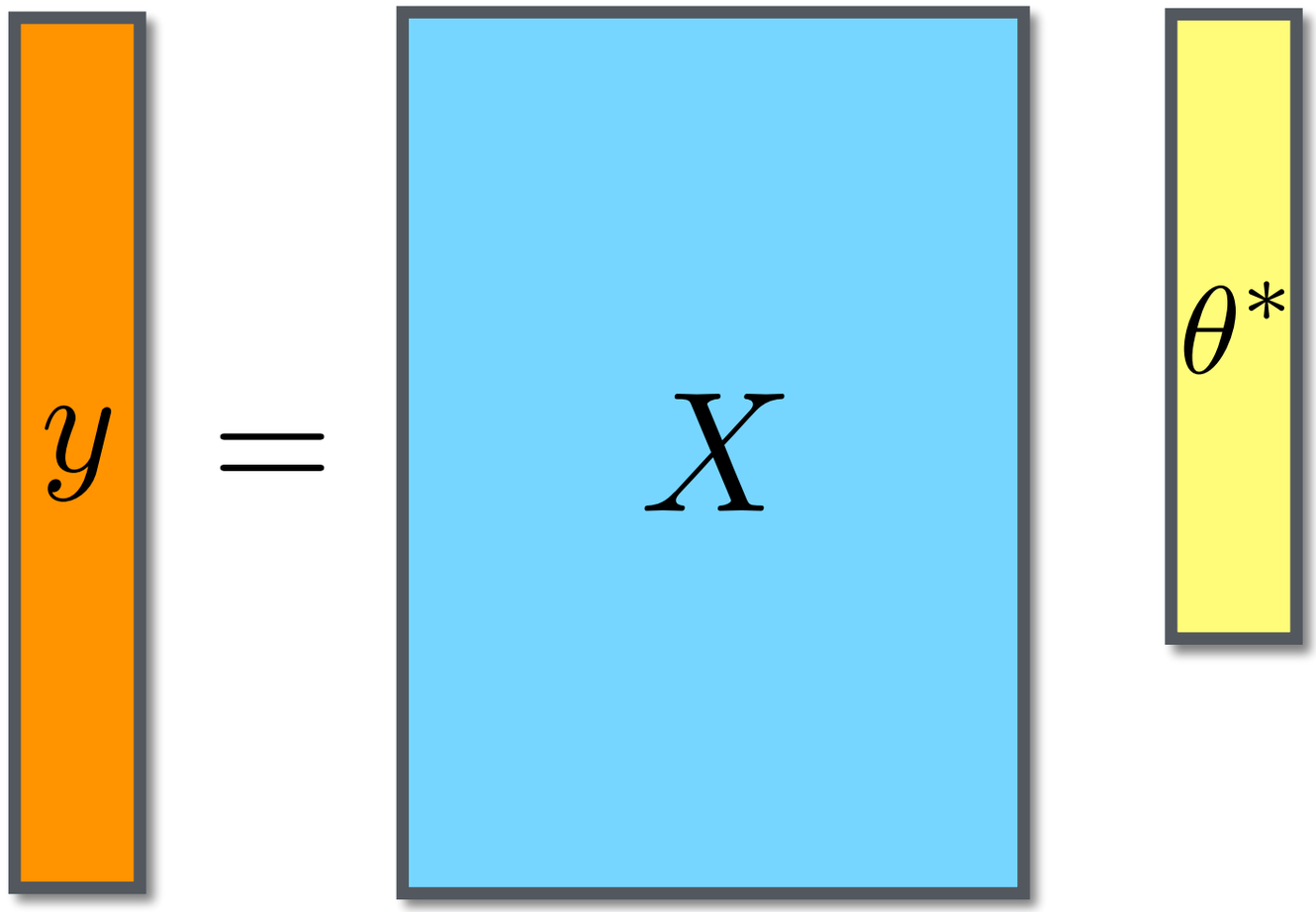
Linear Regression with Graph Constraints

Ludwig Schmidt
MIT → UC Berkeley

Based on joint works with Tobias Ehrenberger, Ernest Fraenkel, Chinmay Hegde, Piotr Indyk, Stefanie Jegelka, Alexander LeNail, Jonathan Li, and Karen Sachs.

Linear Regression



Given: data matrix X and observations y.

Goal: find the unknown θ*.

(also: predict y in the future)

Linear Regression

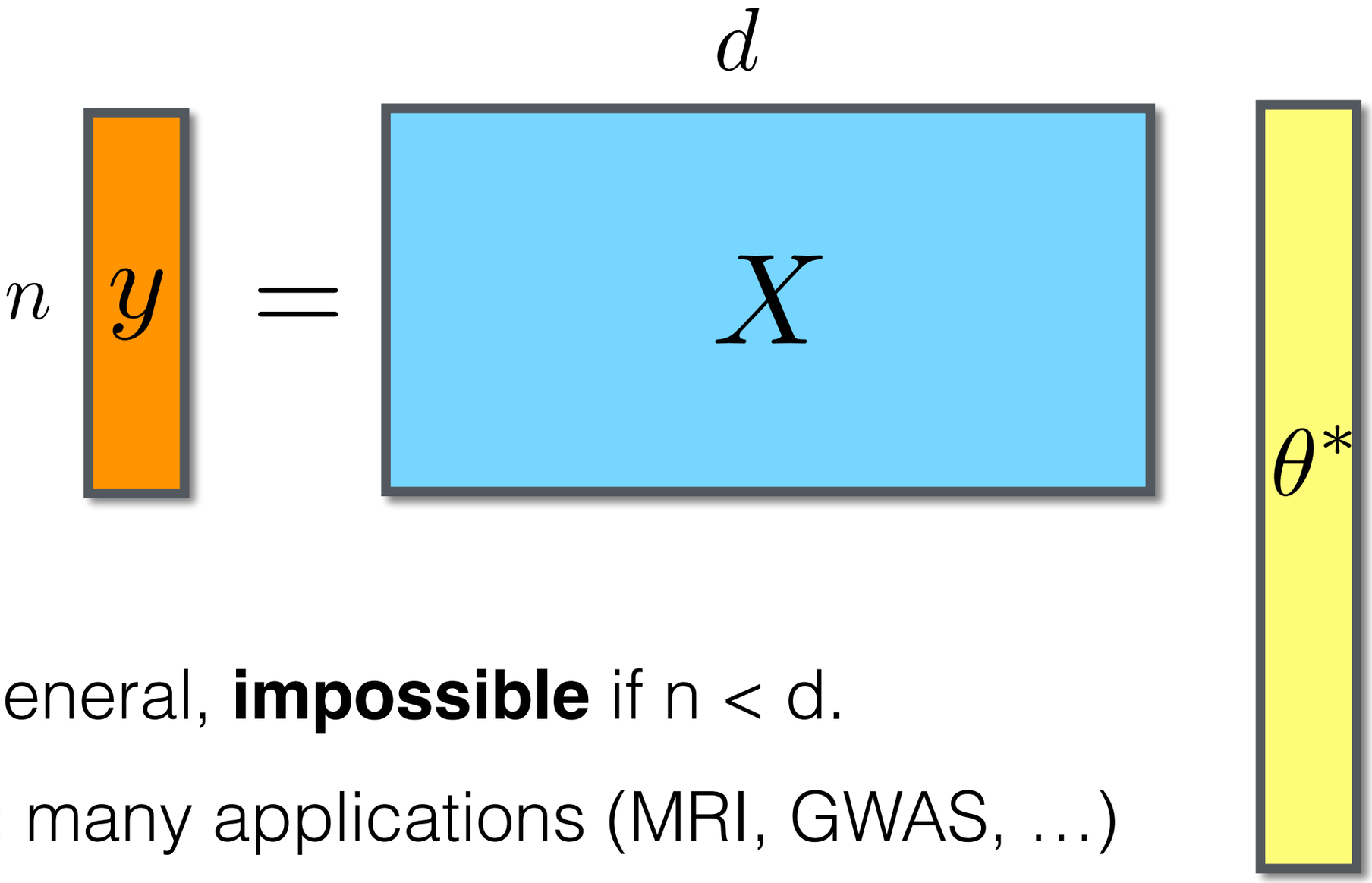
$$y = X \quad \theta^* \quad + \text{noise}$$

Given: data matrix X and observations y.

Goal: find the unknown θ*.
(also: predict y in the future)

Gauss, early 1800s

Constrained Linear Regression



In general, **impossible** if n < d.

But: many applications (MRI, GWAS, …)

Constrained Linear Regression

$$n \begin{bmatrix} y \end{bmatrix} = \overset{d}{\begin{bmatrix} X \end{bmatrix}} \begin{bmatrix} \theta^* \end{bmatrix}$$

In general, **impossible** if n < d.

But: many applications (MRI, GWAS, …)

Constrained Linear Regression

$$n \begin{bmatrix} y \end{bmatrix} = \overset{d}{\begin{bmatrix} X \end{bmatrix}} \begin{bmatrix} \theta^* \end{bmatrix}$$

In general, **impossible** if n < d.

But: many applications (MRI, GWAS, …)

→ Use extra structure such as **sparsity**.

Solving Linear Regression with Constraints

	Standard Linear Regression	Sparse Linear Regression
Algorithm	$\arg\min_{\theta} \|y - X\theta\|_2^2$	
	Least Squares	

Solving Linear Regression with Constraints

	Standard Linear Regression	Sparse Linear Regression
Algorithm	$\arg\min_{\theta} \|y - X\theta\|_2^2$	$\arg\min_{\|\theta\|_0 \leq \lambda} \|y - X\theta\|_2^2$
	Least Squares	

Solving Linear Regression with Constraints

	Standard Linear Regression	Sparse Linear Regression
Algorithm	$\arg\min_{\theta} \|y - X\theta\|_2^2$	$\arg\min \|y - X\theta\|_2^2$ $\|\theta\|_1 \leq \lambda$
	Least Squares	Lasso

Convex relaxation $\ell_0 \rightarrow \ell_1$

Solving Linear Regression with Constraints

	Standard Linear Regression	Sparse Linear Regression
Algorithm	$\arg\min_{\theta} \|y - X\theta\|_2^2$ Least Squares	$\arg\min_{\|\theta\|_1 \leq \lambda} \|y - X\theta\|_2^2$ Lasso Convex relaxation $\ell_0 \rightarrow \ell_1$
Sample Complexity	$n = O(d)$	$n = O(s \log d)$

(d is the ambient dimension, s is the sparsity of the unknown vector)

Widely Used Methods

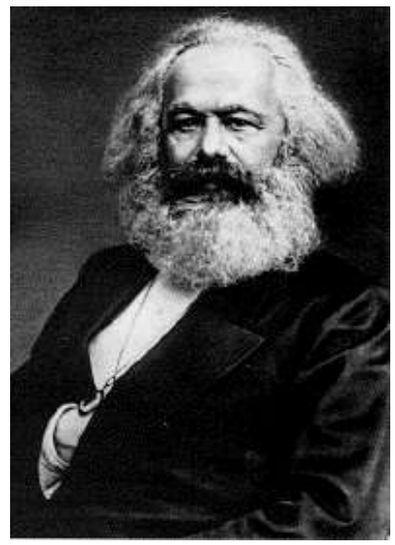
Rob Tibshirani

Popularized the Lasso

Widely Used Methods

Rob Tibshirani

Popularized the Lasso



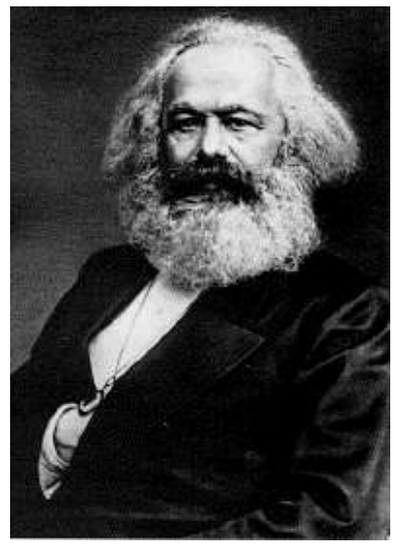
Karl Marx

Popularized Communism

Widely Used Methods

Rob Tibshirani

Popularized the Lasso



Karl Marx

Popularized Communism

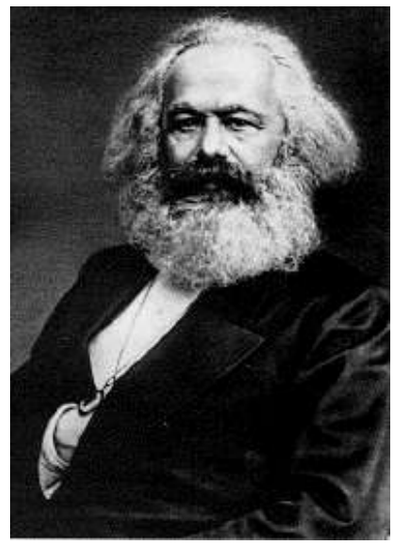
What do they have in common?

