathbf{\Lambda} + \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\mathsf{T}$ is maximized when all $\mathbf{x}_t$ are aligned with axes.

$$|\mathbf{V}_T| \leq \max_{\sum t_i = T} \prod (\lambda_i + t_i)$$

$$\ln \frac{|\mathbf{V}_T|}{|\mathbf{\Lambda}|} \leq \max_{\sum t_i = T} \sum \ln \left(1 + \frac{t_i}{\lambda_i}\right)$$

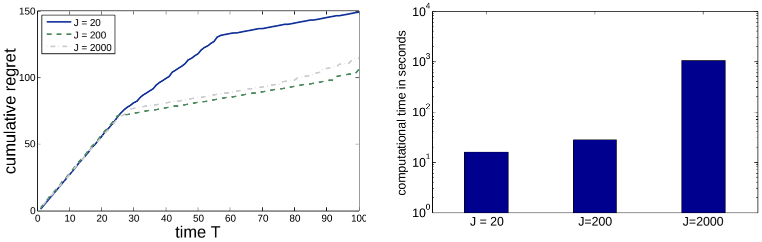
$$\ln \frac{|\mathbf{V}_T|}{|\mathbf{\Lambda}|} \leq \sum_{i=1}^{d} \ln \left(1 + \frac{T}{\lambda}\right) + \sum_{i=d+1}^{N} \ln \left(1 + \frac{t_i}{\lambda_{d+1}}\right)$$

$$\leq d \ln \left(1 + \frac{T}{\lambda}\right) + \frac{T}{\lambda_{d+1}}$$

$$\leq 2d \ln \left(1 + \frac{T}{\lambda}\right)$$

SpectralUCB: Improving the running time

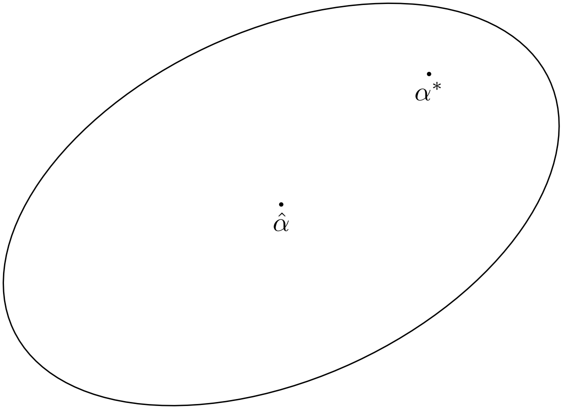
- **Reduced basis:** We only need first few eigenvectors.
- **Getting J eigenvectors:** $\mathcal{O}(Jm \log m)$ time for m edges
- Computationally less expensive, comparable performance.



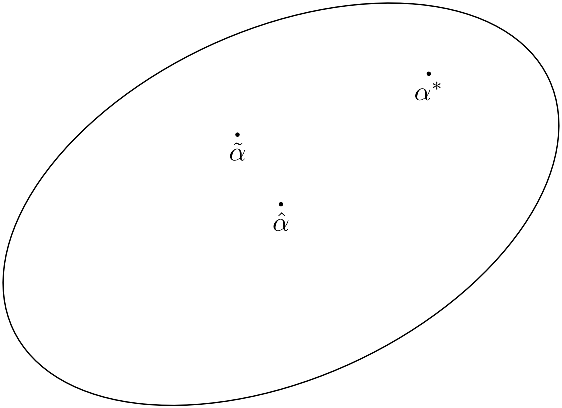
SpectralUCB: How to make it even faster?

- UCB-style algorithms need to (re)-compute UCBs every t
- Can be a problem for large set of arms $\rightarrow D^2 N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best
    - Sample $\widetilde{\boldsymbol{\alpha}}$ from the distribution $\mathcal{N}(\widehat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
    - Play arm which maximizes $\mathbf{x}^{\mathsf{T}}\widetilde{\boldsymbol{\alpha}}$ and observe reward
  - Compute posterior distribution according to reward received
- Only requires $D^2 + DN \rightarrow N^2$ per step update

