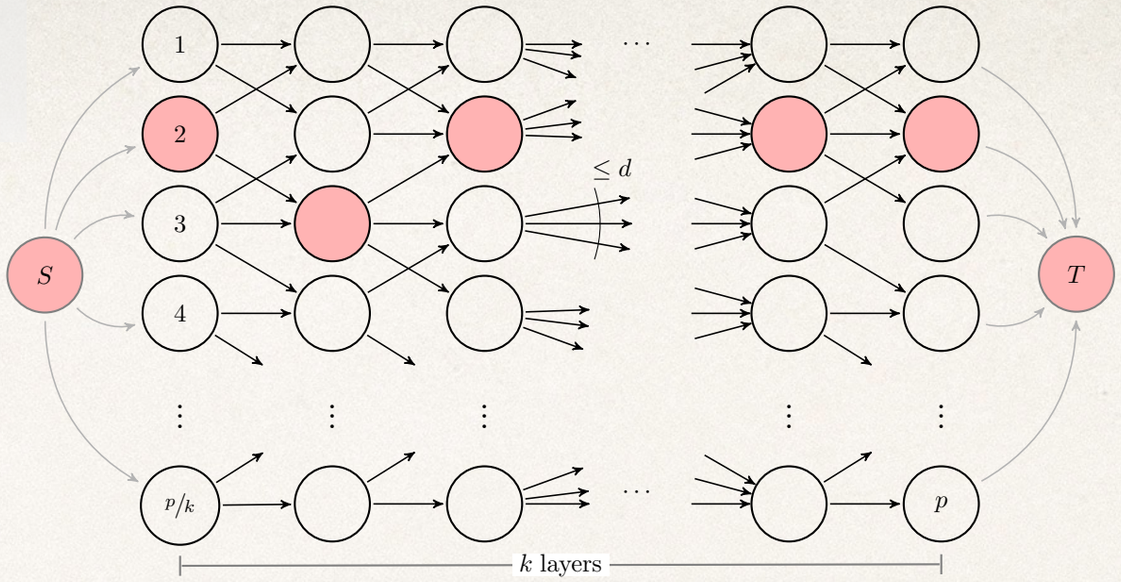
THE UNIVERSITY OF TEXAS AT AUSTIN



Stay on path: PCA along graph paths

Megasthenis Asteris
Anastasios Kyrillidis
Alexandros Dimakis

Department of Electrical and Computer Engineering

Han-Gyol Yi
Bharath Chandrasekaran

Department of Communication Sciences and Disorders

07/07/2015 - ICML

Supported by …

Setup and notation

Data:

$$\mathbf{Y} \in \mathbb{R}^{p \times n}$$

- p : number of variables.
- n : number of samples.
- In most cases, we work in the *high-dimensional regime* $n \ll p$.

Setup and notation

Data:

$$\mathbf{Y} \in \mathbb{R}^{p \times n}$$

- p : number of variables.
- n : number of samples.
- In most cases, we work in the *high-dimensional regime* $n \ll p$.

$$\hat{\mathbf{\Sigma}} = \frac{1}{n} \cdot \mathbf{Y}\mathbf{Y}^\top$$

$$\hat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}$$

- We look for away to find: (Dis)similarities among variables, patterns of variation.

PCA and Sparse PCA (in concept)

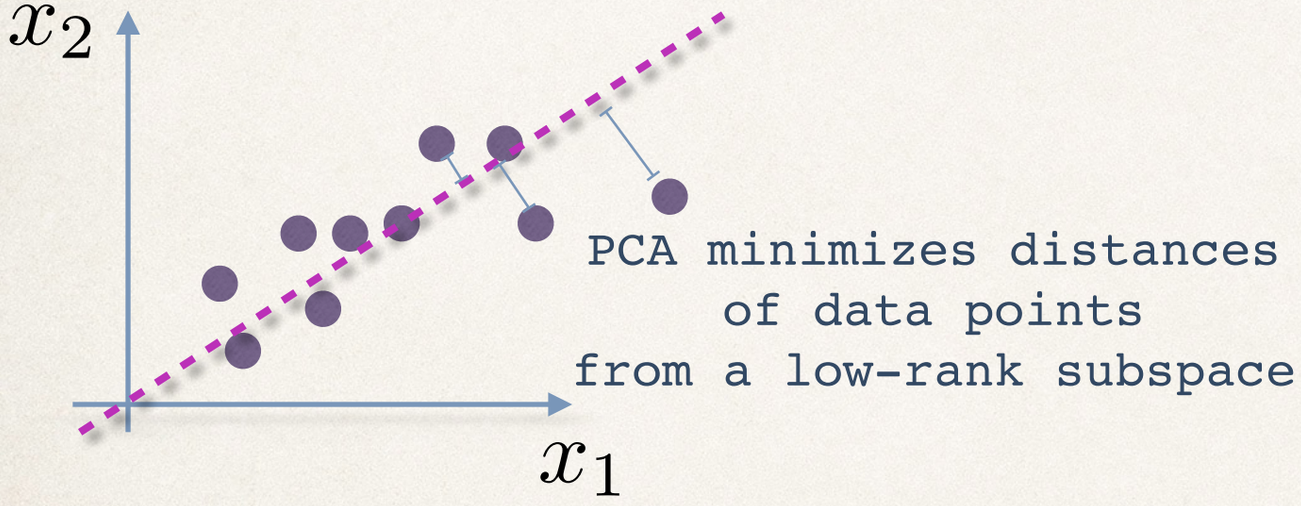
Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.

PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

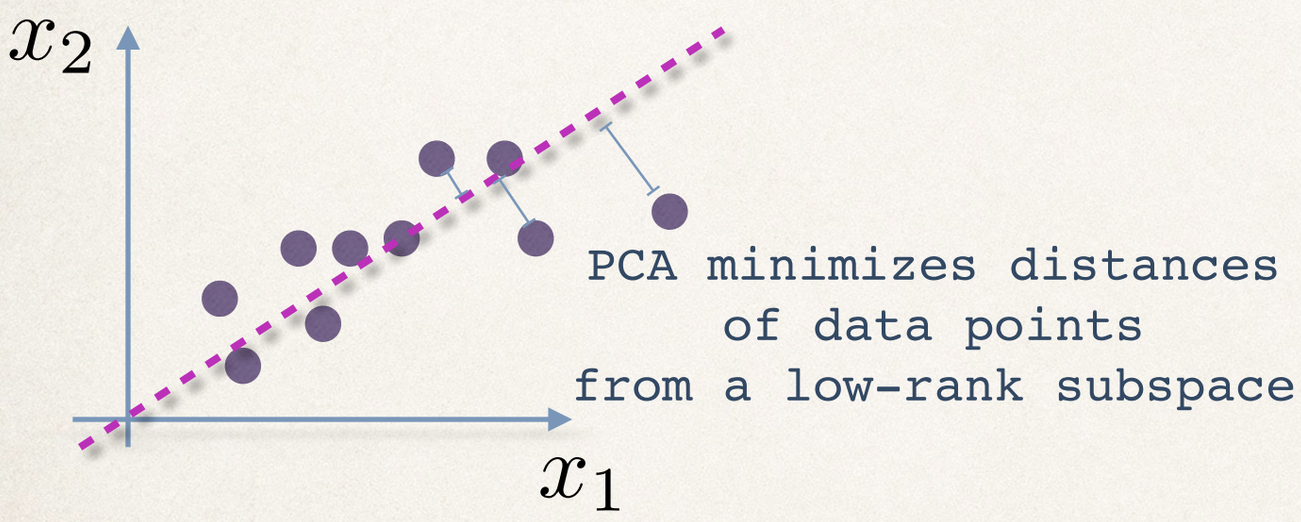
- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.

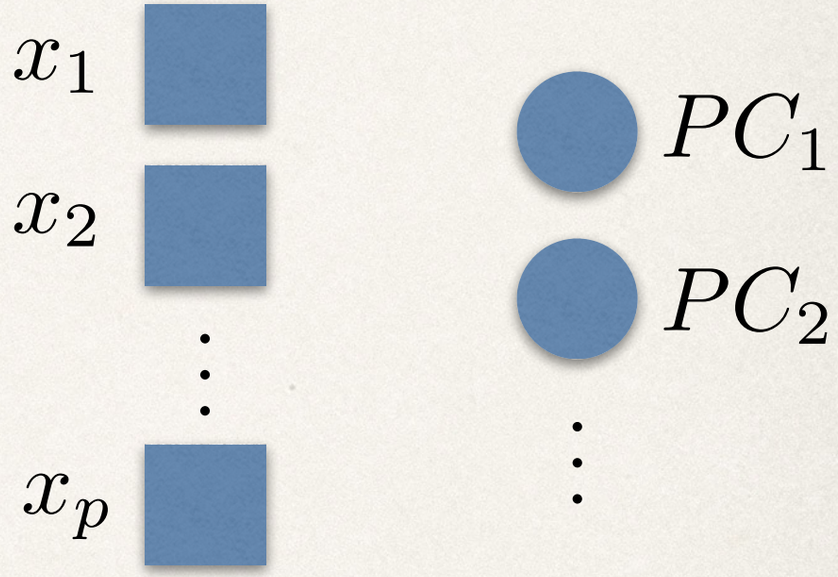


PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.