Widely Used Methods

Rob Tibshirani

Popularized the Lasso

Google Scholar:
266,534 citations

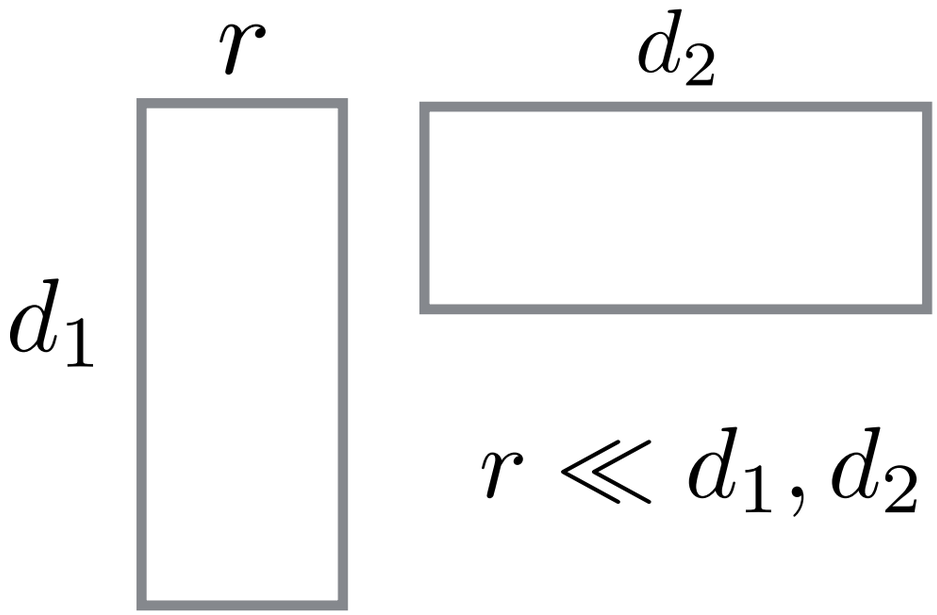


Karl Marx

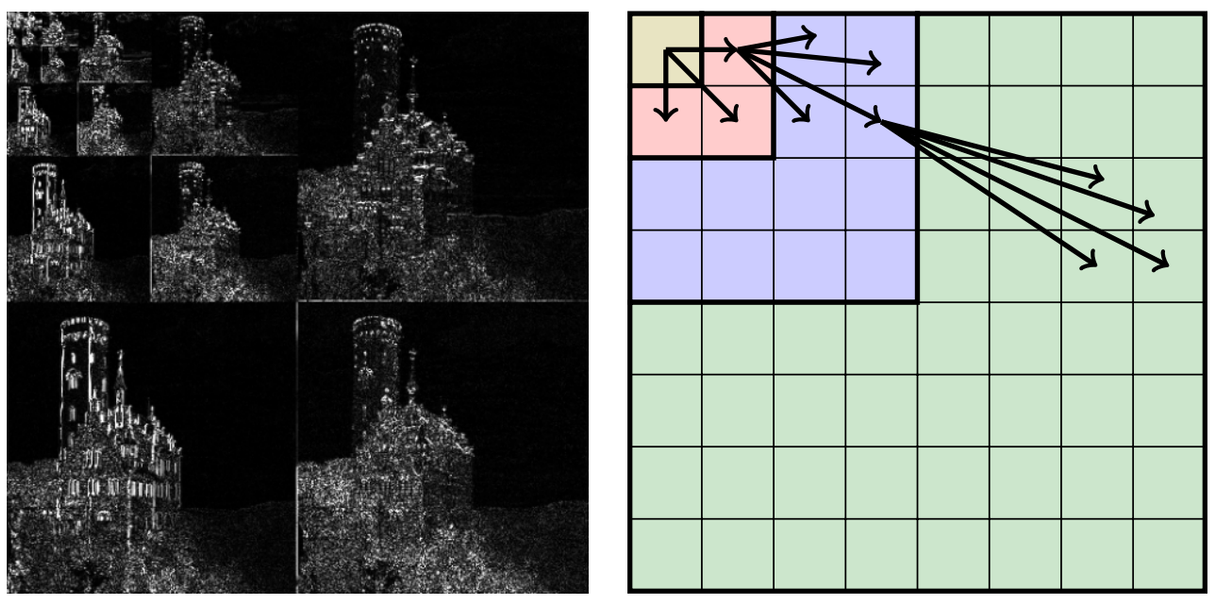
Popularized Communism

Google Scholar:
275,099 citations

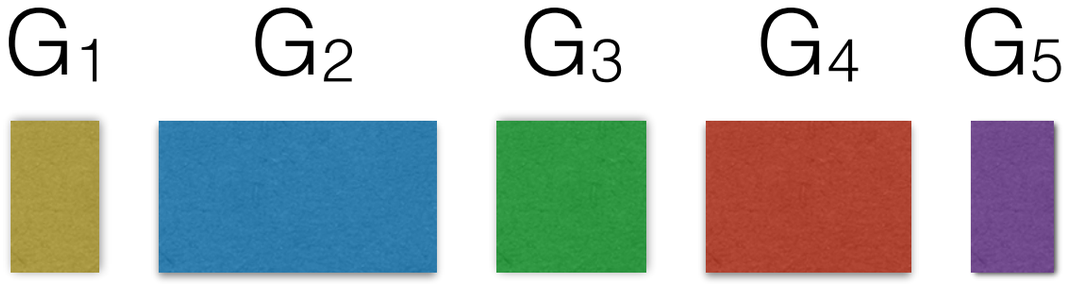
Wide Range of Constraints



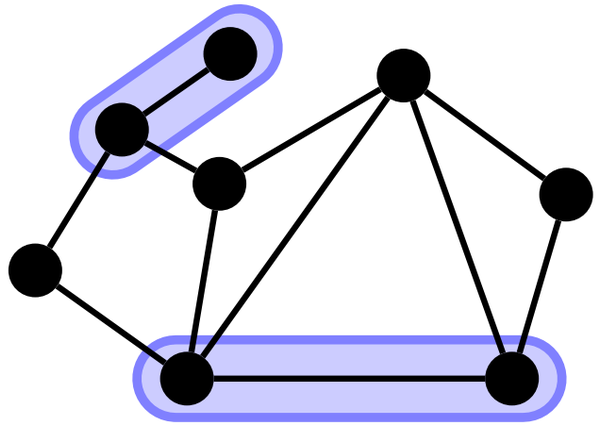
Low rank



Tree sparsity

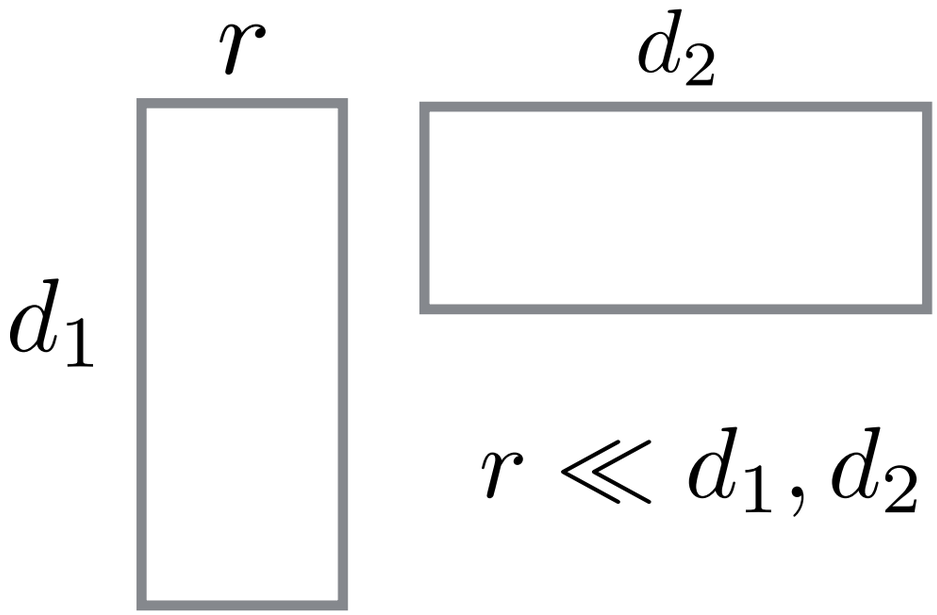


Group sparsity

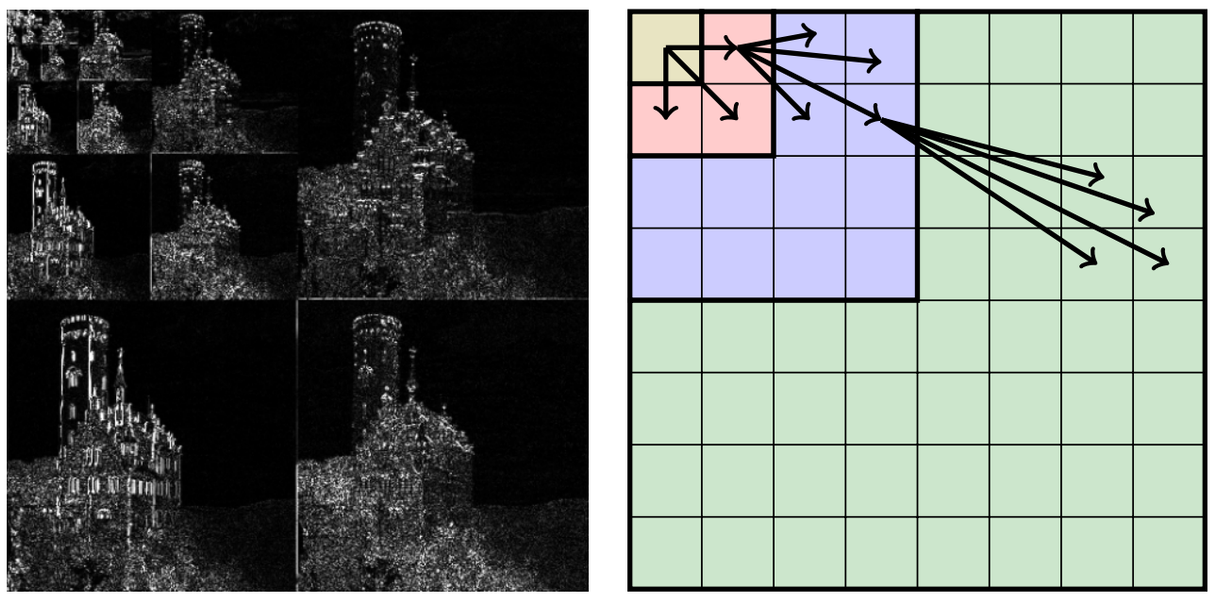


Graph sparsity

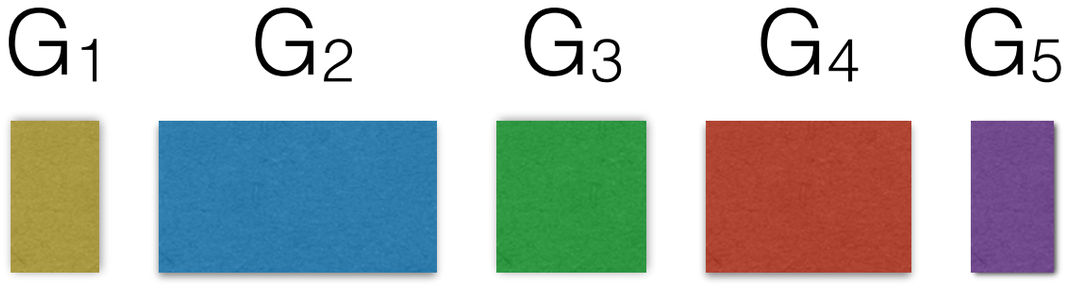
Wide Range of Constraints