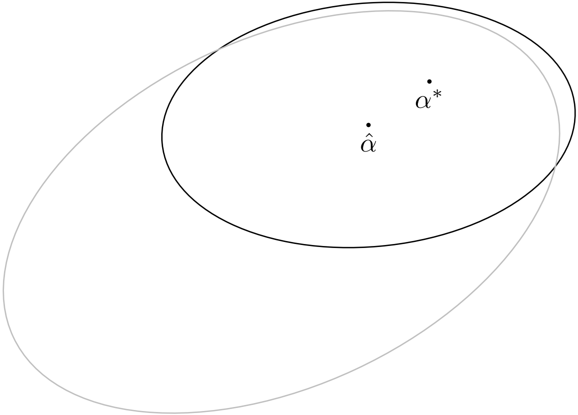
Thomson Sampling: Estimate



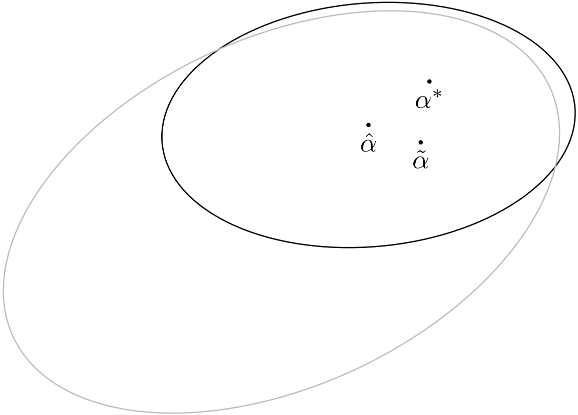
Thomson Sampling: Sample



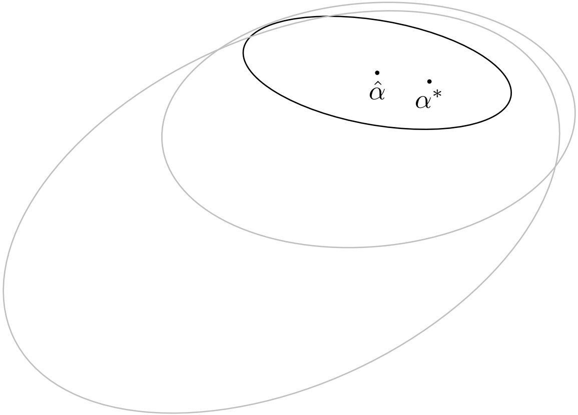
Thomson Sampling: Estimate



Thomson Sampling: Sample



Thomson Sampling: Estimate . . .



SpectralTS for Graphs

Input:
$N,\ T,\ \{\boldsymbol{\Lambda}_{\mathbf{L}}, \mathbf{Q}\},\ \lambda,\ \delta,\ R,\ C$
Initialization:
$v = R\sqrt{6d\log((\lambda + T)/\delta\lambda)} + C$
$\widehat{\boldsymbol{\alpha}} = 0_N$
$\mathbf{f} = 0_N$
$\mathbf{V} = \boldsymbol{\Lambda}_{\mathbf{L}} + \lambda\mathbf{I}_N$
Run:
for $t = 1$ **to** T **do**
Sample $\widetilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\widehat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
$\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}}\widetilde{\boldsymbol{\alpha}}$
Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$
$\mathbf{f} \leftarrow \mathbf{f} + \mathbf{x}_{\pi(t)}r(t)$
Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^{\mathsf{T}}$
Update $\widehat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\mathbf{f}$
end for

SpectralTS: Regret bound

- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

The **cumulative regret** R_T of **SpectralTS** is with probability $1 - \delta$ bounded as

$$\mathcal{R}_T \leq \frac{11g}{p}\sqrt{\frac{4+4\lambda}{\lambda} dT \log \frac{\lambda + T}{\lambda}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T \log \frac{2}{\delta}},$$

where $p = 1/(4e\sqrt{\pi})$ and

$$g = \sqrt{4 \log TN}\left(R\sqrt{6d \log\left(\frac{\lambda + T}{\delta \lambda}\right)} + C\right) + R\sqrt{2d \log\left(\frac{(\lambda + T)T^2}{\delta \lambda}\right)} + C.$$

$$R_T \approx d\sqrt{T \log N}$$

SpectralTS: Analysis sketch

Divide arms into two groups

- $\Delta_i = \mathbf{x}_*^\mathsf{T}\alpha - \mathbf{x}_i^\mathsf{T}\alpha \leq g\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$ arm i is **unsaturated**
- $\Delta_i = \mathbf{x}_*^\mathsf{T}\alpha - \mathbf{x}_i^\mathsf{T}\alpha > g\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$ arm i is **saturated**

Saturated arm

- Small standard deviation $\rightarrow$ accurate regret estimate.
- **High regret** on playing the arm $\rightarrow$ **Low probability** of picking

Unsaturated arm

- **Low regret** bounded by a factor of standard deviation
- **High probability** of picking

SpectralTS: Analysis sketch

- Confidence ellipsoid for estimate $\widehat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ (with probability $1-\delta/T^2$)
  - Using analysis of OFUL algorithm (Abbasi-Yadkori et al., 2011)

$$|\mathbf{x}_i^{\mathsf{T}}\widehat{\boldsymbol{\alpha}} - \mathbf{x}_i^{\mathsf{T}}\boldsymbol{\alpha}| \leq \left(R\sqrt{2\,d\log\left(\frac{(\lambda+T)T^2}{\delta\lambda}\right)} + C\right)\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}} = \ell\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$$