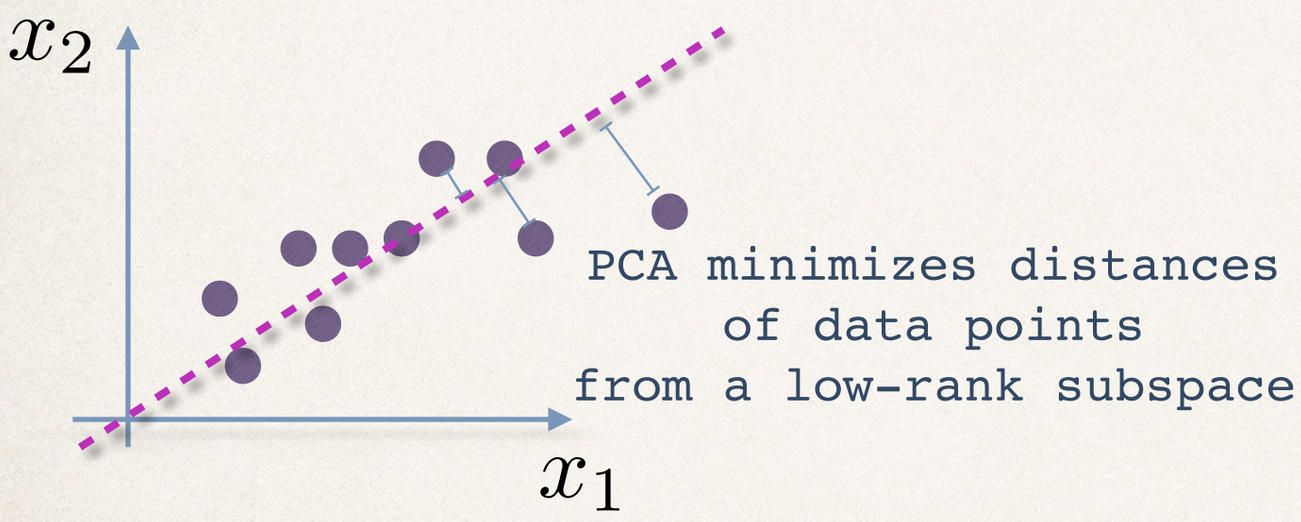


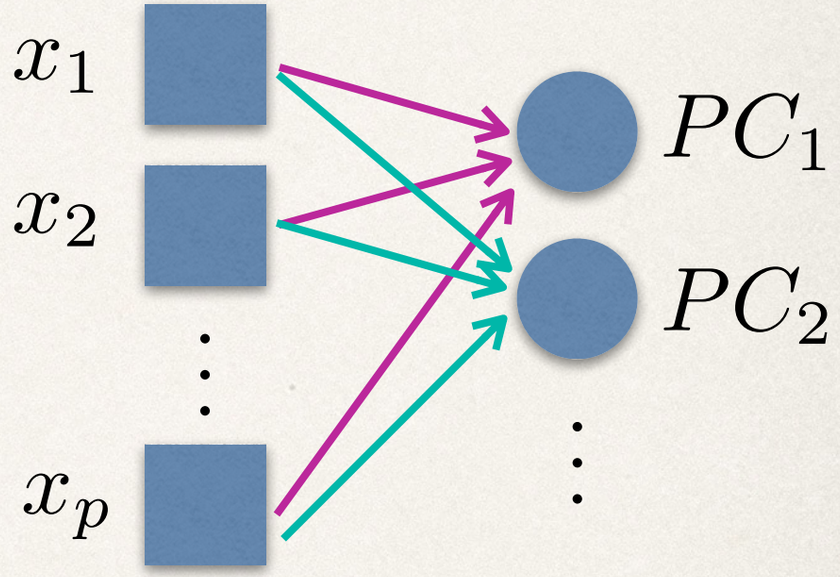


PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.

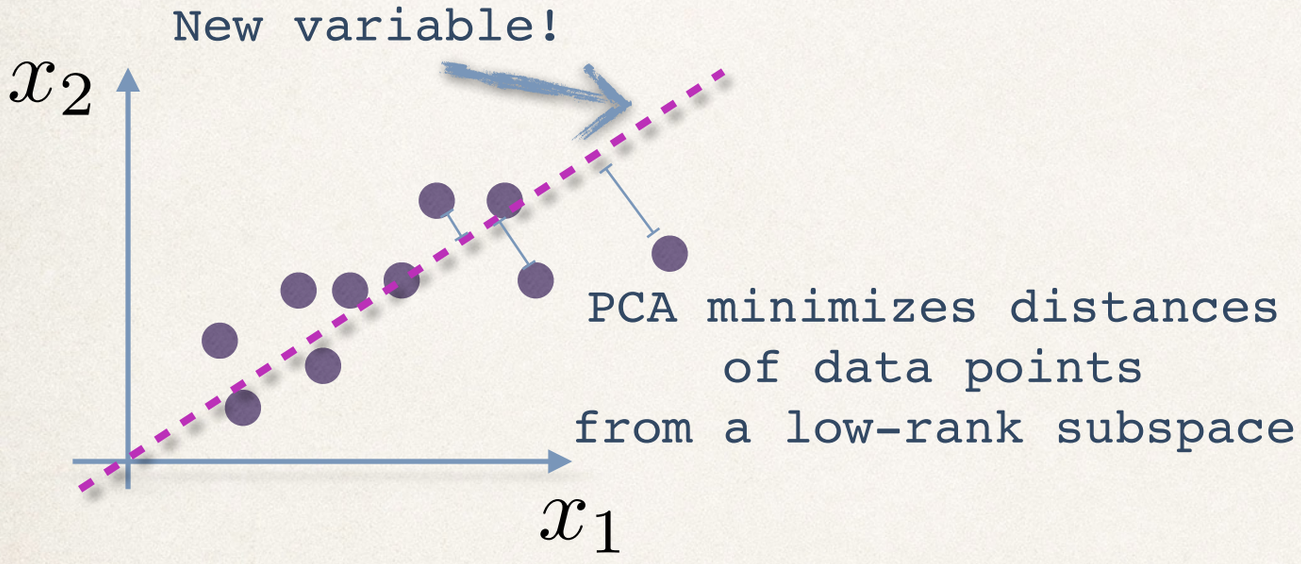


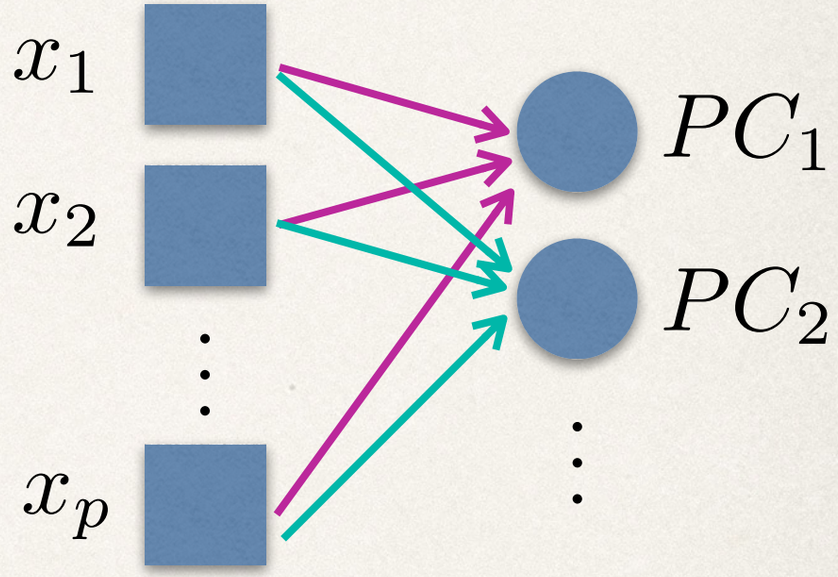


PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.

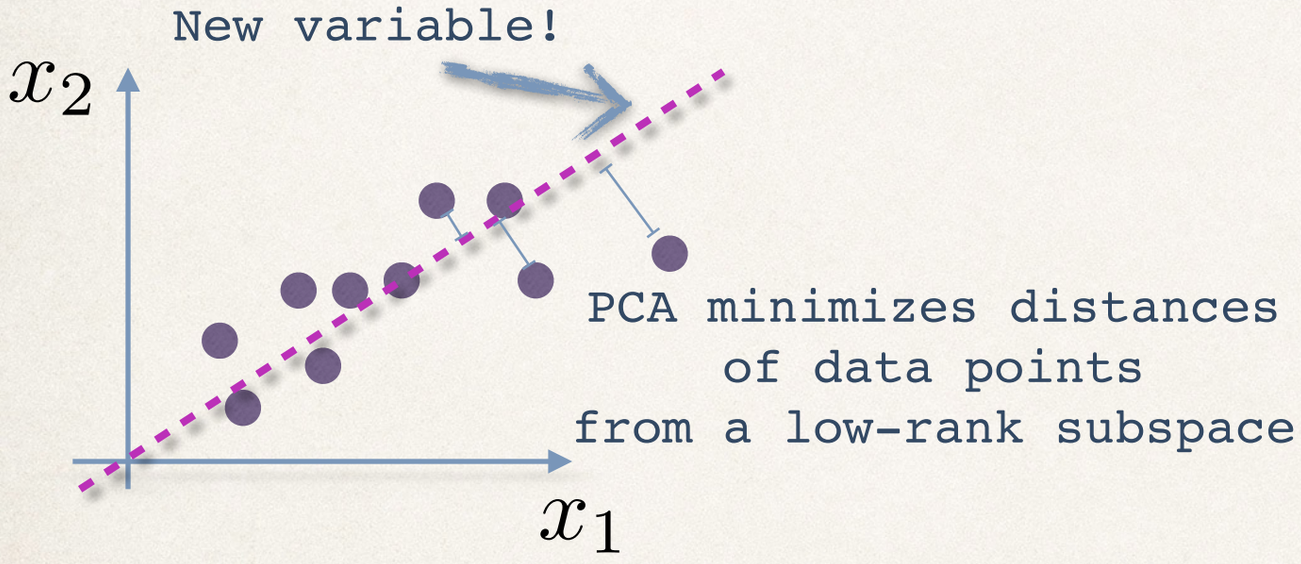


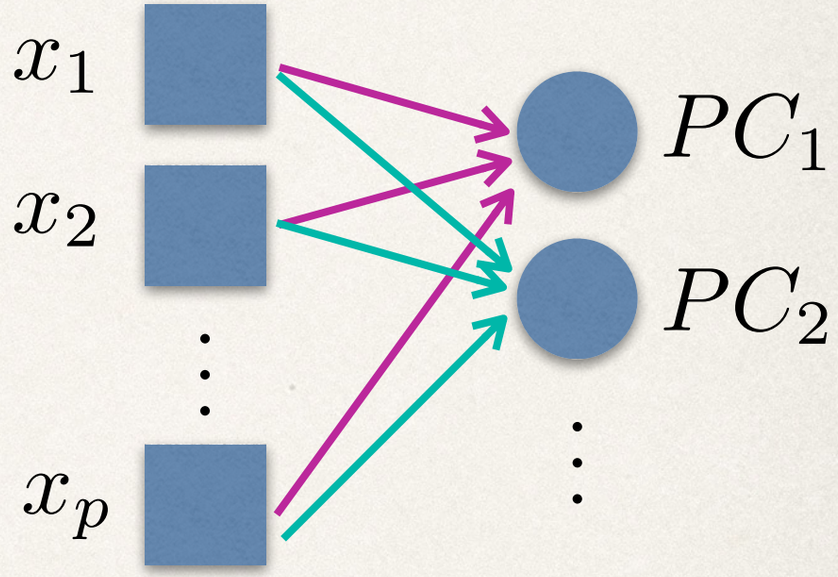


PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.