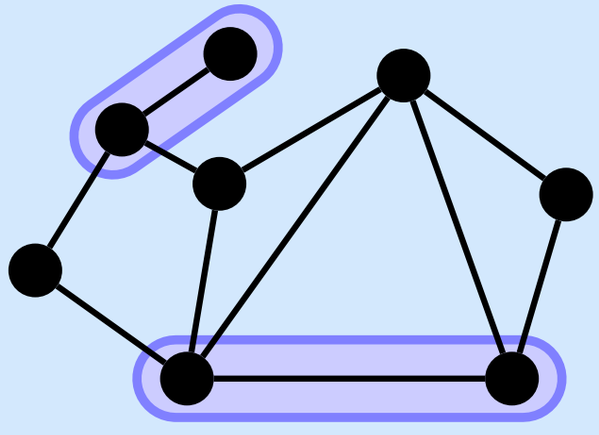
Low rank



Tree sparsity

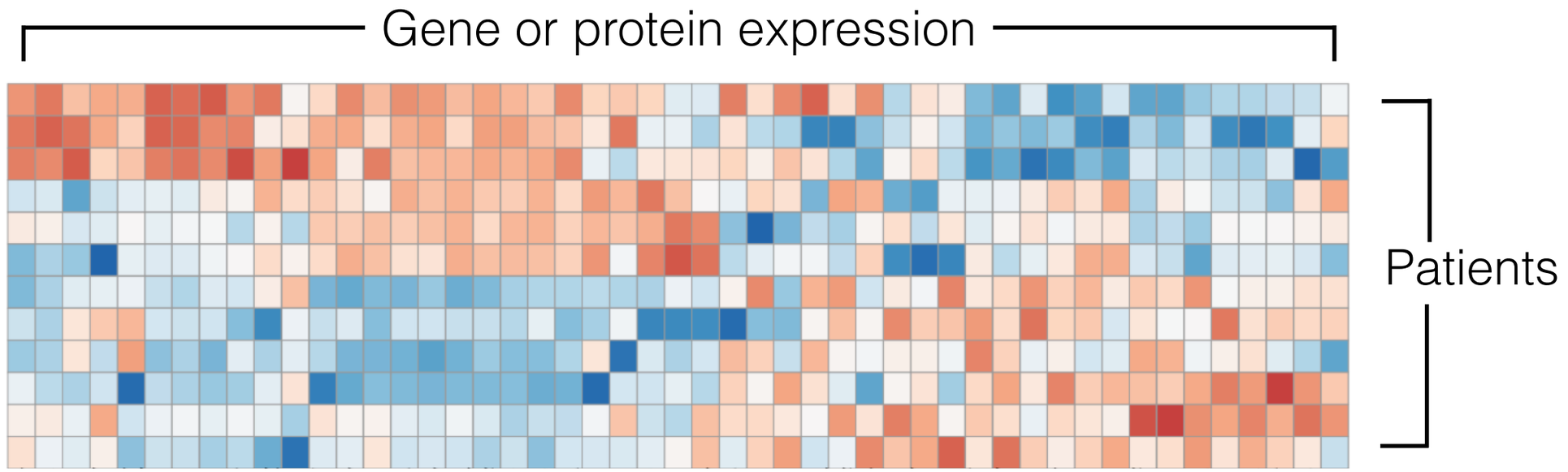


Group sparsity

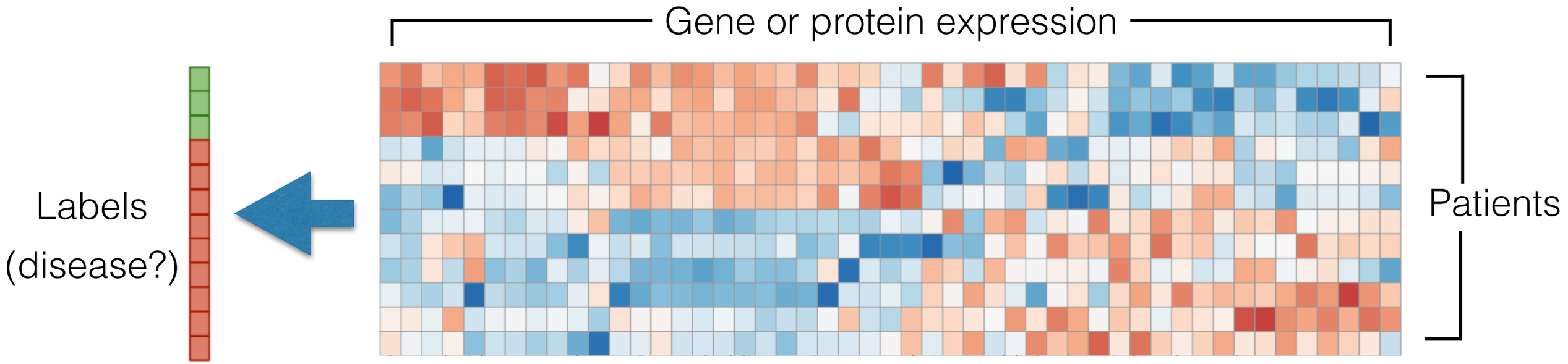


Graph sparsity

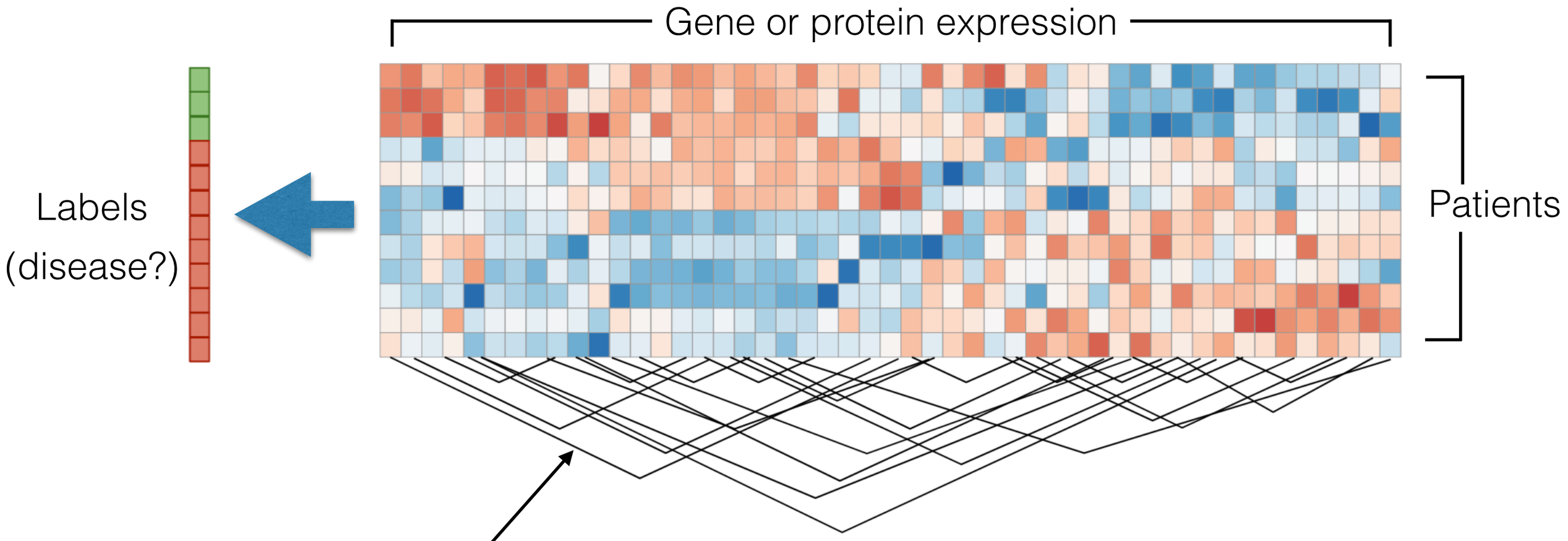
Example: Biological Network Data



Example: Biological Network Data

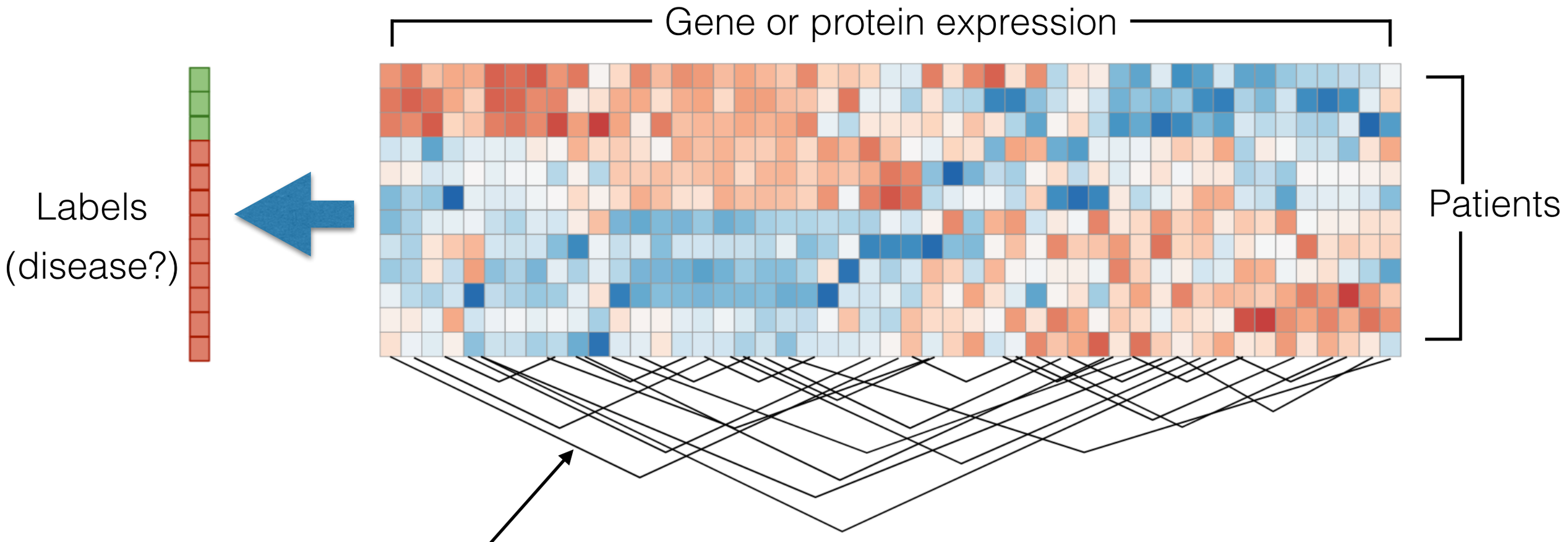


Example: Biological Network Data



Some features
(genes / proteins)
are related.

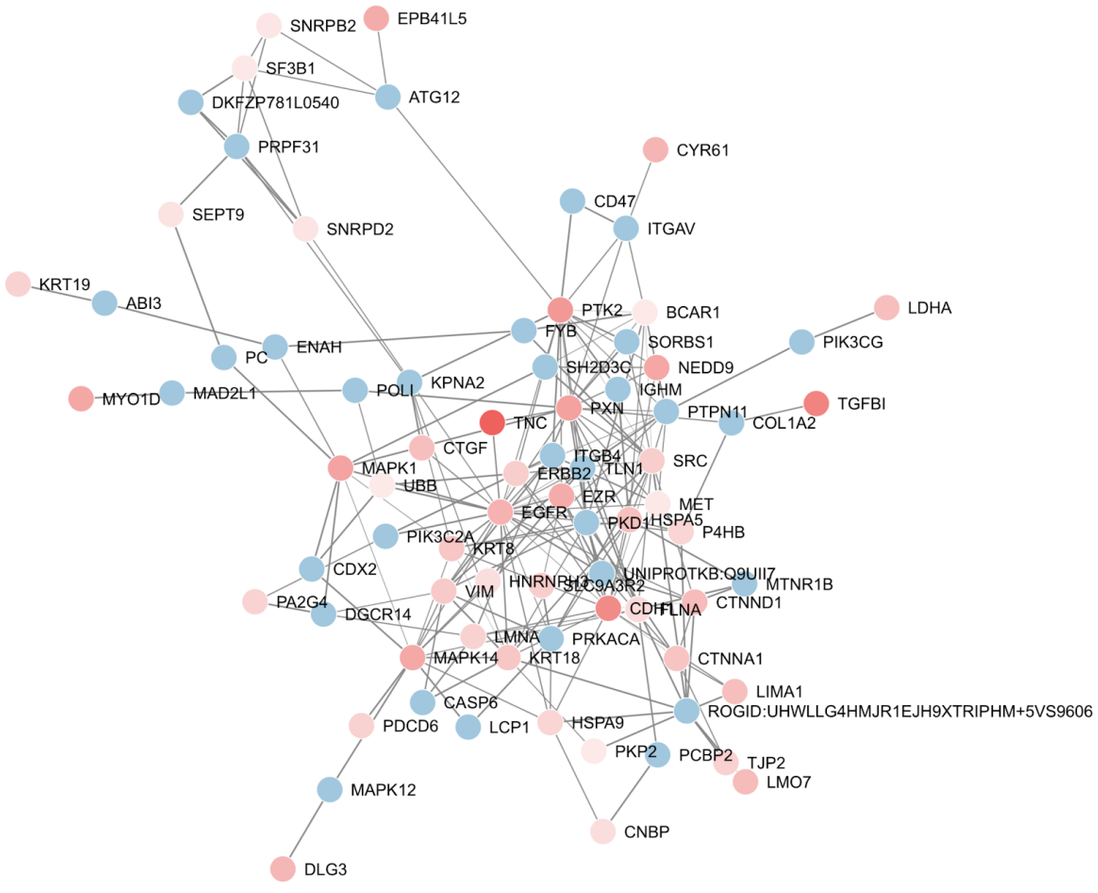
Example: Biological Network Data



Goal: find the most relevant genes / proteins.