- The key result coming from spectral properties of $\mathbf{V}_t$.

$$\log\frac{|\mathbf{V}_t|}{|\boldsymbol{\Lambda}|} \leq 2d\log\left(1+\frac{T}{\lambda}\right)$$

- Concentration of sample $\widetilde{\boldsymbol{\alpha}}$ around mean $\widehat{\boldsymbol{\alpha}}$ (with probability $1-1/T^2$)
  - Using concentration inequality for Gaussian random variable.

$$|\mathbf{x}_i^{\mathsf{T}}\widetilde{\boldsymbol{\alpha}} - \mathbf{x}_i^{\mathsf{T}}\widehat{\boldsymbol{\alpha}}| \leq \left(R\sqrt{6d\log\left(\frac{\lambda+T}{\delta\lambda}\right)} + C\right)\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\sqrt{4\log(TN)} = v\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\sqrt{4\log(TN)}$$

SpectralTS: Analysis sketch

Define $\text{regret}'(t) = \text{regret}(t) \cdot \mathbb{1}\{|\mathbf{x}_i^\mathsf{T}\widehat{\boldsymbol{\alpha}}(t) - \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}| \leq \ell \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\}$

$$\text{regret}'(t) \leq \frac{11g}{p}\|\mathbf{x}_{a(t)}\|_{\mathbf{V}_t^{-1}} + \frac{1}{T^2}$$

Super-martingale (i.e. $\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] \leq 0$)

$$X_t = \text{regret}'(t) - \frac{11g}{p}\|\mathbf{x}_{a(t)}\|_{\mathbf{V}_t^{-1}} - \frac{1}{T^2}$$

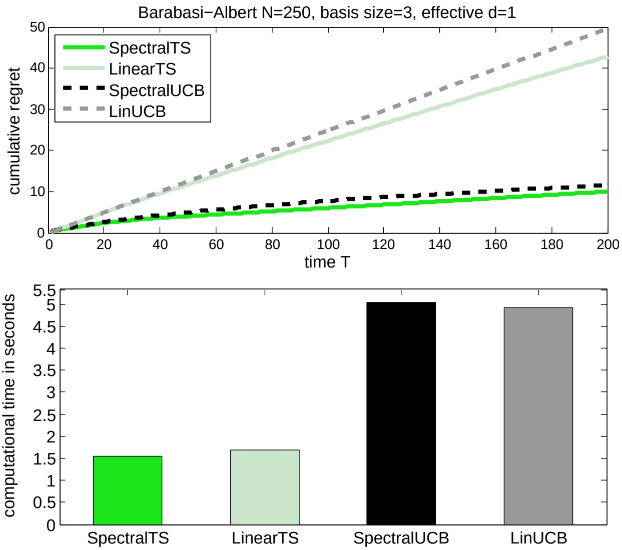
$$Y_t = \sum_{w=1}^{t} X_w.$$

$(Y_t;\ t = 0, \ldots, T)$ is a **super-martingale** process w.r.t. history $\mathcal{F}_t$.

Azuma-Hoeffding inequality for super-martingales, w.p. $1 - \delta/2$:

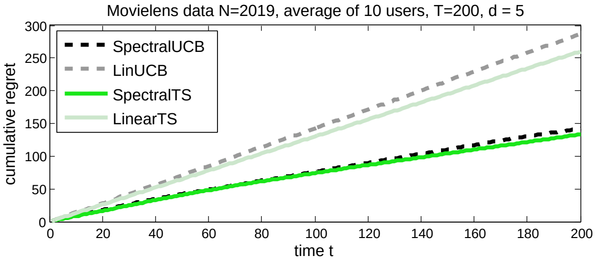
$$\sum_{t=1}^{T} \text{regret}'(t) \leq \frac{11g}{p} \sum_{t=1}^{T} \|\mathbf{x}_{a(t)}\|_{\mathbf{V}_t^{-1}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T \ln \frac{2}{\delta}}$$

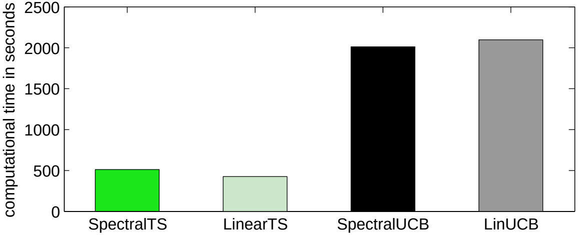
Spectral Bandits: Synthetic experiment



Spectral Bandits: Real world experiment

MovieLens dataset of 6k users who rated one million movies.