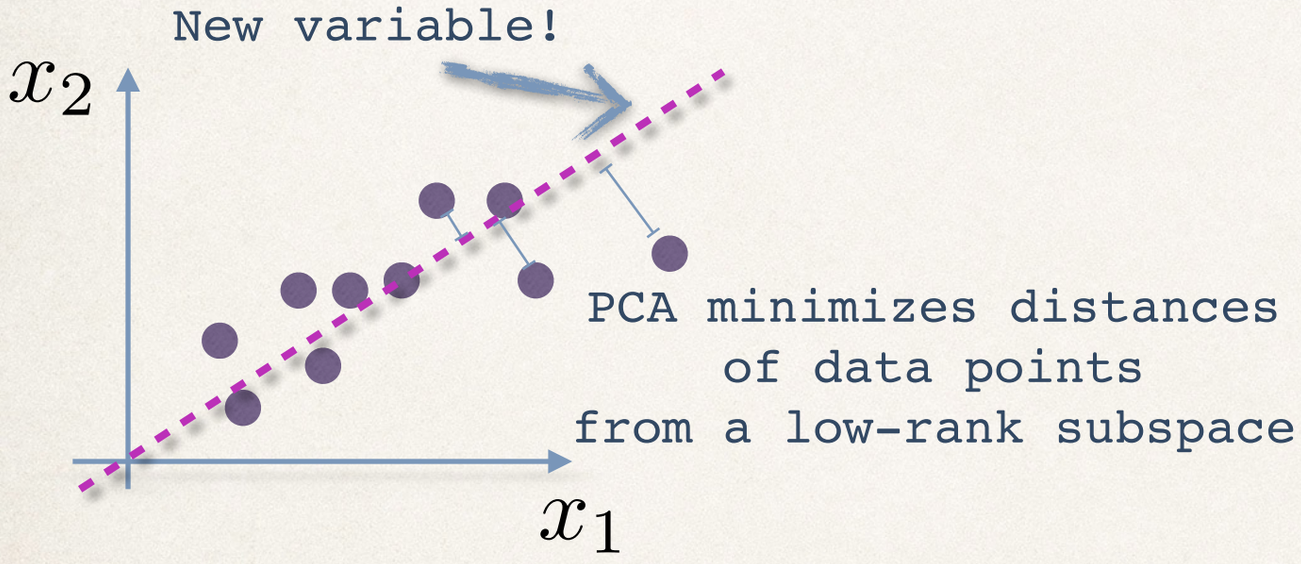
Sparse PCA [Johnstone & Lu, 2004]

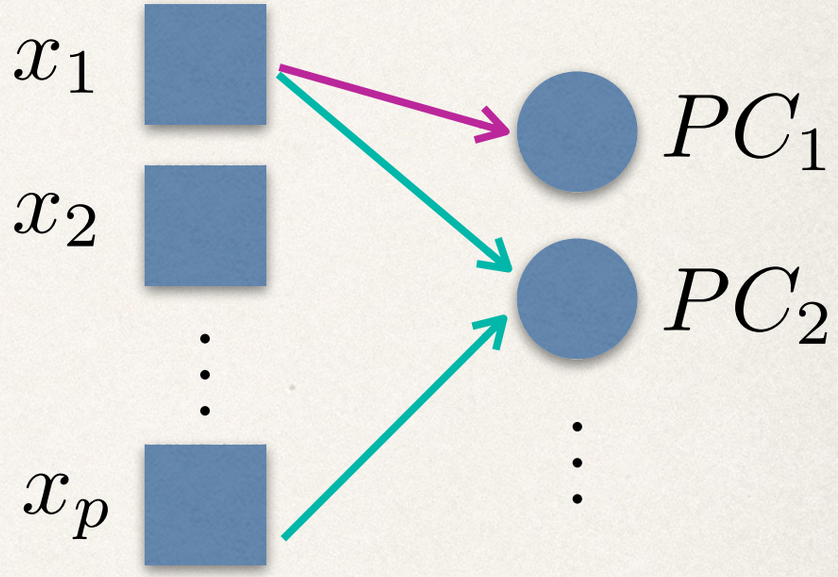
- Seeks subspaces that depend only on a few variables **(sparsity)**
- Improve **interpretabilty** (meta-analysis of the results).
- Numerically hard: sparsity is a combinatorial concept.

PCA and Sparse PCA (in concept)

Principal component analysis (PCA) [Jolliffe2002]

- Seeks for low-dimensional subspaces that explain well data variance.
- Classical dimensionality reduction tool.
- Numerically cheap for small- and moderate-sized problems via SVD.





Sparse PCA [Johnstone & Lu, 2004]

- Seeks subspaces that depend only on a few variables **(sparsity)**
- Improve **interpretabilty** (meta-analysis of the results).
- Numerically hard: sparsity is a combinatorial concept.

PCA and Sparse PCA (in math)

Principal component analysis (PCA) [Jolliffe2002]

- Given empirical covariance matrix $\hat{\mathbf{\Sigma}} = \frac{1}{n} \cdot \mathbf{Y}\mathbf{Y}^\top$, we solve:

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^p}{\text{maximize}} \quad & \mathbf{x}^\top \hat{\mathbf{\Sigma}} \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_2 = 1. \end{aligned}$$

Maximize data variance explained by the principal factor.

to get the first principal factor.

PCA and Sparse PCA (in math)

Principal component analysis (PCA) [Jolliffe2002]

- Given empirical covariance matrix $\hat{\mathbf{\Sigma}} = \frac{1}{n} \cdot \mathbf{Y}\mathbf{Y}^\top$, we solve:

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^p}{\text{maximize}} \quad & \mathbf{x}^\top \hat{\mathbf{\Sigma}} \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_2 = 1. \end{aligned}$$

Maximize data variance explained by the principal factor.

to get the first principal factor.

Sparse PCA [Johnstone & Lu, 2004; d'Aspremont et al., 2007]

- We further restrict the cardinality of principal factor as:

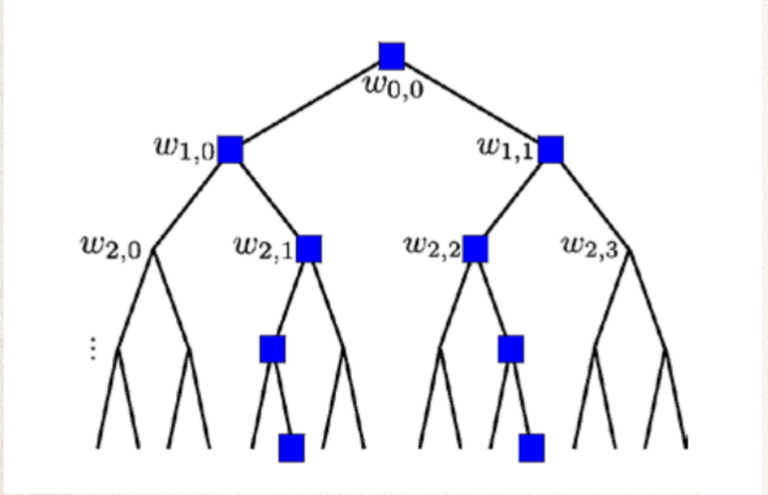
$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^p}{\text{maximize}} \quad & \mathbf{x}^\top \hat{\mathbf{\Sigma}} \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_2 = 1, \\ & \|\mathbf{x}\|_0 = k. \end{aligned}$$

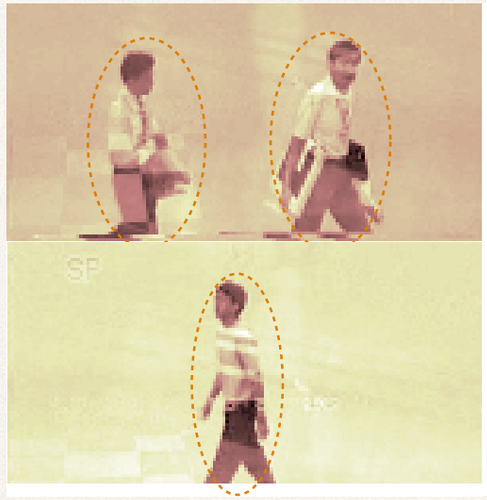
Beyond plain sparsity (to be updated)

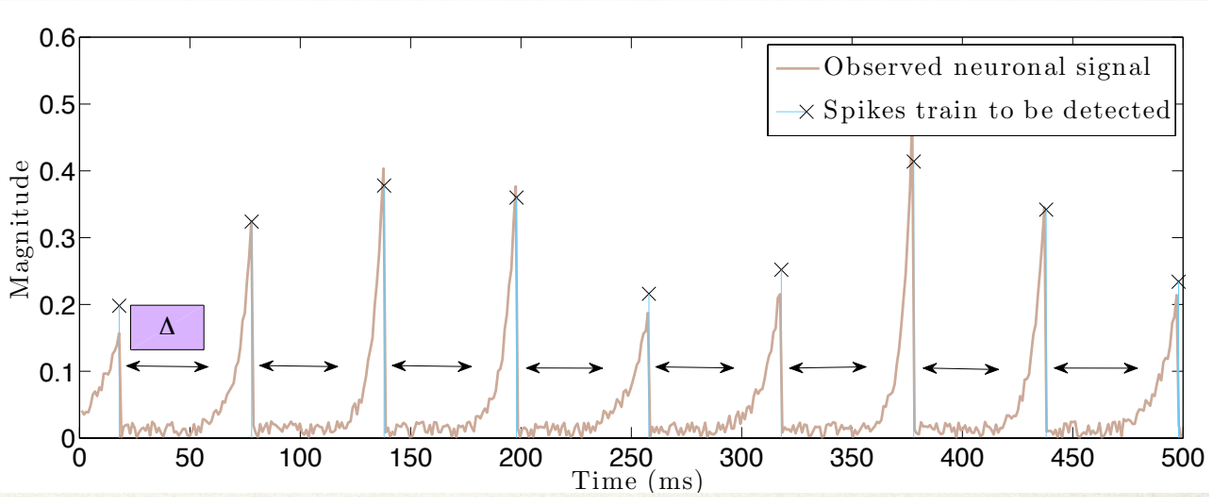
- In many cases, sparsity does not fully grasp the underlying structure

  e.g., Image processing (wavelets coeff. over trees, block sparsity),

  Neuronal signal processing, etc.
  [Baraniuk et al., 2008; Friedman et al., 2010, …]