Graph Structure

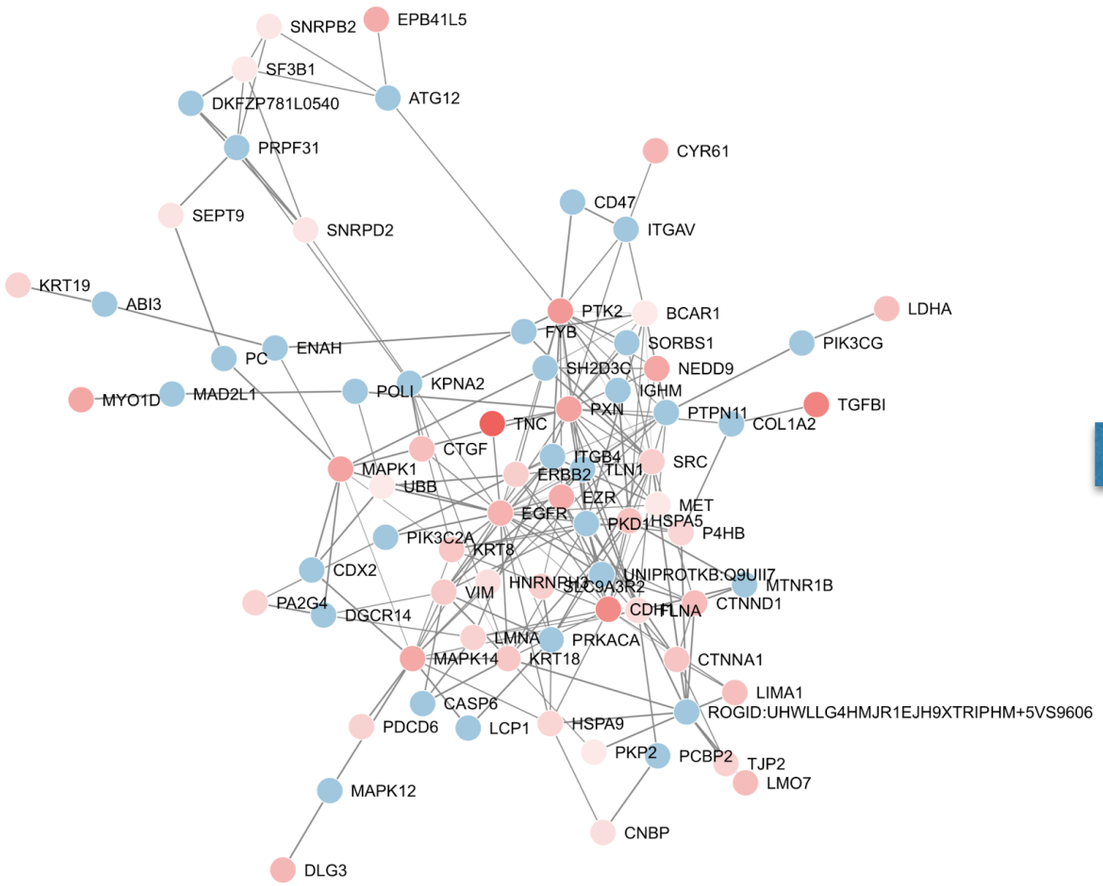
Input



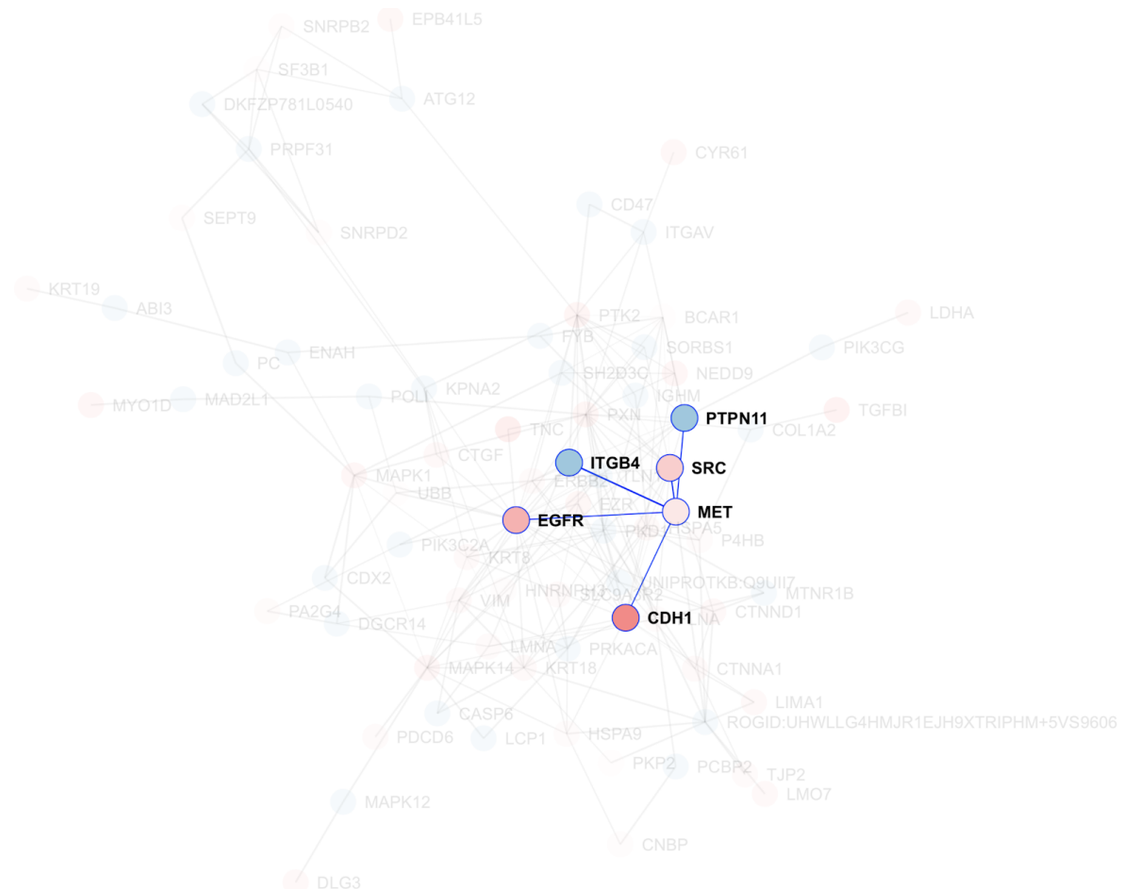
Every node corresponds to a feature (gene / protein).

Graph Structure

Input



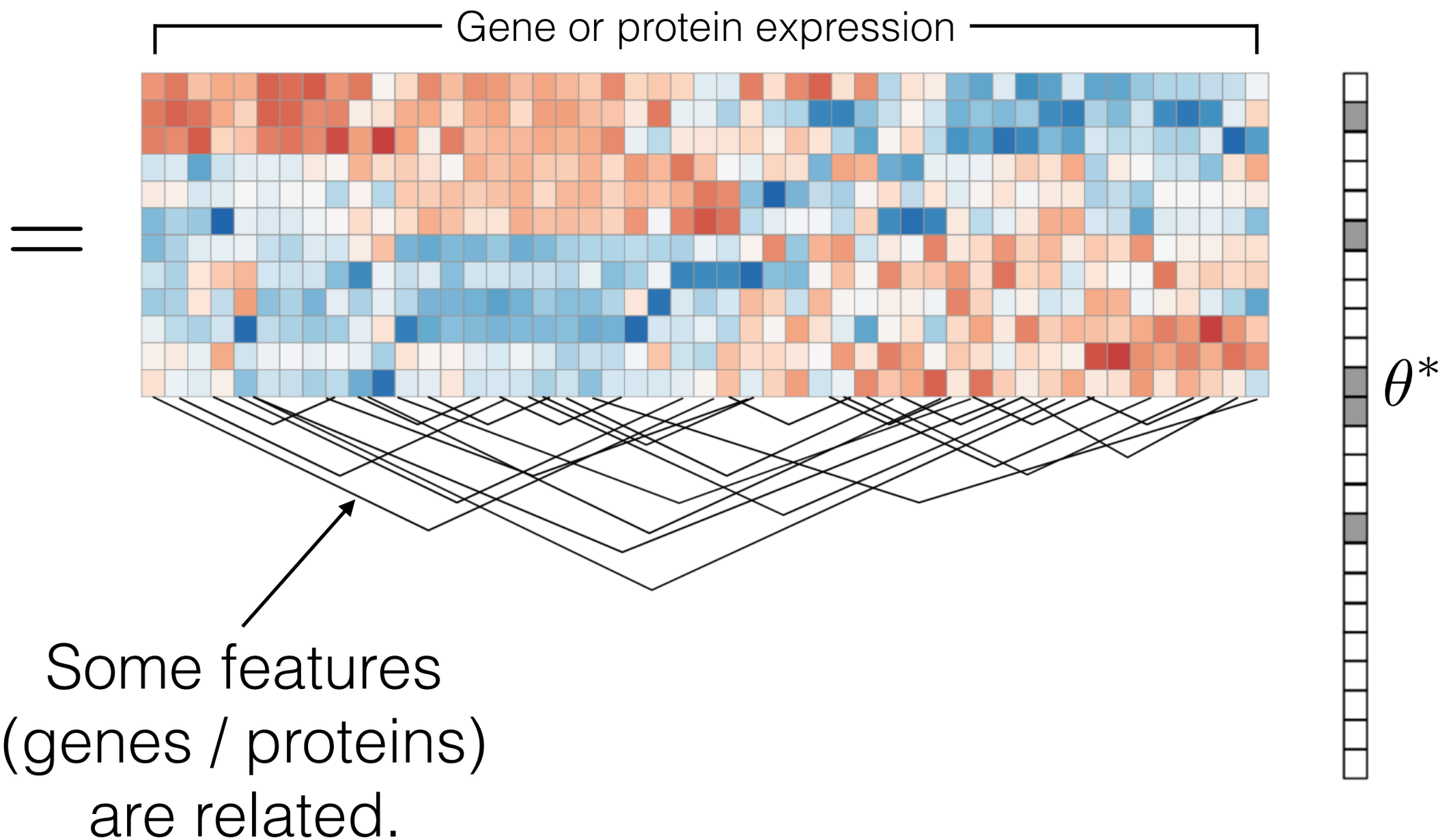
Output



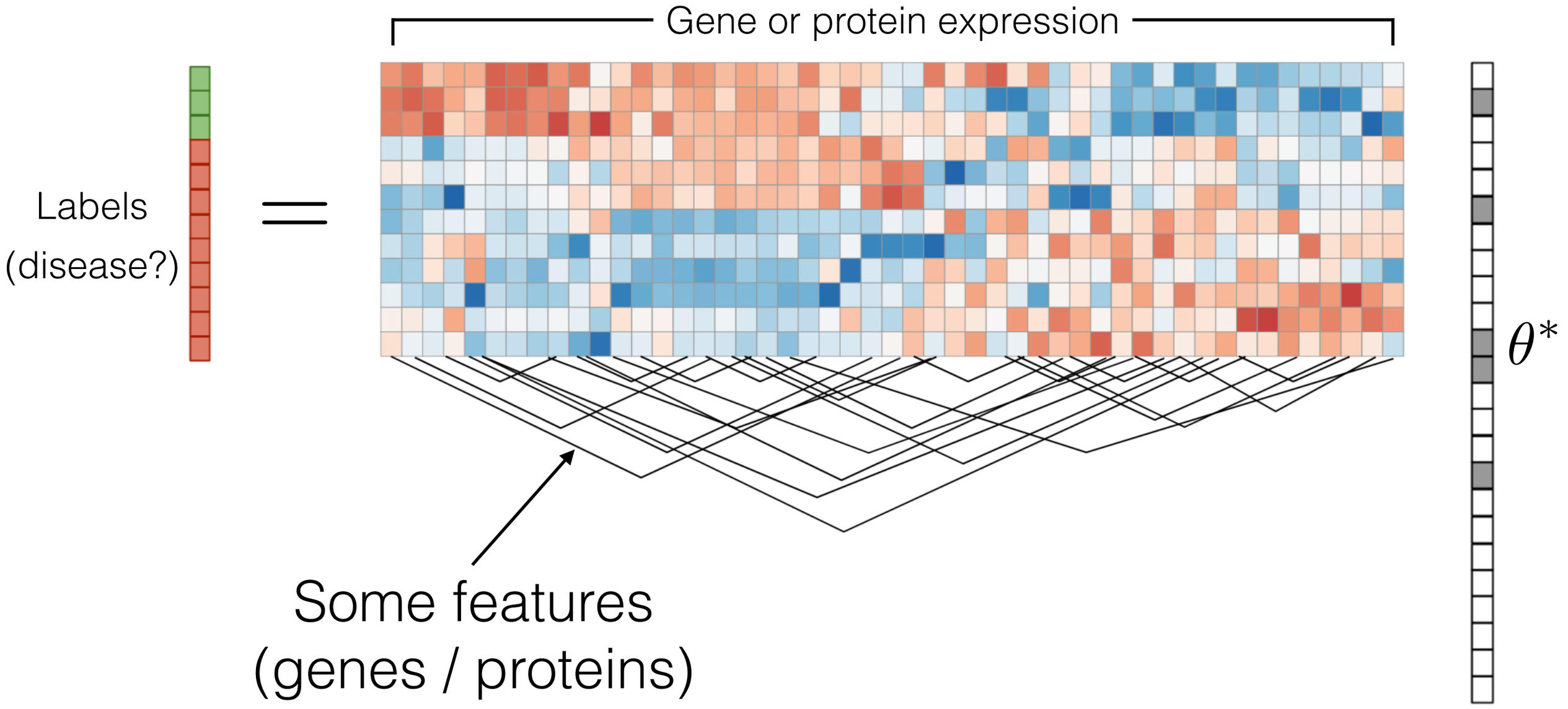
Every node corresponds to a feature (gene / protein).