Spectral Bandits Summary

- Spectral bandit setting **(smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$
- **SpectralTS**
  - Regret bound $R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln N}\right)$
  - Computationally more efficient.
- **SpectralEliminator**
  - Regret bound $R_T = \widetilde{\mathcal{O}}\left(\sqrt{dT \ln T}\right)$
  - Better upper, empirically does not seem to work well (yet)
- Bounds scale with **effective dimension** $d \ll D$.

SpectralEliminator: Pseudocode

Input:
N : the number of nodes, T : the number of pulls
$\{\mathbf{\Lambda_L}, \mathbf{Q}\}$ spectral basis of $\mathbf{L}$
λ : regularization parameter
$\beta, \{t_j\}_j^J$ parameters of the elimination and phases
$A_1 \leftarrow \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.
for $j = 1$ **to** J **do**
$\mathbf{V}_{t_j} \leftarrow \gamma \mathbf{\Lambda_L} + \lambda \mathbf{I}$
for $t = t_j$ **to** $\min(t_{j+1} - 1, T)$ **do**
Play $\mathbf{x}_t \in A_j$ with the largest width to observe r_t:
$\mathbf{x}_t \leftarrow \arg\max_{\mathbf{x} \in A_j} \|\mathbf{x}\|_{\mathbf{V}_t^{-1}}$
$\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \mathbf{x}_t \mathbf{x}_t^\mathsf{T}$
end for
Eliminate the arms that are not promising:
$\widehat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1} [\mathbf{x}_{t_j}, \ldots, \mathbf{x}_t][r_{t_j}, \ldots, r_t]^\mathsf{T}$
$A_{j+1} \leftarrow \left\{ \mathbf{x} \in A_j, \langle \widehat{\boldsymbol{\alpha}}_t, \mathbf{x} \rangle + \|\mathbf{x}\|_{V_t^{-1}} \beta \geq \max_{\mathbf{x} \in A_j} \left[\langle \widehat{\boldsymbol{\alpha}}_t, \mathbf{x} \rangle - \|\mathbf{x}\|_{\mathbf{V}_t^{-1}} \beta \right] \right\}$
end for

SpectralEliminator: Analysis

SpectralEliminator

- Divide time into sets $(t_1 = 1 \leq t_2 \leq \dots)$ to introduce independence for Azuma-Hoeffding inequality and observe $R_T \leq \sum_{j=0}^{J}(t_{j+1} - t_j)\left[\langle \mathbf{x}^* - \mathbf{x}_t, \widehat{\alpha}_j \rangle + (\|\mathbf{x}^*\|_{\mathbf{V}_j^{-1}} + \|\mathbf{x}_t\|_{\mathbf{V}_j^{-1}})\beta\right]$
- Bound $\langle \mathbf{x}^* - \mathbf{x}_t, \widehat{\alpha}_j \rangle$ for each phase
- No bad arms: $\langle \mathbf{x}^* - \mathbf{x}_t, \widehat{\alpha}_j \rangle \leq (\|\mathbf{x}^*\|_{\mathbf{V}_j^{-1}} + \|\mathbf{x}_t\|_{\mathbf{V}_j^{-1}})\beta$
- By algorithm: $\|\mathbf{x}\|^2_{\mathbf{V}_j^{-1}} \leq \frac{1}{t_j - t_{j-1}} \sum_{s=t_{j-1}+1}^{t_j} \|\mathbf{x}_s\|^2_{\mathbf{V}_{s-1}^{-1}}$
- $\sum_{s=t_{j-1}+1}^{t_j} \min\left(1, \|\mathbf{x}_s\|^2_{\mathbf{V}_{s-1}^{-1}}\right) \leq \log \frac{|\mathbf{V}_j|}{|\mathbf{\Lambda}|}$

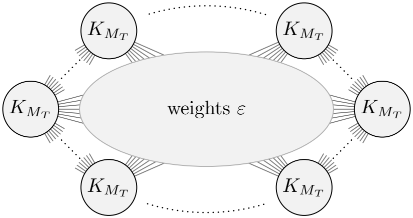
Spectral Bandits: Is it possible to do better?

Is d a good quantity that embodies the difficulty?

Lower bound!

For any d, we construct a graph that for any reasonable algorithm, the regret is at least $\Omega(\sqrt{dT})$.

How? By reduction to d-arm bandits problem.



Michal Valko
michal.valko@inria.fr
sequel.lille.inria.fr

SequeL – Inria Lille

MVA 2015/2016