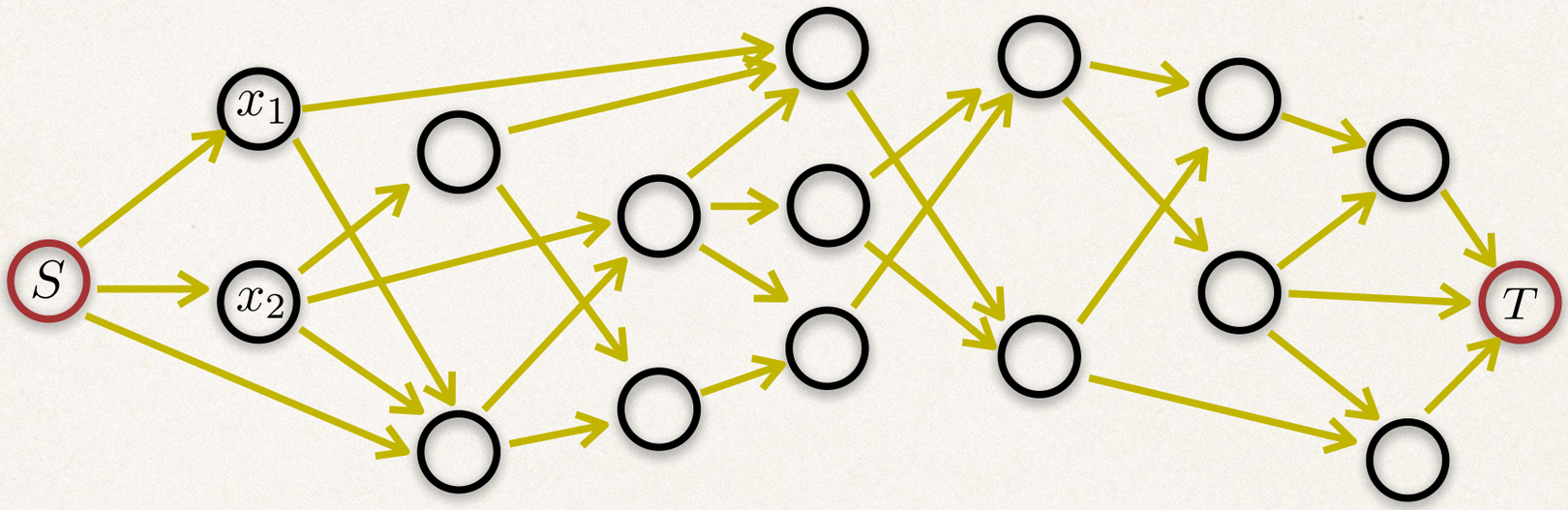
- Even within (sparse) PCA regime:

  Structured sparse PCA [Jenatton et al., 2010]

PCA on graph paths

PCA on graph paths

- Consider a *directed acyclic graph (DAG)* $G = (V, E)$ over variables.



$\mathcal{P}(G)$: collection of all $S - T$ paths.

- PCA on graph paths solves:

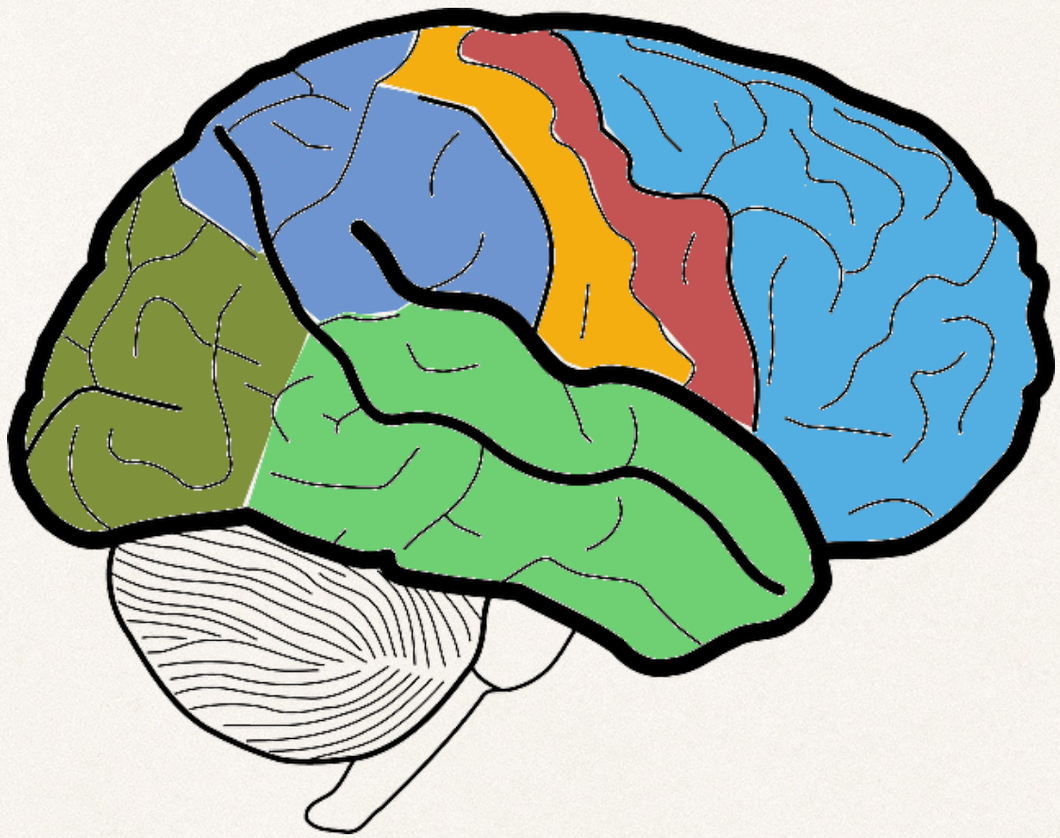
$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^p}{\text{maximize}} \quad & \mathbf{x}^\top \widehat{\mathbf{\Sigma}} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}(G). \end{aligned}$$

where:

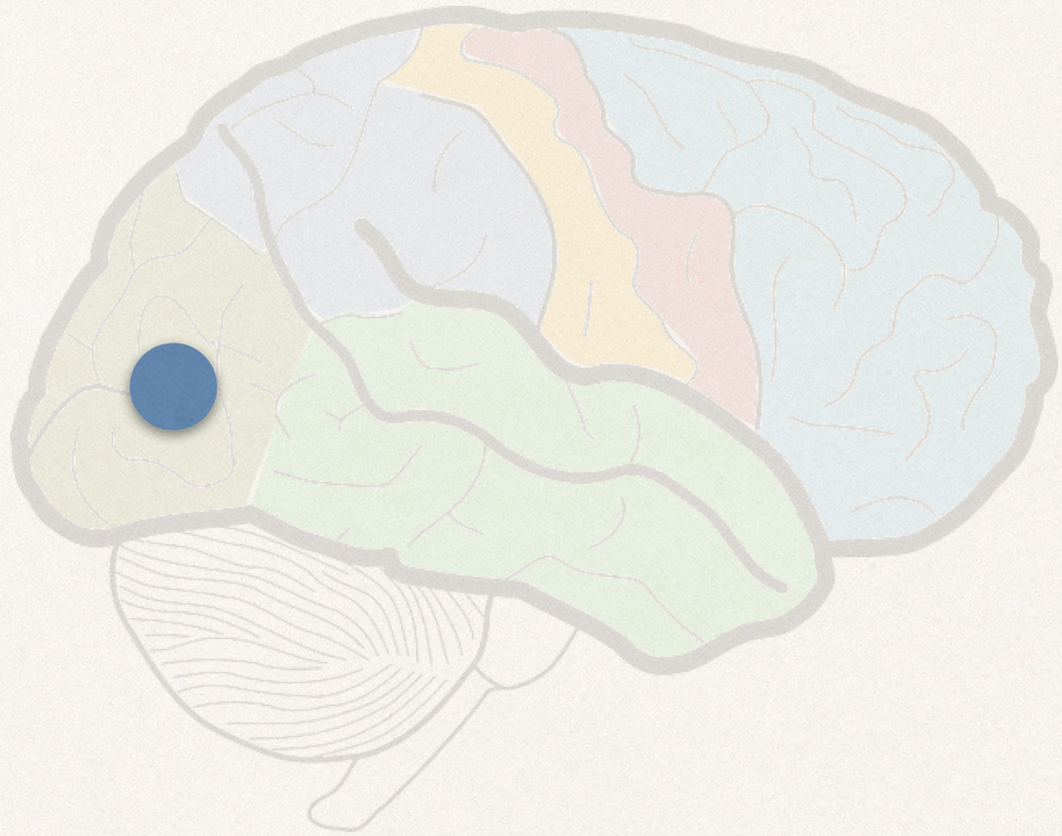
$$\mathcal{X}(G) = \{\mathbf{x} \in \mathbb{R}^p \ : \ \|\mathbf{x}\|_2 = 1, \text{supp}(\mathbf{x}) \in \mathcal{P}(G)\}$$

Motivation (through an example)
(why?)