Goal: Find one (or a few) **connected subgraph** with highly predictive features.

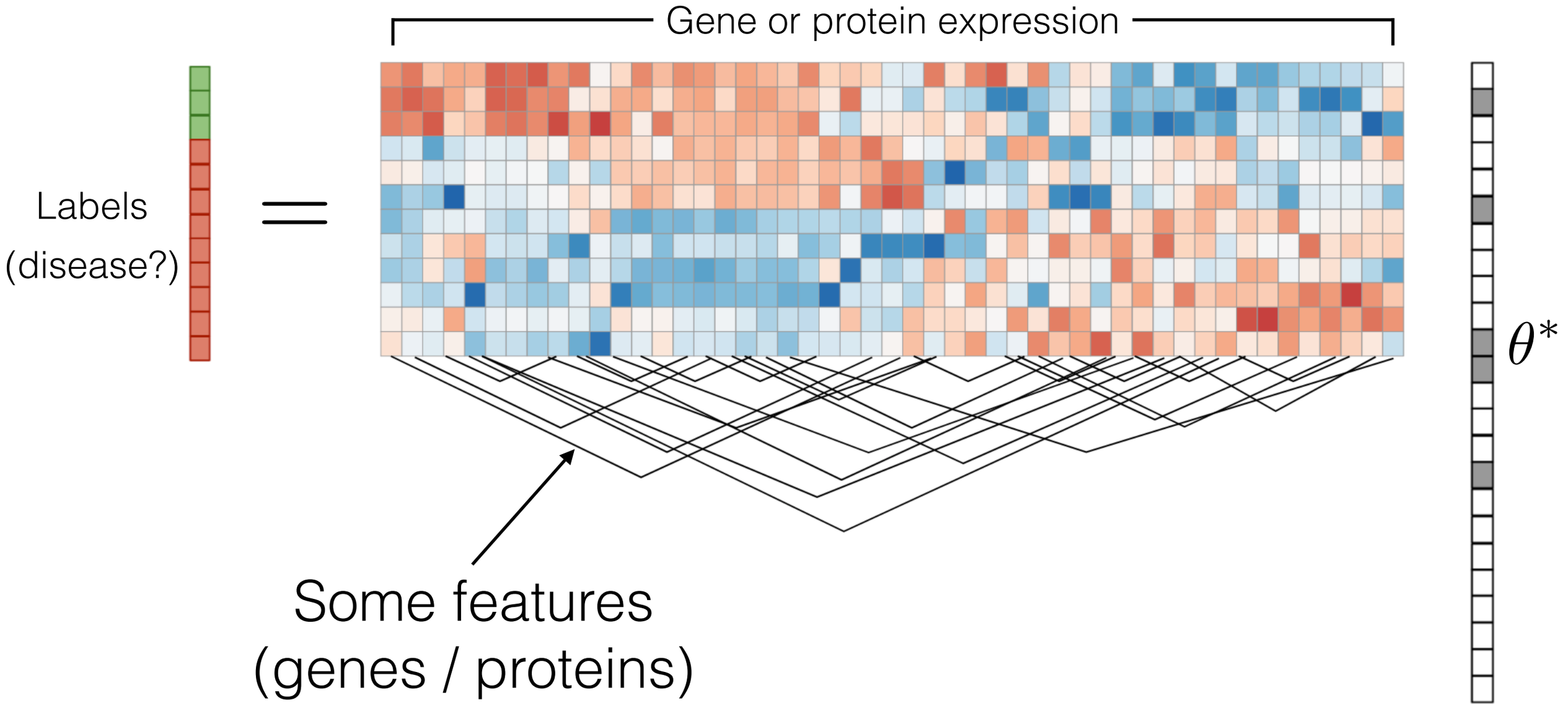
Linear Regression with Graph Constraint



Linear Regression with Graph Constraint



Linear Regression with Graph Constraint



Goal: find a **graph**-sparse set of parameters θ*.

Solving Linear Regression with Constraints

	Sparse Linear Regression	Graph-Sparse Linear Regression
Algorithm	$\arg\min_{\|\theta\|_1 \leq \lambda} \|y - X\theta\|_2^2$	$\arg\min_{\theta \in \mathcal{C}} \|y - X\theta\|_2^2$

The set C is the set of graph-sparse vectors,
i.e., vectors with a sparsity pattern
corresponding to connected subgraphs.

Solving Linear Regression with Constraints

	Sparse Linear Regression	Graph-Sparse Linear Regression
Algorithm	$\arg\min_{\|\theta\|_1 \leq \lambda} \|y - X\theta\|_2^2$	$\arg\min_{\theta \in \mathcal{C}} \|y - X\theta\|_2^2$

The set $\mathcal{C}$ is the set of graph-sparse vectors,
i.e., vectors with a sparsity pattern
corresponding to connected subgraphs.

How can we solve this problem?

Projected Gradient Descent

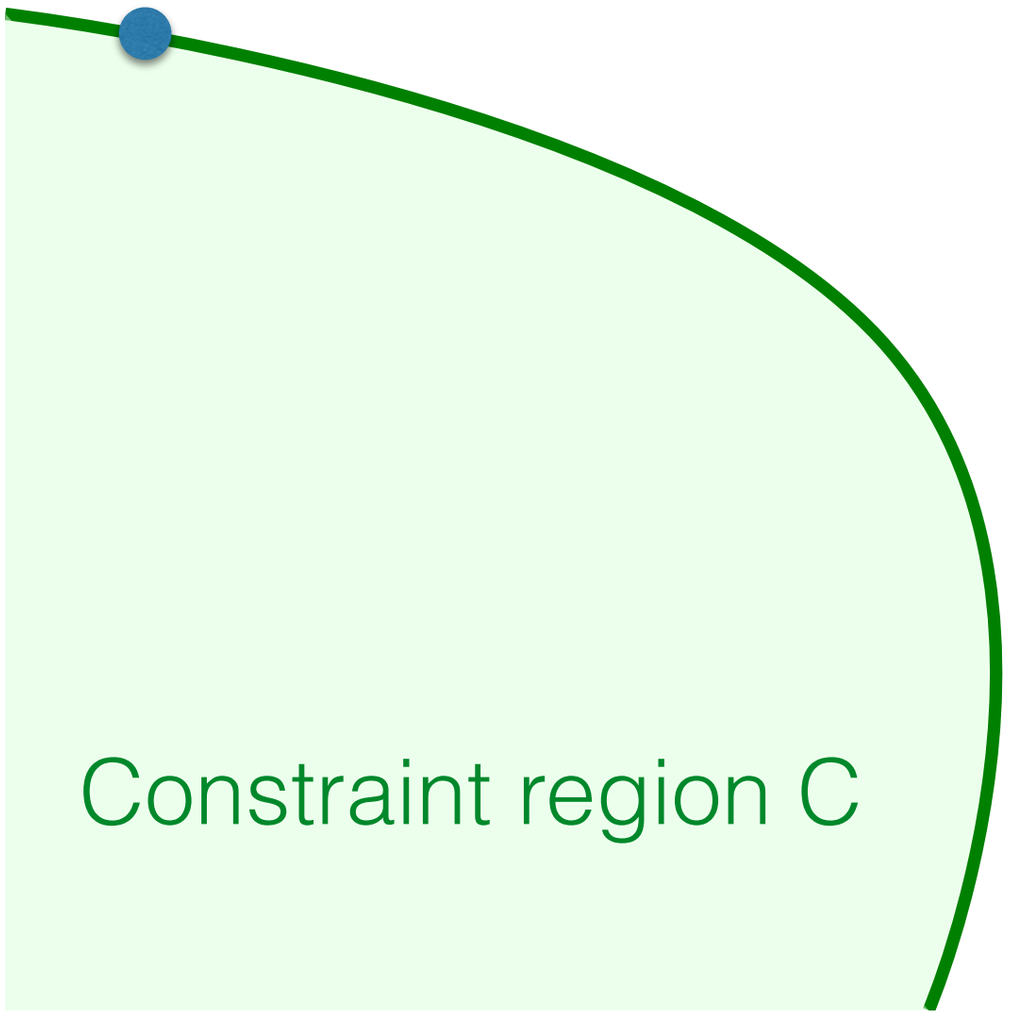
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

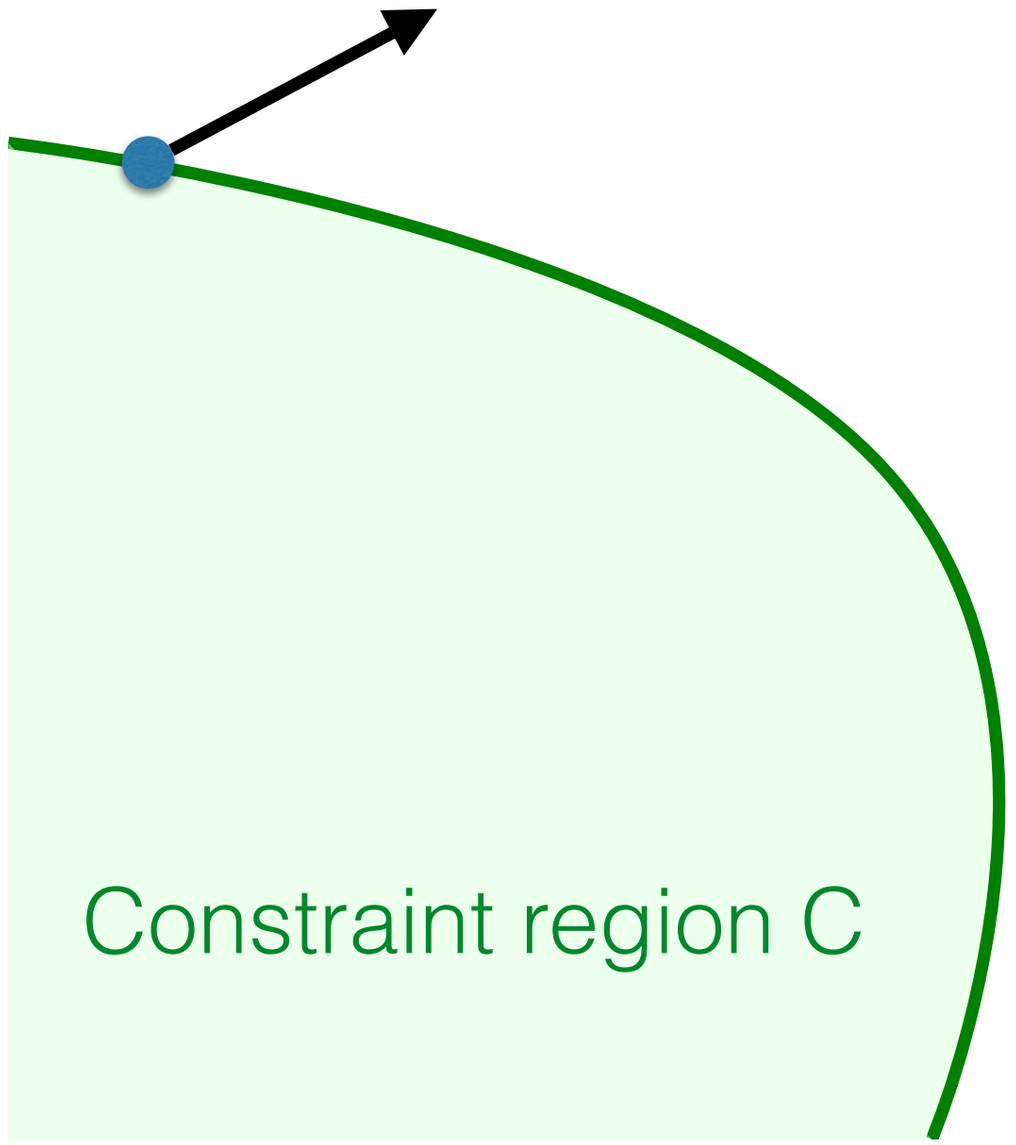
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

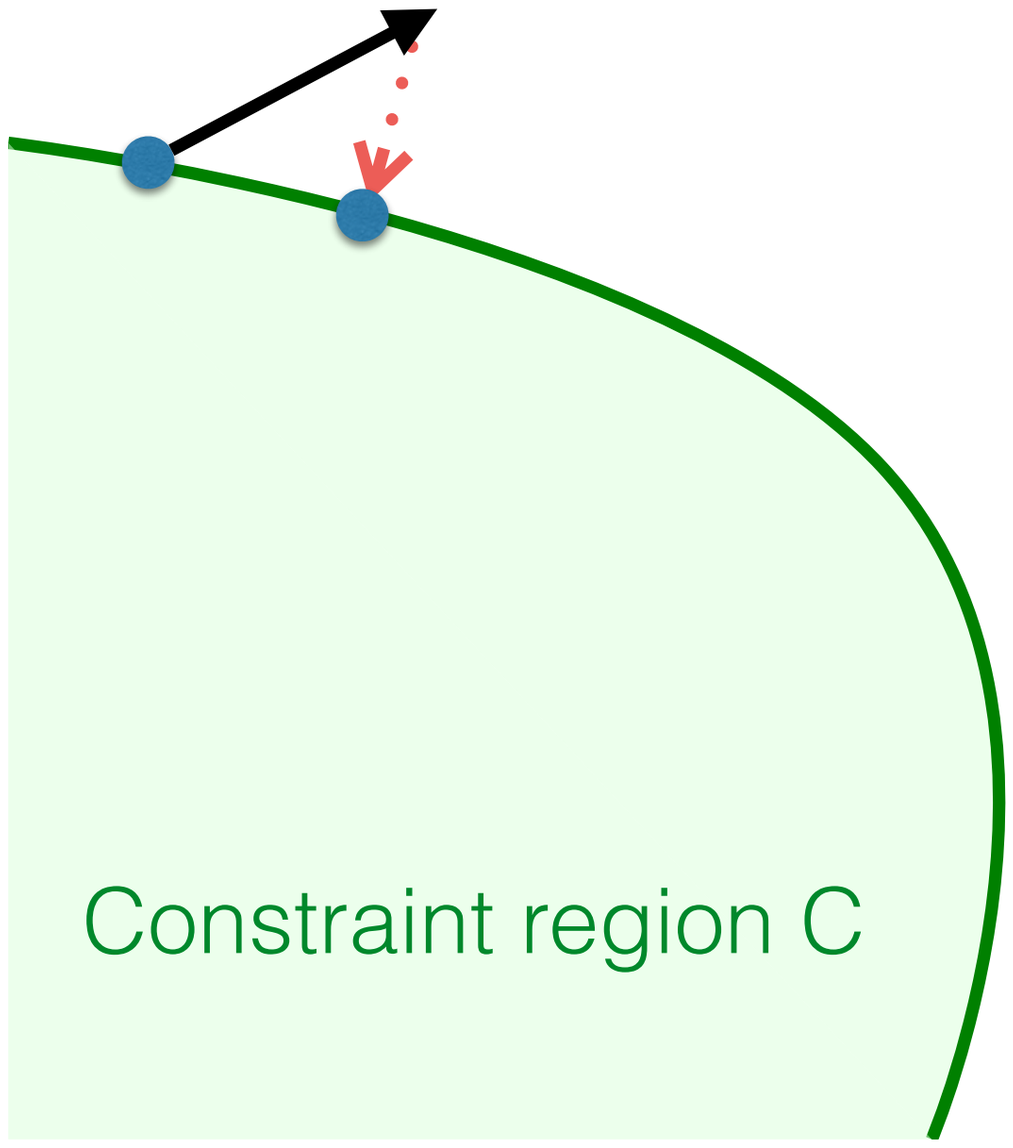
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

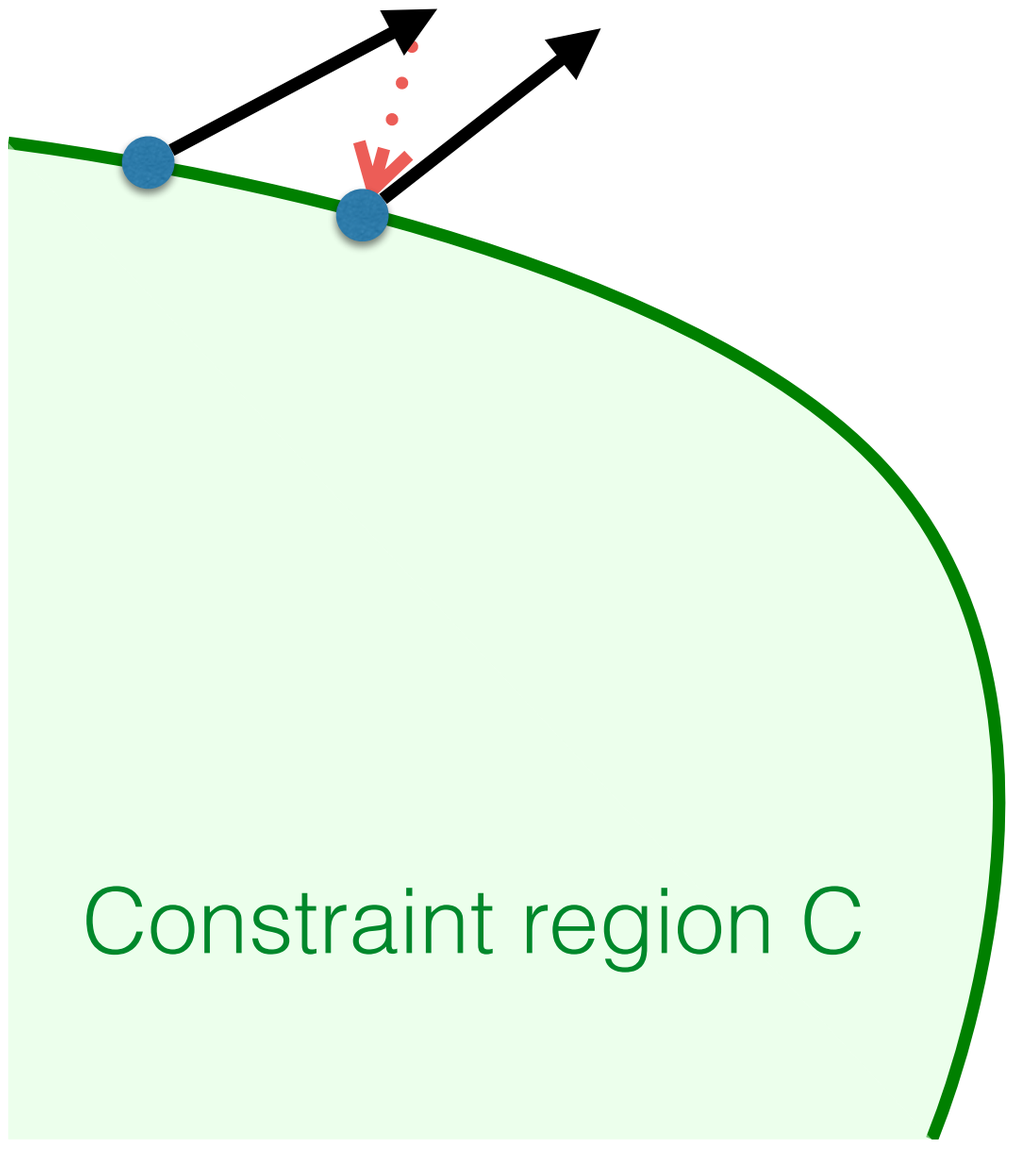
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

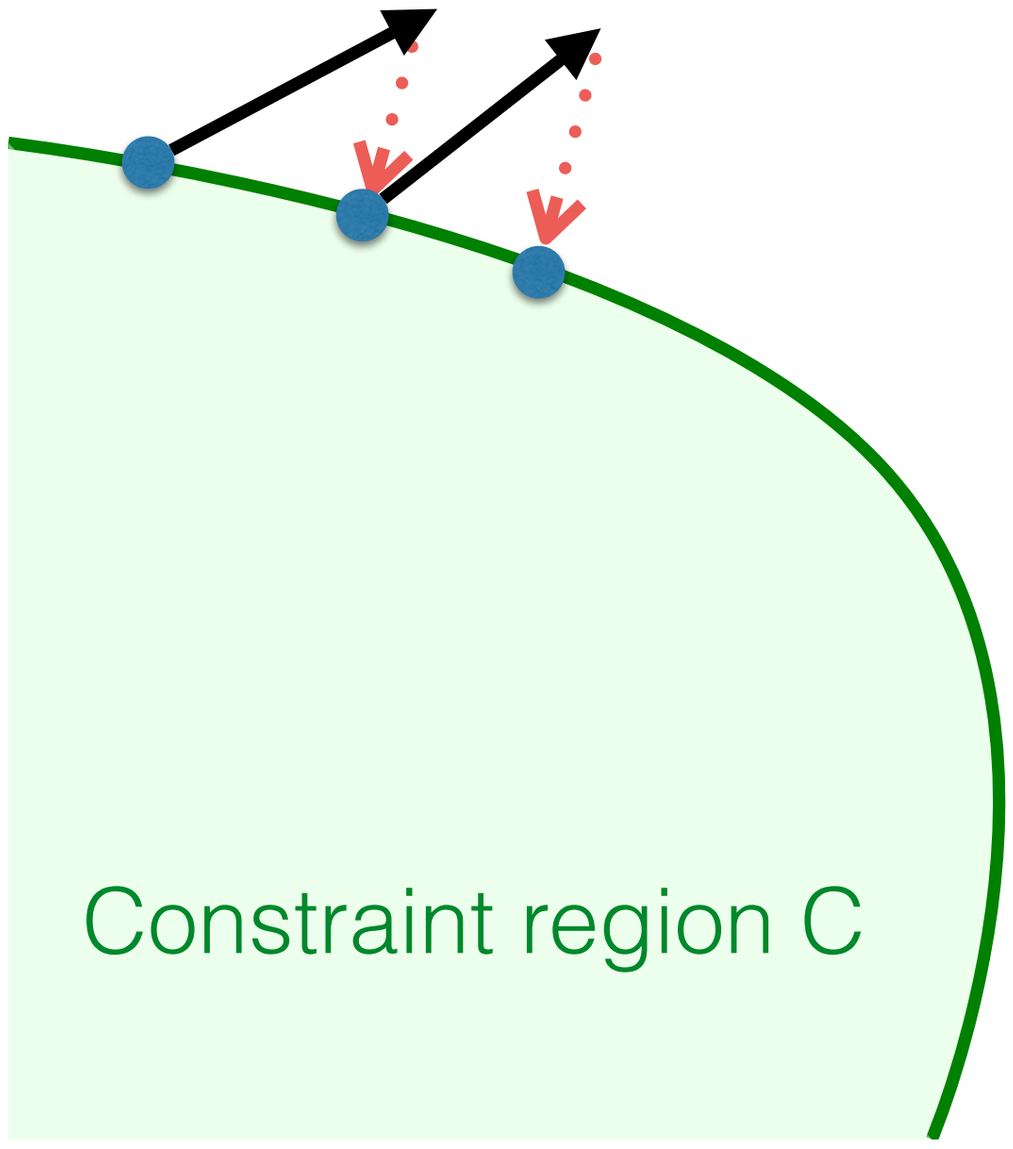
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