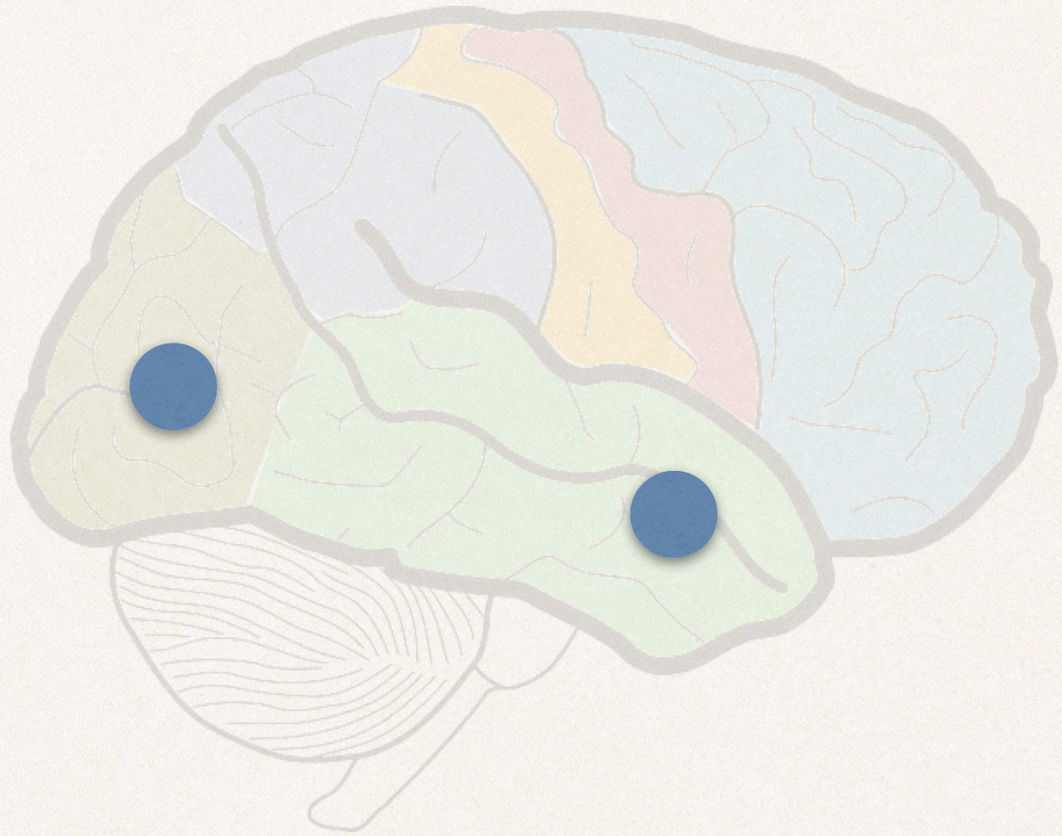
Find meaningful “streams” on brain



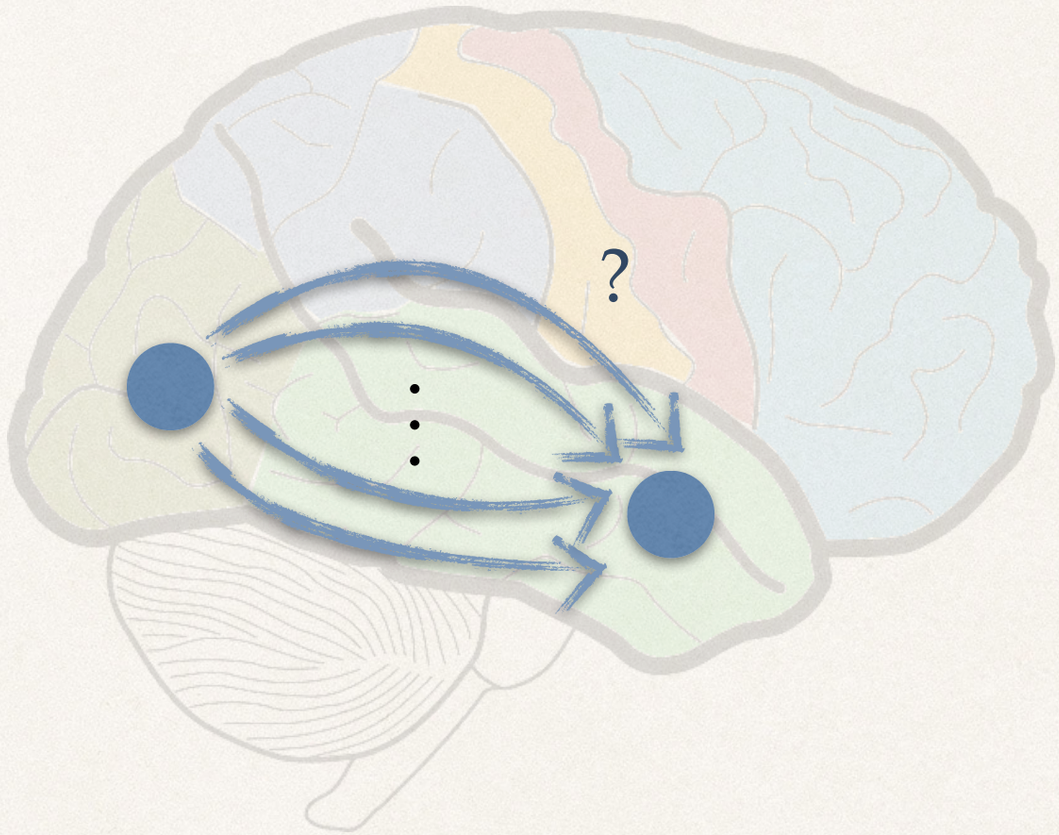
Find meaningful “streams” on brain



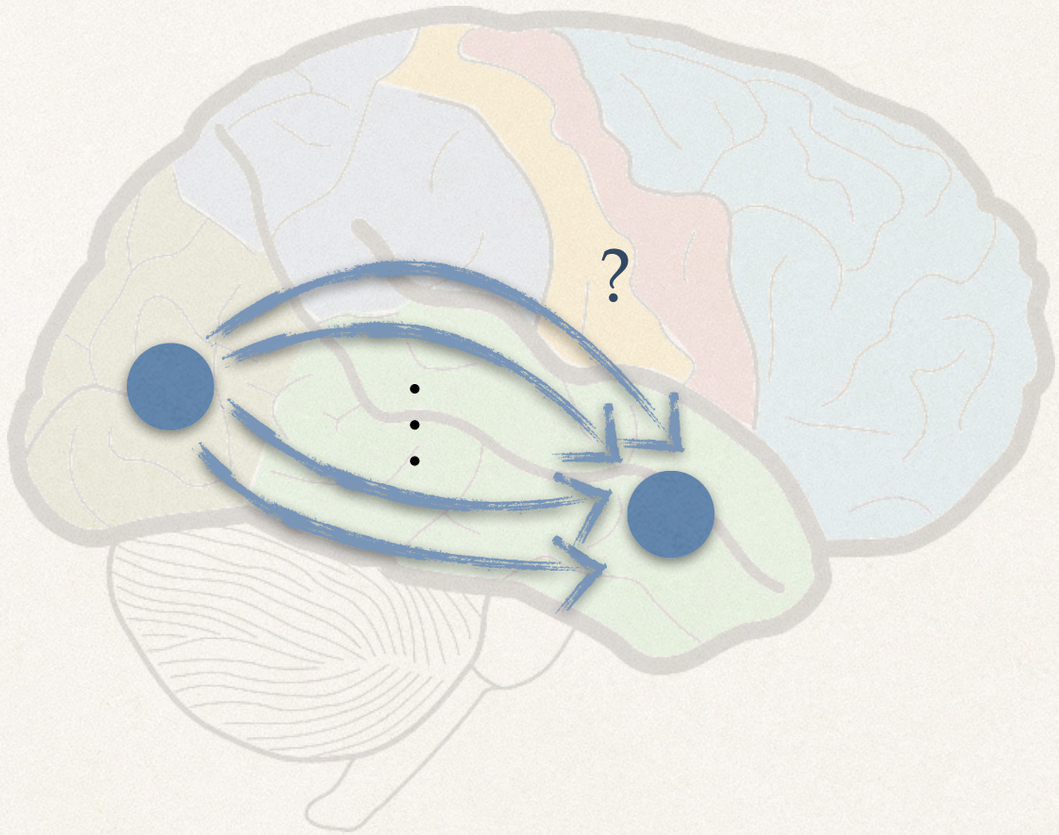
Find meaningful “streams” on brain



Find meaningful “streams” on brain



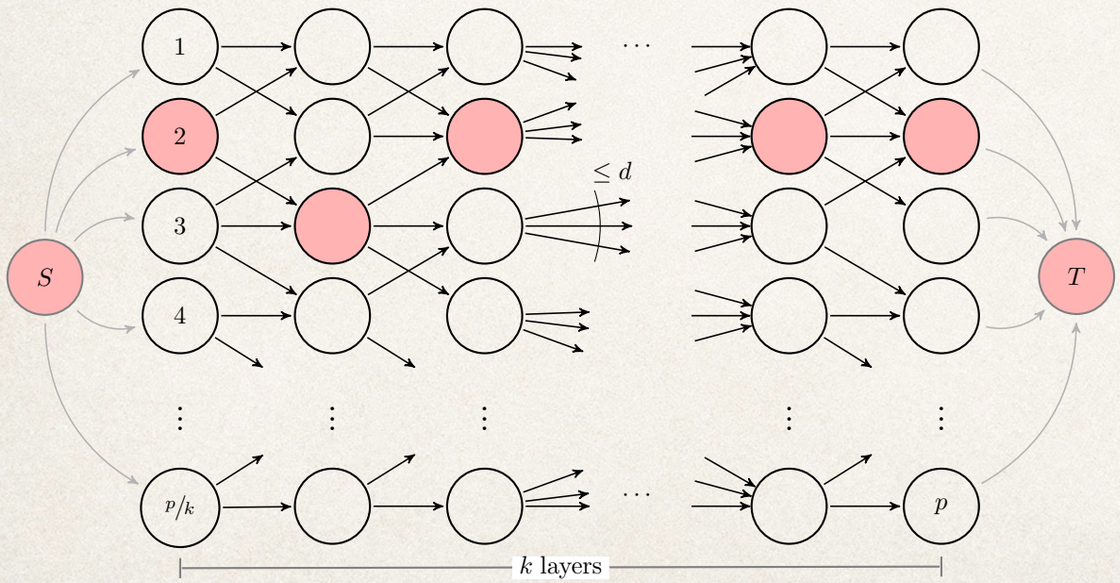
Find meaningful "streams" on brain

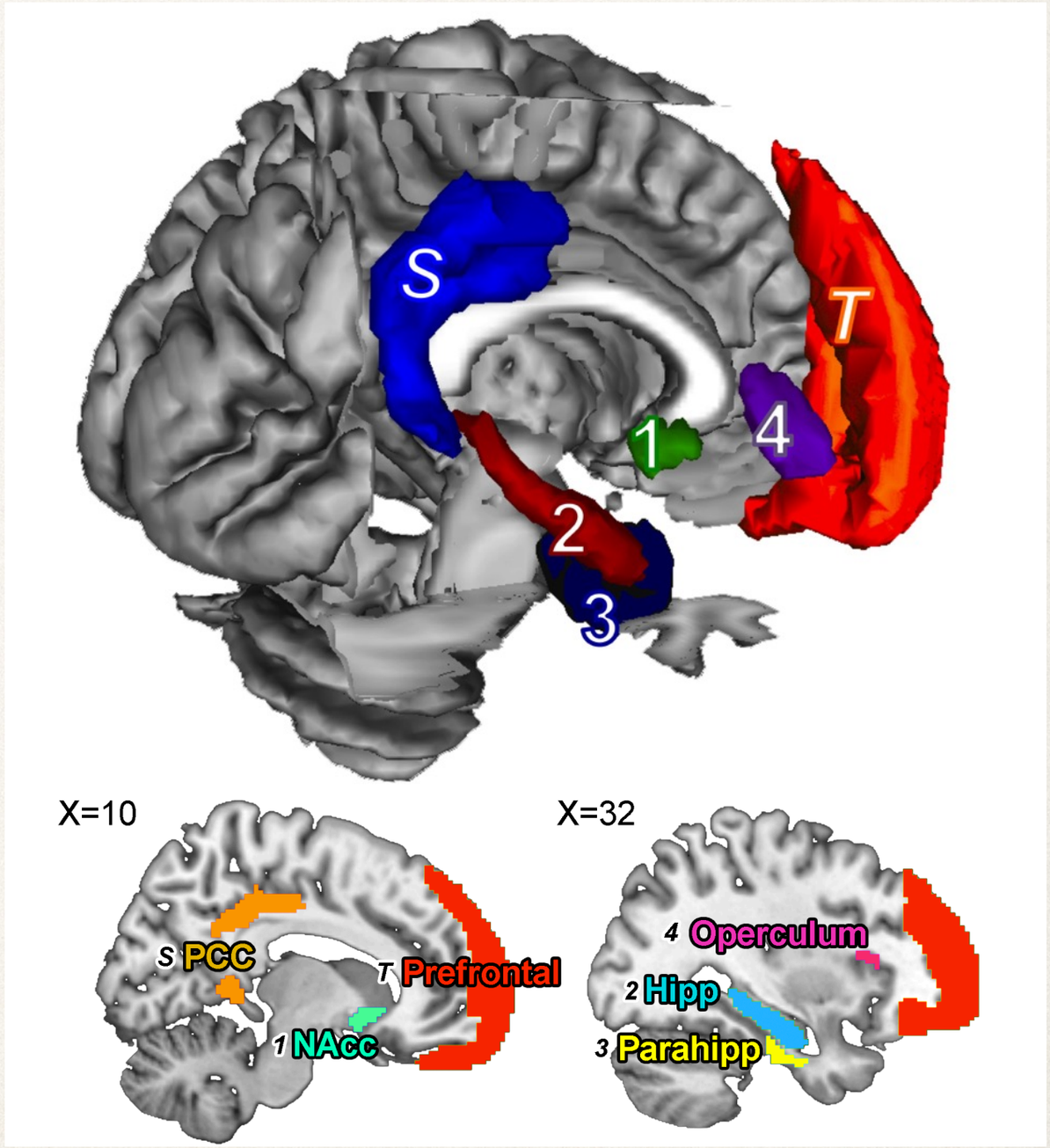


e.g., two stream hypothesis

Brain fMRI networks (preliminary results)

- Resting state fMRI dataset.*
- 111 regions of interest (ROIs) (variables), extracted based on Harvard-Oxford Atlas [Desikan et al., 2006].
- Graph extracted based on Euclidean distances between center of mass of ROIs.

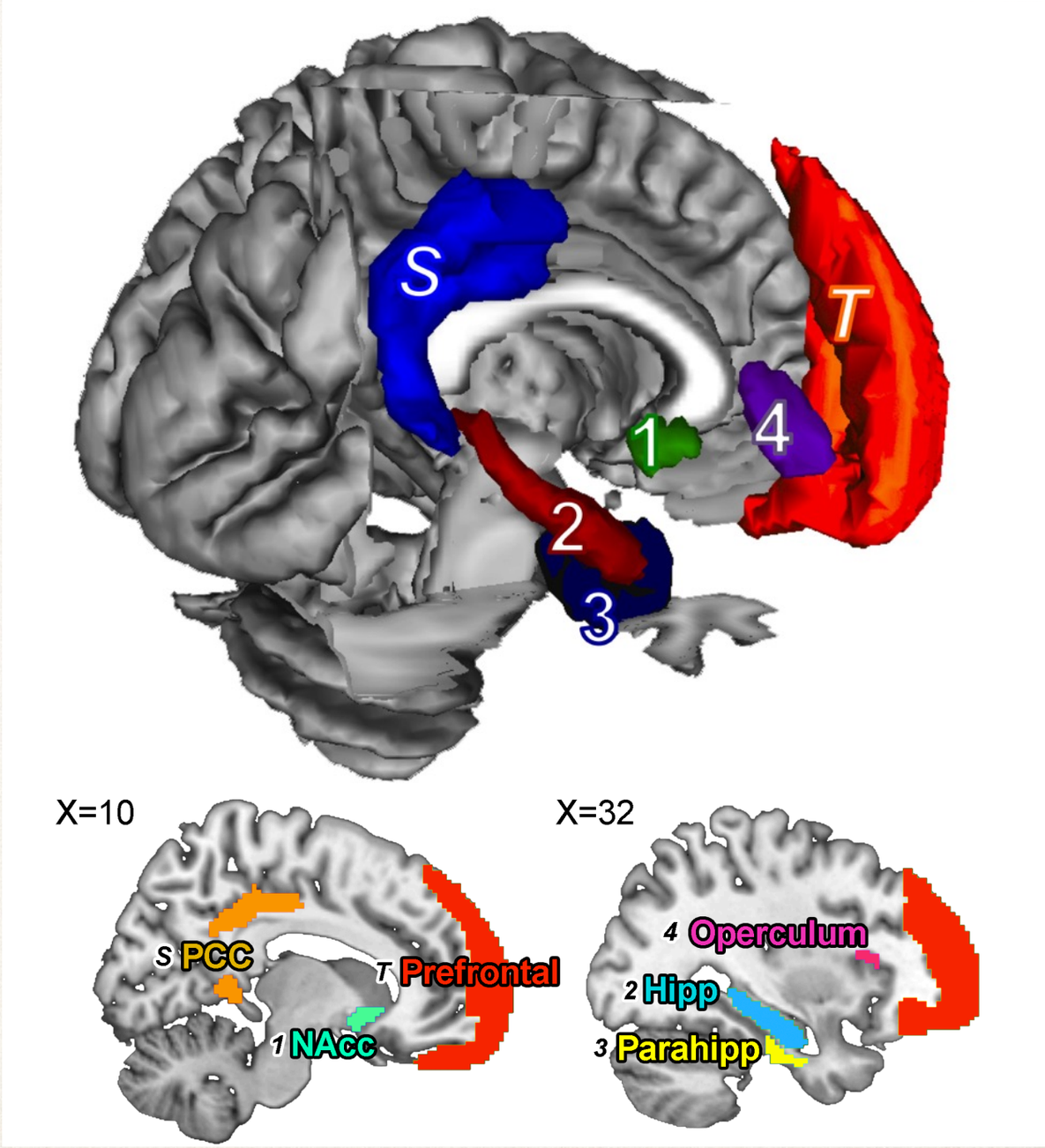




**Human Connectome Project, WU-Minn Consortium.*

Brain fMRI networks (preliminary results)

- Setup: 6 layers (sparsity)
- Starting node S:
  Posterior cingulate cortex
- Ending node T:
  Prefrontal cortex
- Path:
  1. Nucleus accumbens
  2. Hippocampus
  3. Parahippocampal gyrus
  4. Frontal operculum
- **All of them core neural components of the memory network.**



Αλγόριθμοι - Algorithms

(how?)

Graph-truncated power method

Algorithm 1 Graph-Truncated Power Method

input $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p},\ G = (V, E),\ \mathbf{x}_0 \in \mathbb{R}^p$

1: $i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w}_i \leftarrow \widehat{\mathbf{\Sigma}} \mathbf{x}_i$
4: $\quad \mathbf{x}_{i+1} \leftarrow \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$
5: $\quad i \leftarrow i + 1$
6: **until** Convergence/Stop Criterion

output $\mathbf{x}_i$

- Similar to truncated power method of [Yuan & Zhang, 2013].

Graph-truncated power method

Algorithm 1 Graph-Truncated Power Method

input $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p},\ G = (V, E),\ \mathbf{x}_0 \in \mathbb{R}^p$

1: $i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w}_i \leftarrow \widehat{\mathbf{\Sigma}} \mathbf{x}_i$
4: $\quad \mathbf{x}_{i+1} \leftarrow \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$
5: $\quad i \leftarrow i + 1$
6: **until** Convergence/Stop Criterion

output $\mathbf{x}_i$

- Similar to truncated power method of [Yuan & Zhang, 2013].
- Main difference with previous work: projection operation