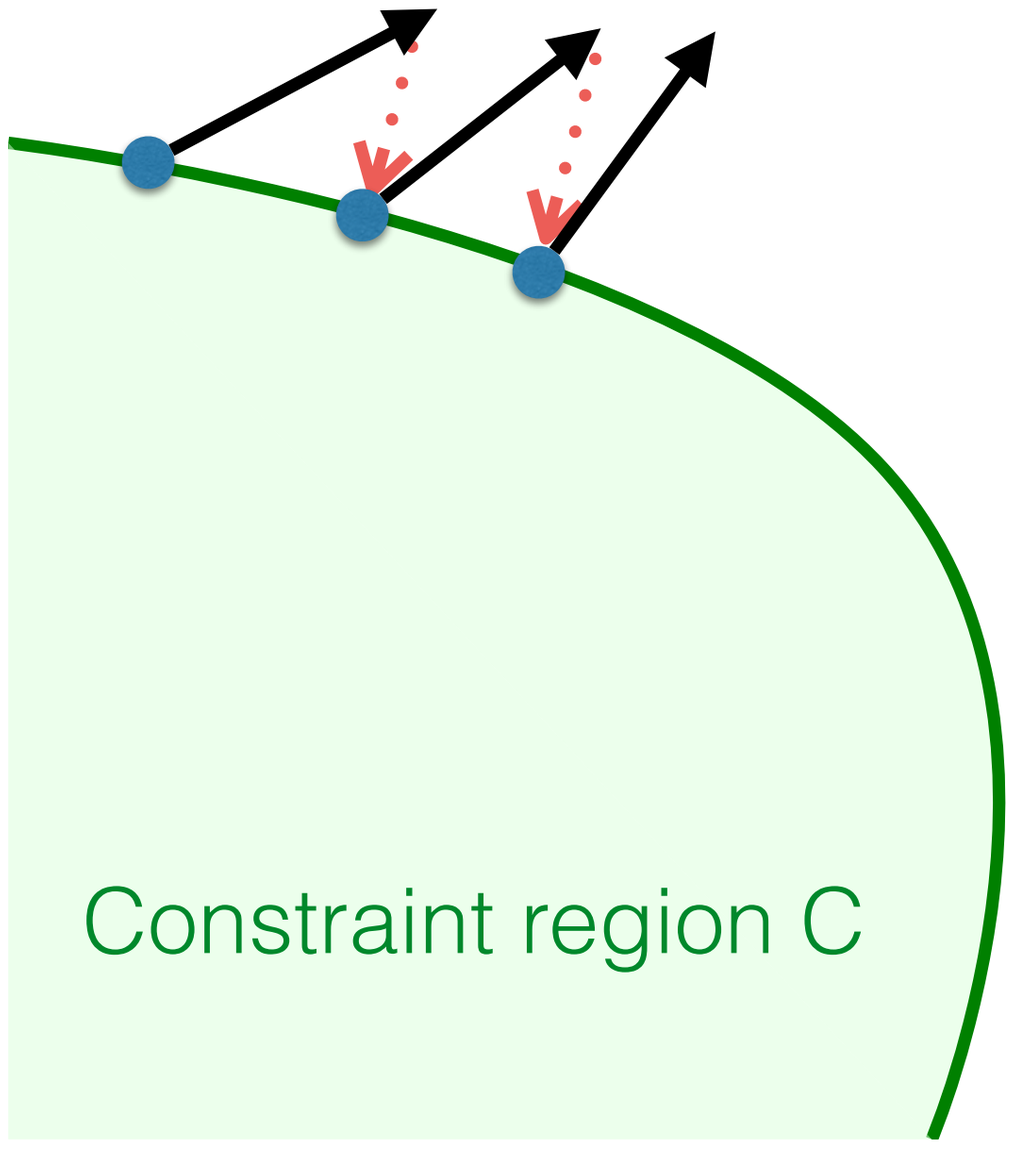
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

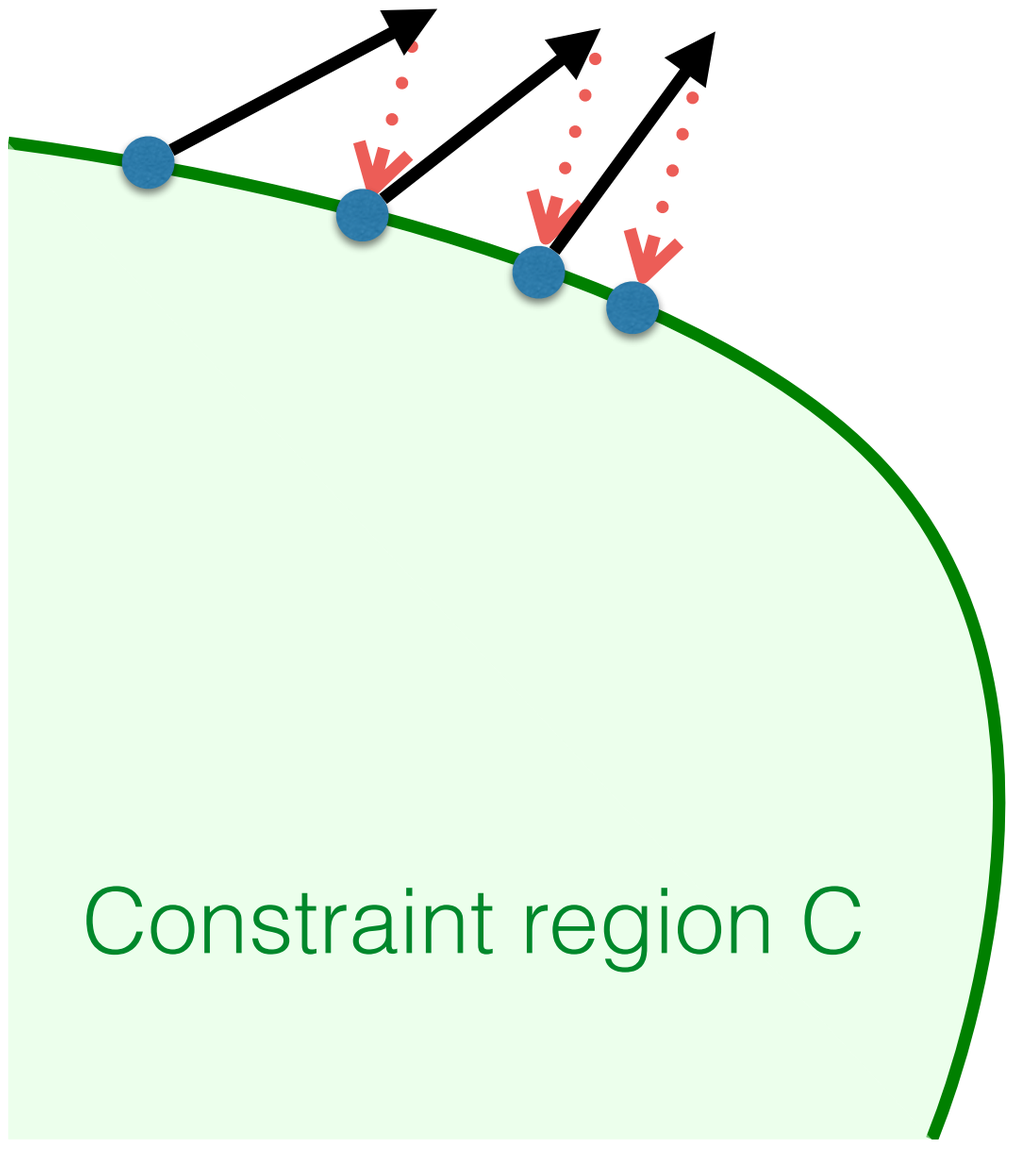
Repeat the following:

1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$



Projected Gradient Descent

Repeat the following:

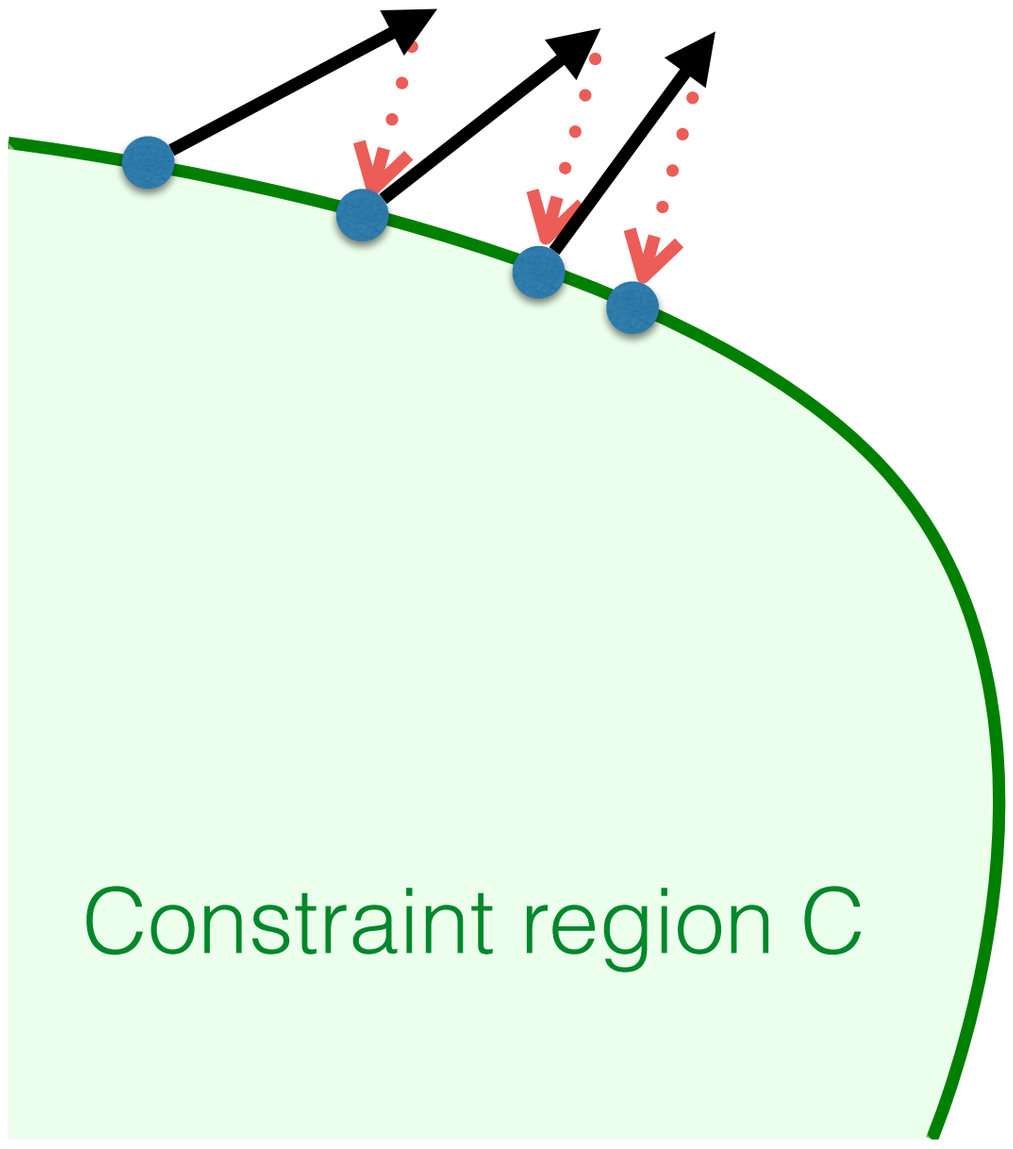
1. Compute gradient $\nabla f(\theta^i)$
2. Take a gradient step

$$\tilde{\theta}^{i+1} \leftarrow \theta^i + \eta \nabla f(\theta^i)$$

3. Project back to constraint set

$$\theta^{i+1} \leftarrow P_{\mathcal{C}}(\tilde{\theta}^{i+1})$$

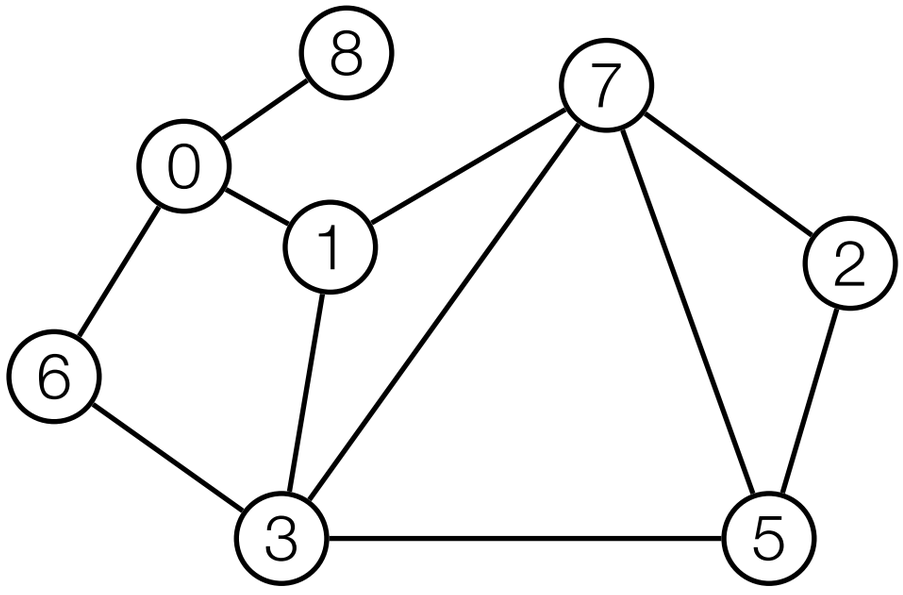
Computational bottleneck



Graph Sparsity Projection

Given a node-weighted graph, find a set of **g subtrees** with **total size s** so that the **sum of node weights** is maximized.