$$\mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg \min_{\mathbf{x} \in \mathcal{X}(G)} \frac{1}{2} \|\mathbf{x} - \mathbf{w}\|_2^2$$

(defined later)

Low dim. sample and project

Algorithm 2 Low-Dimensional Sample and Project

input $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}$, $G = (V, E)$, $r \in [p]$, $\epsilon > 0$

$[\mathbf{Q}, \mathbf{\Lambda}] \leftarrow \texttt{svd}(\widehat{\mathbf{\Sigma}}, r)$

$\mathbf{V} \leftarrow \mathbf{Q}\mathbf{\Lambda}^{1/2}$ $\quad \{\widehat{\mathbf{\Sigma}}_r \triangleq \mathbf{V}\mathbf{V}^\top\}$

$\mathcal{C} \leftarrow \emptyset$ $\quad \{\text{Candidate solutions}\}$

for $i = 1 : O(\epsilon^{-r} \cdot \log p)$ **do**

  $\mathbf{c}_i \leftarrow$ uniformly sampled from $\mathbb{S}^{r-1}$

  $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{c}_i$

  $\mathbf{x}_i \leftarrow \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$

  $\mathcal{C} = \mathcal{C} \cup \{\mathbf{x}_i\}$

end for

output $\widehat{\mathbf{x}}_r \leftarrow \arg\max_{\mathbf{x} \in \mathcal{C}} \|\mathbf{V}^\top \mathbf{x}\|_2^2$

- Based on Spannogram algorithm by [Asteris et al, 2014].

Low dim. sample and project

Algorithm 2 Low-Dimensional Sample and Project

input $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}, G = (V, E), r \in [p], \epsilon > 0$

$[\mathbf{Q}, \mathbf{\Lambda}] \leftarrow \texttt{svd}(\widehat{\mathbf{\Sigma}}, r)$

$\mathbf{V} \leftarrow \mathbf{Q}\mathbf{\Lambda}^{1/2}$ $\quad \{\widehat{\mathbf{\Sigma}}_r \triangleq \mathbf{V}\mathbf{V}^\top\}$

$\mathcal{C} \leftarrow \emptyset$ $\quad \{\text{Candidate solutions}\}$

for $i = 1 : O(\epsilon^{-r} \cdot \log p)$ **do**

  $\mathbf{c}_i \leftarrow$ uniformly sampled from $\mathbb{S}^{r-1}$

  $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{c}_i$

  $\mathbf{x}_i \leftarrow \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$

  $\mathcal{C} = \mathcal{C} \cup \{\mathbf{x}_i\}$

end for

output $\widehat{\mathbf{x}}_r \leftarrow \arg\max_{\mathbf{x} \in \mathcal{C}} \|\mathbf{V}^\top \mathbf{x}\|_2^2$

- Based on Spannogram algorithm by [Asteris et al, 2014].
- Main idea: approximate solution via a low-rank approximation of $\widehat{\Sigma}$ and compute candidate solutions in $\mathcal{X}(G)$ solving easy sub-problems over an ε-net.

Low dim. sample and project

Algorithm 2 Low-Dimensional Sample and Project

input $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}$, $G = (V, E)$, $r \in [p]$, $\epsilon > 0$

$[\mathbf{Q}, \mathbf{\Lambda}] \leftarrow \texttt{svd}(\widehat{\mathbf{\Sigma}}, r)$

$\mathbf{V} \leftarrow \mathbf{Q}\mathbf{\Lambda}^{1/2}$ $\quad \{\widehat{\mathbf{\Sigma}}_r \triangleq \mathbf{V}\mathbf{V}^\top\}$

$\mathcal{C} \leftarrow \emptyset$ $\quad \{\text{Candidate solutions}\}$

for $i = 1 : O(\epsilon^{-r} \cdot \log p)$ **do**

  $\mathbf{c}_i \leftarrow$ uniformly sampled from $\mathbb{S}^{r-1}$

  $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{c}_i$

  $\mathbf{x}_i \leftarrow \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$

  $\mathcal{C} = \mathcal{C} \cup \{\mathbf{x}_i\}$

end for

output $\widehat{\mathbf{x}}_r \leftarrow \arg\max_{\mathbf{x} \in \mathcal{C}} \|\mathbf{V}^\top \mathbf{x}\|_2^2$

- Based on Spannogram algorithm by [Asteris et al, 2014].
- Main idea: approximate solution via a low-rank approximation of $\widehat{\mathbf{\Sigma}}$ and compute candidate solutions in $\mathcal{X}(G)$ solving easy sub-problems over an ε-net.
- Main difference with previous work: projection operation

$$\text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg \min_{\mathbf{x} \in \mathcal{X}(G)} \frac{1}{2}\|\mathbf{x} - \mathbf{w}\|_2^2$$ *(defined later)*

Projection operator

Projection operator

$$\mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathcal{X}(G)} \frac{1}{2}\|\mathbf{x}-\mathbf{w}\|_2^2 \quad \Leftrightarrow \quad \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\max_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{w}^\top \mathbf{x}$$

Projection operator

$$\text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathcal{X}(G)} \frac{1}{2}\|\mathbf{x}-\mathbf{w}\|_2^2 \quad \Leftrightarrow \quad \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\max_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{w}^\top \mathbf{x}$$

- By Cauchy-Schwarz:

$$\mathbf{w}^\top \mathbf{x} = \sum_{i\in\pi} w_i x_i \leq \sum_{i\in\pi} w_i^2 = \widehat{\mathbf{w}}^\top \mathbf{1}_\pi$$

where: $\pi \in \mathcal{P}(G)$, $\widehat{w}_i = w_i^2, \forall i \in [p]$ and, $\mathbf{1}_\pi \in \{0,1\}^p$ denotes the characteristic of π. For fixed π, equality is achieved for:

$$\mathbf{x}_\pi = \mathbf{w}_\pi / \|\mathbf{w}_\pi\|_2 \qquad \text{and} \qquad \mathbf{x}_{\pi^c} = 0$$

Projection operator

$$\mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathcal{X}(G)} \frac{1}{2}\|\mathbf{x}-\mathbf{w}\|_2^2 \quad \Leftrightarrow \quad \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\max_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{w}^\top \mathbf{x}$$

- By Cauchy-Schwarz:

$$\mathbf{w}^\top \mathbf{x} = \sum_{i\in\pi} w_i x_i \leq \sum_{i\in\pi} w_i^2 = \widehat{\mathbf{w}}^\top \mathbf{1}_\pi$$

where: $\pi \in \mathcal{P}(G)$, $\widehat{w}_i = w_i^2, \forall i \in [p]$ and, $\mathbf{1}_\pi \in \{0,1\}^p$ denotes the characteristic of π. For fixed π, equality is achieved for:

$$\mathbf{x}_\pi = \mathbf{w}_\pi / \|\mathbf{w}_\pi\|_2 \qquad \text{and} \qquad \mathbf{x}_{\pi^c} = 0$$

- "Equivalent" problem: $\pi(\mathbf{w}) \in \arg\max_{\pi\in\mathcal{P}(G)} \widehat{\mathbf{w}}^\top \mathbf{1}_\pi$

i.e., **solving the longest (weighted) path problem.**

Projection operator

$$\text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathcal{X}(G)} \frac{1}{2}\|\mathbf{x}-\mathbf{w}\|_2^2 \quad \Leftrightarrow \quad \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \in \arg\max_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{w}^\top \mathbf{x}$$

- By Cauchy-Schwarz:

$$\mathbf{w}^\top \mathbf{x} = \sum_{i\in\pi} w_i x_i \leq \sum_{i\in\pi} w_i^2 = \widehat{\mathbf{w}}^\top \mathbf{1}_\pi$$

where: $\pi \in \mathcal{P}(G)$, $\widehat{w}_i = w_i^2, \forall i \in [p]$ and, $\mathbf{1}_\pi \in \{0,1\}^p$ denotes the characteristic of π. For fixed π, equality is achieved for:

$$\mathbf{x}_\pi = \mathbf{w}_\pi / \|\mathbf{w}_\pi\|_2 \qquad \text{and} \qquad \mathbf{x}_{\pi^c} = 0$$

- "Equivalent" problem: $\pi(\mathbf{w}) \in \arg\max_{\pi\in\mathcal{P}(G)} \widehat{\mathbf{w}}^\top \mathbf{1}_\pi$

i.e., **solving the longest (weighted) path problem.**

- For DAGs, this problem can be solved in $O(p+|E|)$ time, *i.e.*, linear in the size of the graph.

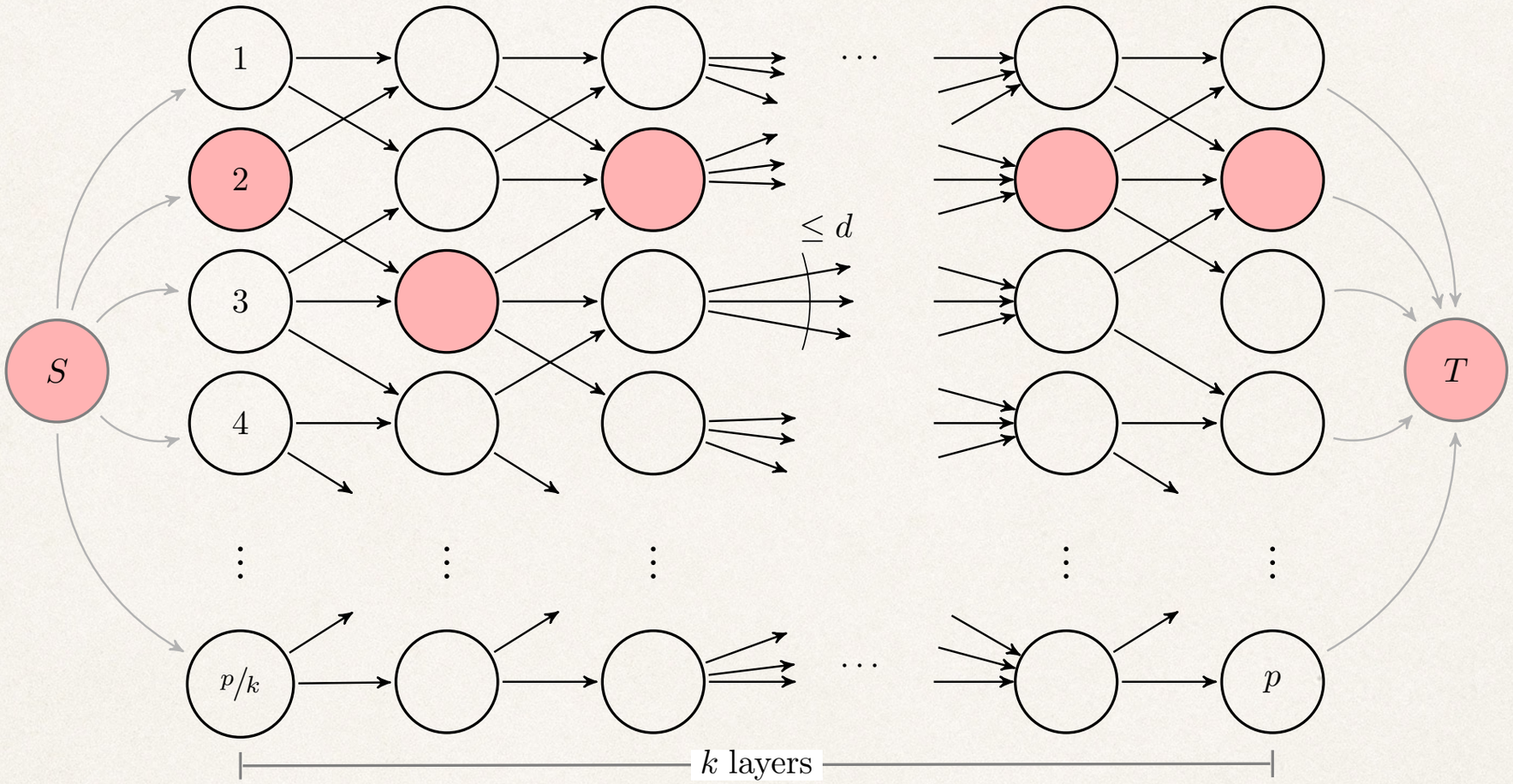
Θεωρία - Theory

(why it works?)

Data model and assumptions

- The layer graph:

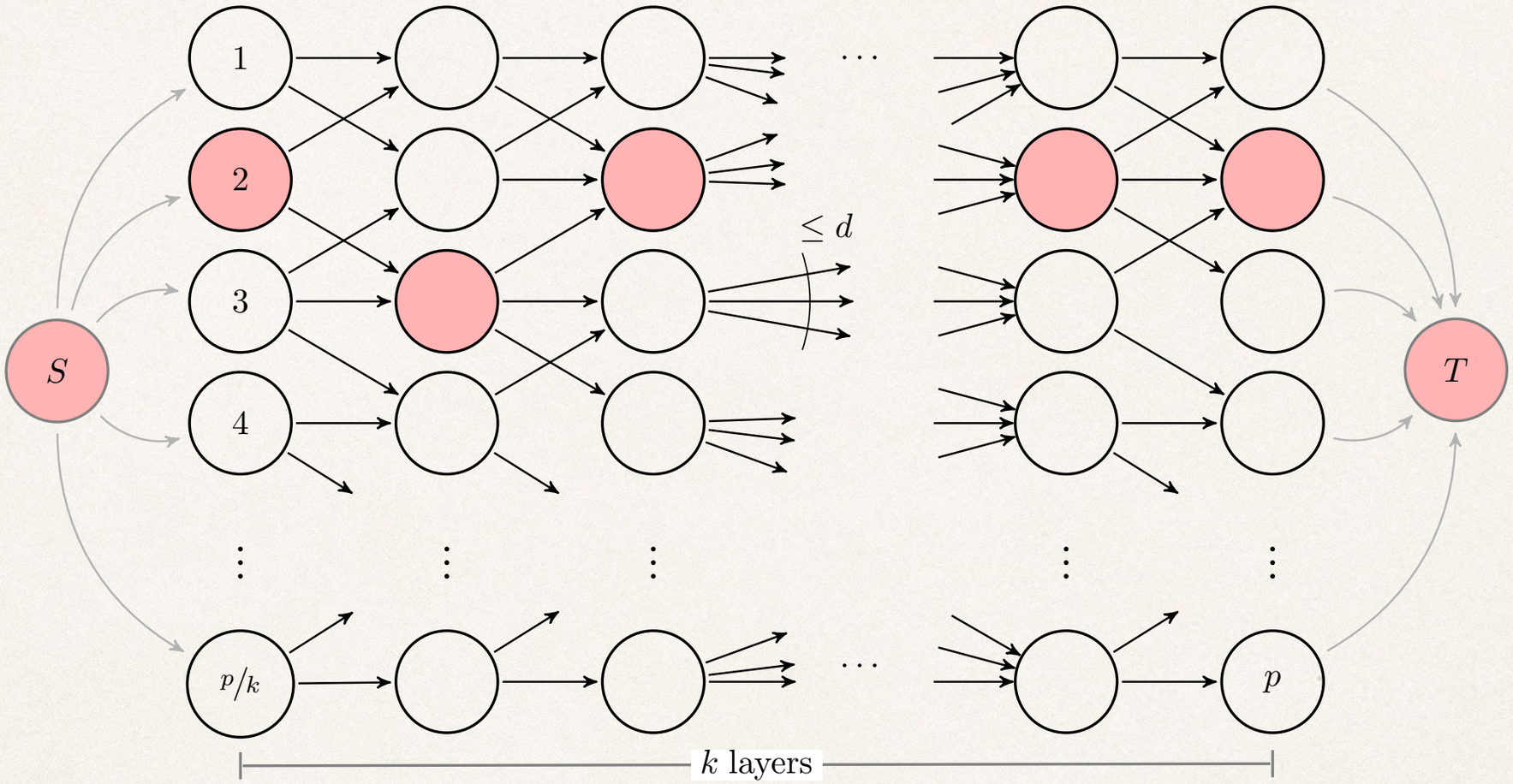
$\mathcal{X}(G)$



Data model and assumptions

- The layer graph:

$\mathcal{X}(G)$



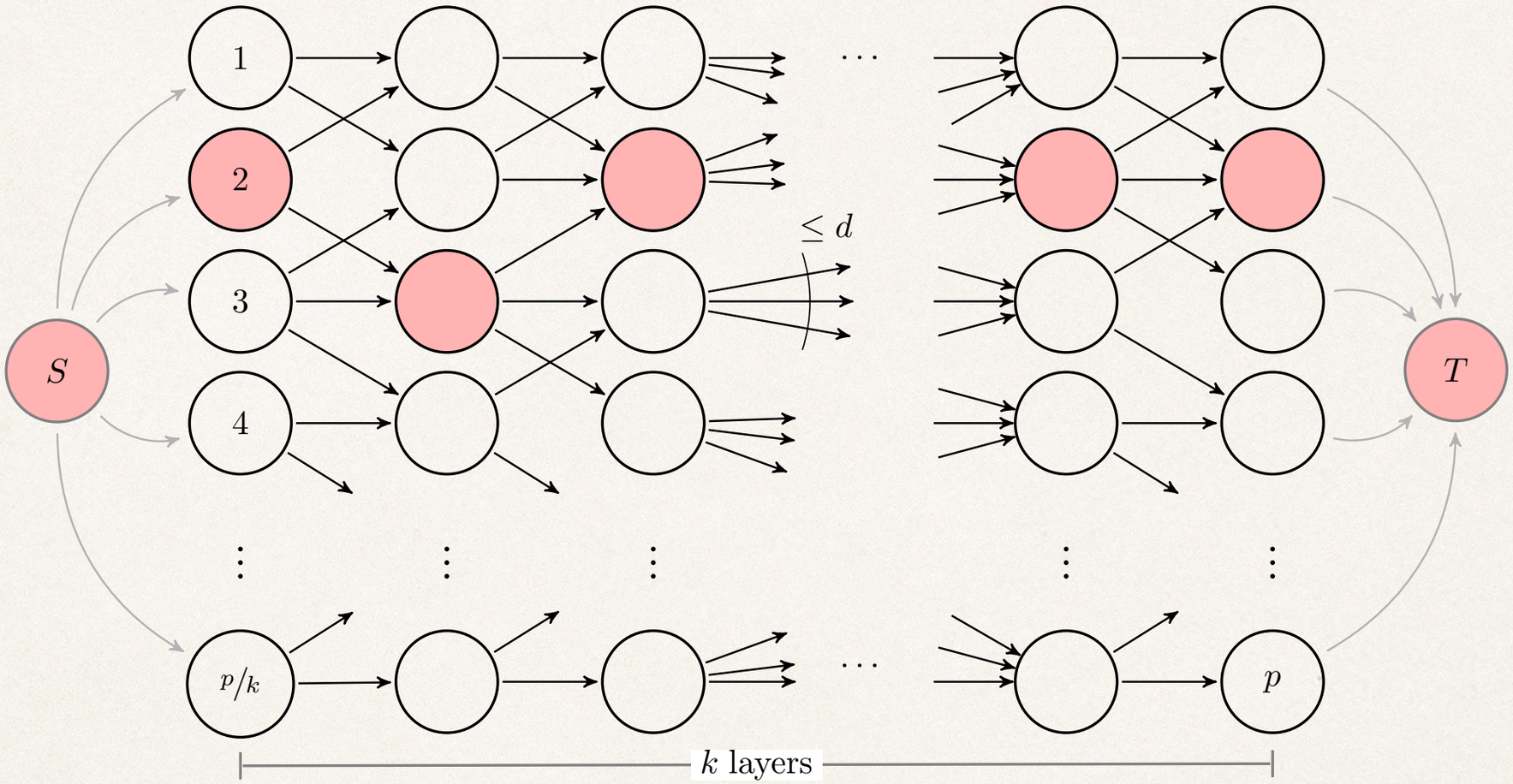
- Number of paths:

$$|\mathcal{P}(G)| = \frac{p-2}{k} \cdot d^{k-1} \leq \binom{p-2}{k}$$

Data model and assumptions

- The layer graph:

$\mathcal{X}(G)$



- Number of paths:

$$|\mathcal{P}(G)| = \frac{p-2}{k} \cdot d^{k-1} \leq \binom{p-2}{k} \qquad \beta > 0$$

- Spike along a path: $\boldsymbol{\Sigma} = \mathbf{I}_p + \beta \cdot \mathbf{x}_\star \mathbf{x}_\star^\top$ and draw samples $\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

Lower bound

Theorem 1 (Lower Bound). *Consider a (p, k, d)-layer graph G on p vertices, with $k \geq 4$, and $\log d \geq 4H(3/4)$. Let $\{\mathbf{y}_i\}_{i=1}^n$ be a sequence of n random observations, independently drawn according to probability density function*

$$\mathcal{D}_p(\mathbf{x}_\star) = \mathcal{N}\big(\mathbf{0}, \mathbf{I}_p + \beta \cdot \mathbf{x}_\star \mathbf{x}_\star^\top\big),$$

for some $\beta > 0$. There exists $\mathbf{x}_\star \in \mathcal{X}(G)$ such that for every estimator $\widehat{\mathbf{x}}$,

$$\mathbb{E}_{\mathcal{D}_p^{(n)}(\mathbf{x}_\star)}\left[\|\widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top\|_{\mathrm{F}}\right] \geq \frac{1}{2\sqrt{2}} \cdot \sqrt{\min\left\{1,\ \frac{C' \cdot (1+\beta)}{\beta^2} \cdot \frac{1}{n}\left(\log \frac{p-2}{k} + \frac{k}{4}\log d\right)\right\}}.$$

- *I.e.,* the minimax error is bounded away from zero, unless:

$$n = \Omega\left(\log \frac{p}{k} + k \log d\right) \qquad \left(\text{vs. } \Omega\left(k \log \frac{p}{k}\right)\right)$$

- Proof based on a **non-trivial local packing set construction** in Fano inequality.

Upper bound

- Upper bound is based on the estimator:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^p}{\text{maximize}} \quad & \mathbf{x}^\top \widehat{\mathbf{\Sigma}} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}(G). \end{aligned}$$

Theorem 2 (Upper bound). *Consider a (p,k,d)-layer graph G and $\mathbf{x}_\star \in \mathcal{X}(G)$. Let $\{\mathbf{y}_i\}_{i=1}^n$ be a sequence of n i.i.d. $\mathcal{N}(\mathbf{0},\mathbf{\Sigma})$ samples, where $\mathbf{\Sigma} \succeq \mathbf{0}$ with eigenvalues $\lambda_1 > \lambda_2 \geq \ldots$, and principal eigenvector $\mathbf{x}_\star$. Let $\widehat{\mathbf{\Sigma}}$ be the empirical covariance of the n samples, $\widehat{\mathbf{x}}$ the estimate of $\mathbf{x}_\star$ obtained via* (3)*, and $\epsilon \triangleq \|\widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star\mathbf{x}_\star^\top\|_{\mathrm{F}}$. Then,*

$$\mathbb{E}[\epsilon] \leq C \cdot \frac{\lambda_1}{\lambda_1 - \lambda_2} \cdot \frac{1}{n} \cdot \max\left\{\sqrt{nA},\, A\right\},$$

where $A = O\big(\log \frac{p-2}{k} + k \log d\big)$.

- Analysis not restricted to the spiked covariance model.
- Thus:

$$n = \Theta\left(\log \frac{p}{k} + k \log d\right)$$

Conclusions (to be updated)

This paper:

- Introduces new Sparse PCA formulation: set of feasible supports determined by paths on DAGs.
- Proposes two efficient algorithmic solutions for the case of longest weighted paths.
- Provides lower and upper bounds for the case of layer graph model.

Conclusions (to be updated)

This paper:

- Introduces new Sparse PCA formulation: set of feasible supports determined by paths on DAGs.
- Proposes two efficient algorithmic solutions for the case of longest weighted paths.
- Provides lower and upper bounds for the case of layer graph model.

Future work:

- Other graph models, beyond longest paths?
- Practical upper bounds, “connected” with algorithms proposed.
- Further investigation for more applications.