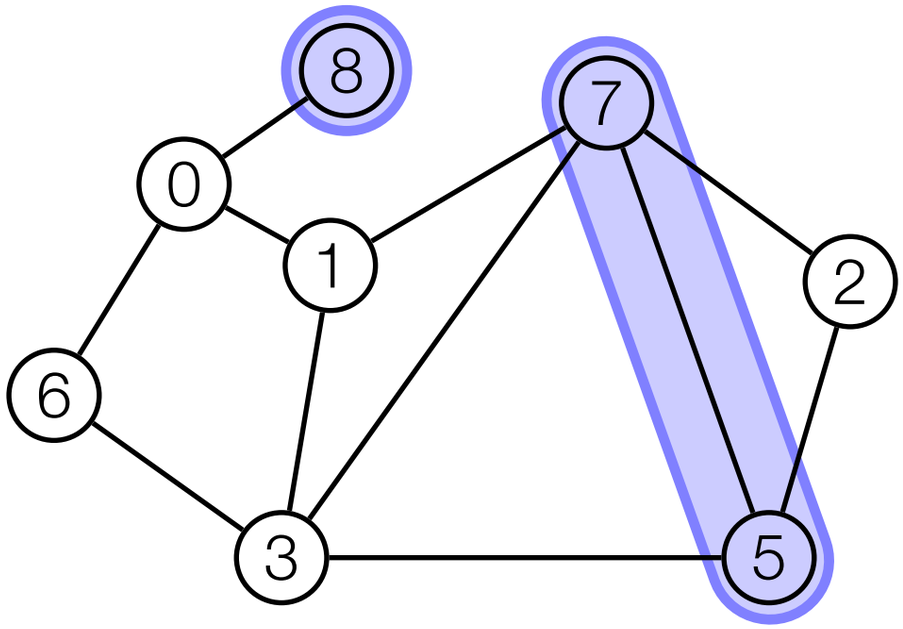
Example: g = 2, s = 3



Graph Sparsity Projection

Given a node-weighted graph, find a set of **g subtrees** with **total size s** so that the **sum of node weights** is maximized.

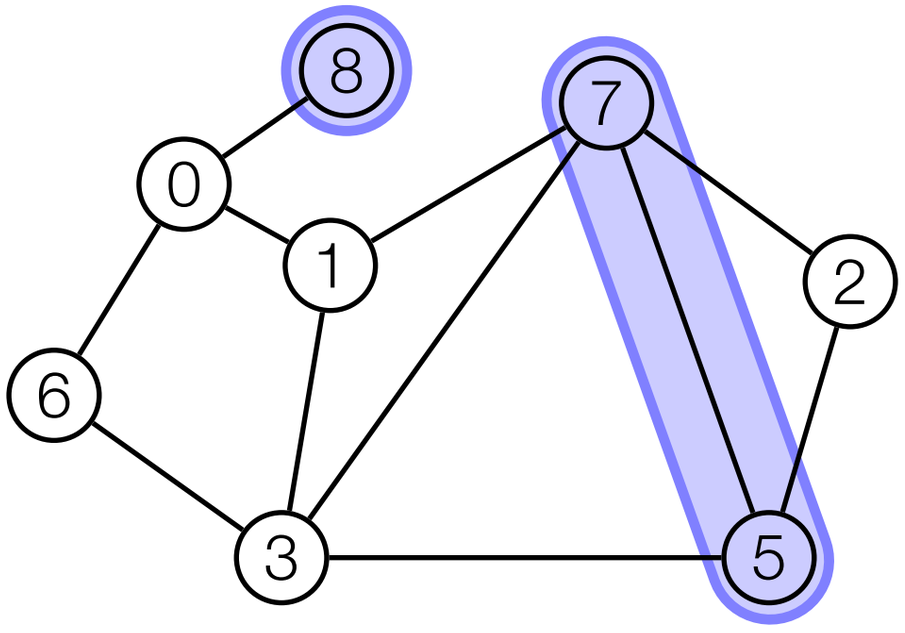
Example: g = 2, s = 3

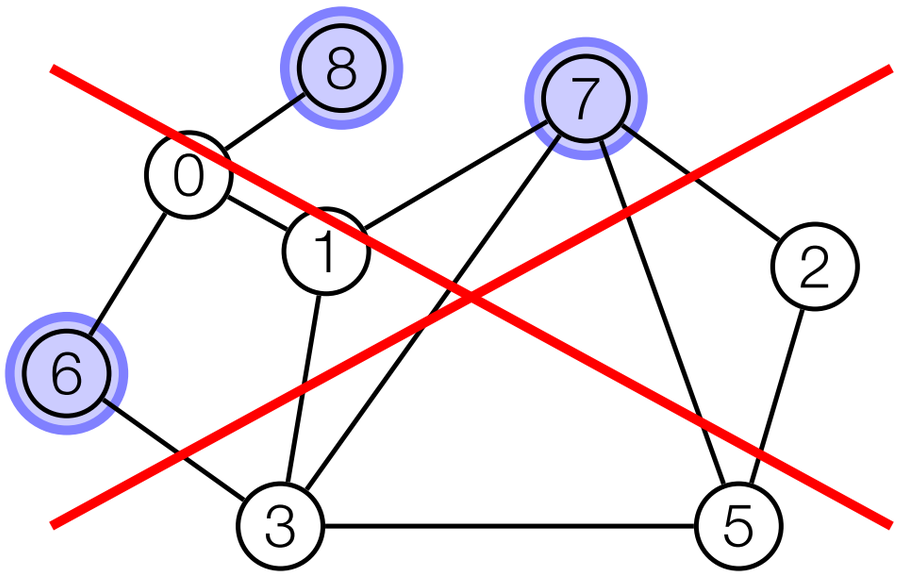


Graph Sparsity Projection

Given a node-weighted graph, find a set of **g subtrees** with **total size s** so that the **sum of node weights** is maximized.

Example: g = 2, s = 3

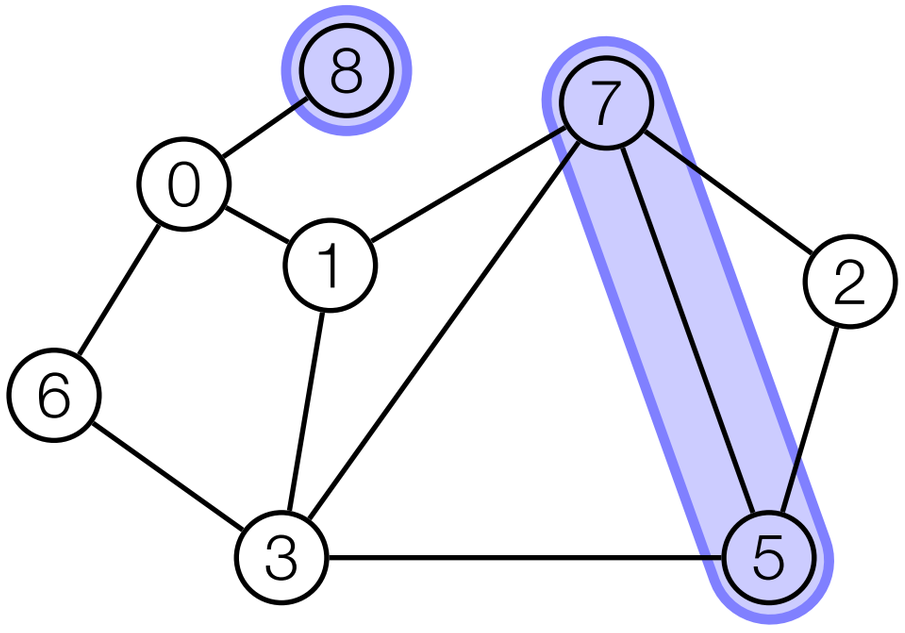


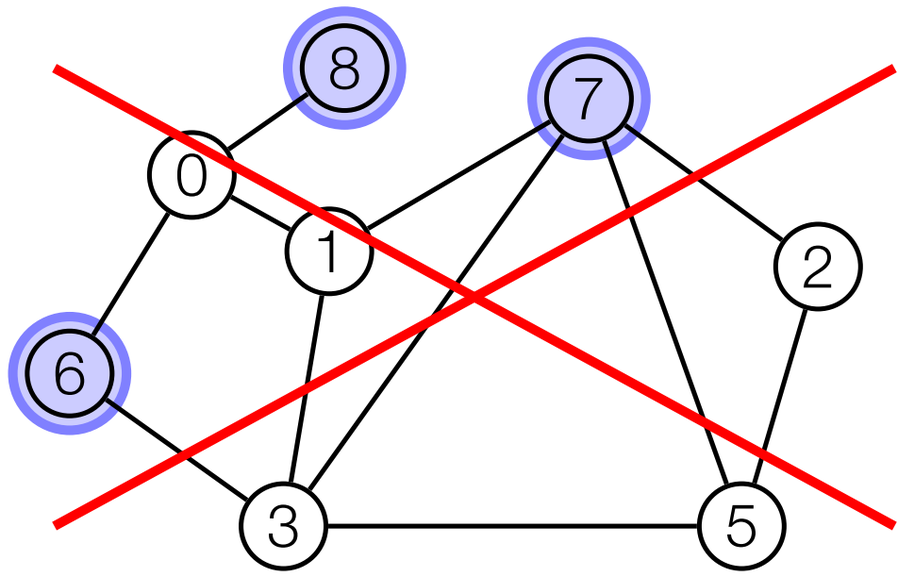


Graph Sparsity Projection

Given a node-weighted graph, find a set of **g subtrees** with **total size s** so that the **sum of node weights** is maximized.

Example: g = 2, s = 3





But: This problem is **NP-hard**.

How Can Approximation Help?

Natural relaxation: **“Tail” Approximation**

Given a point $p \in \mathbb{R}^d$, find a $q \in \mathcal{C}$ such that

$$\|p - q\|_2 \leq \min_{q' \in \mathcal{C}} \|p - q'\|_2$$

How Can Approximation Help?

Natural relaxation: **“Tail” Approximation**

Given a point $p \in \mathbb{R}^d$, find a $q \in \mathcal{C}$ such that

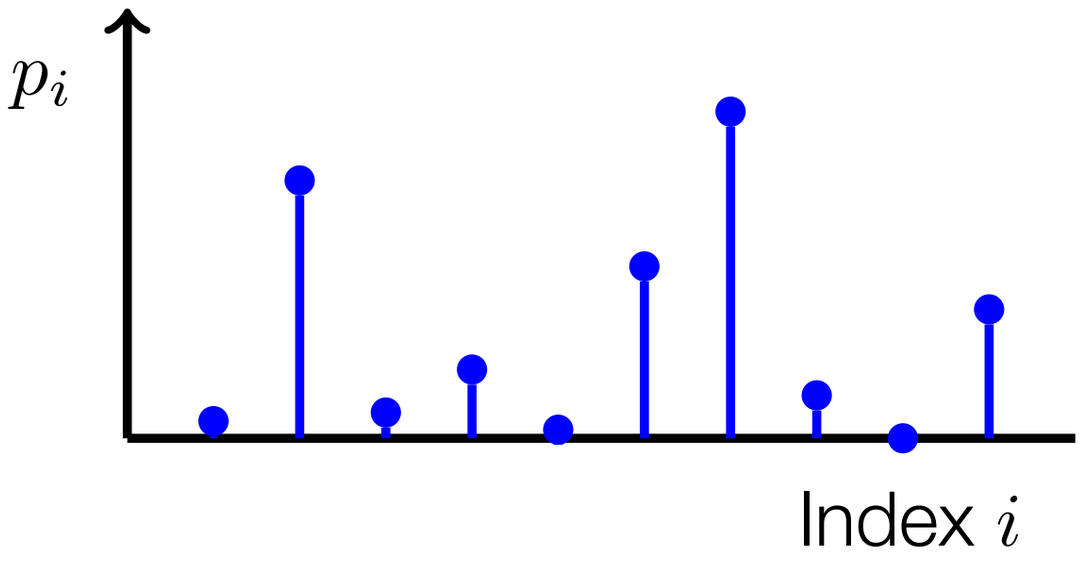
$$\|p - q\|_2 \;\leq\; c_{\mathcal{T}} \cdot \min_{q' \in \mathcal{C}} \|p - q'\|_2$$

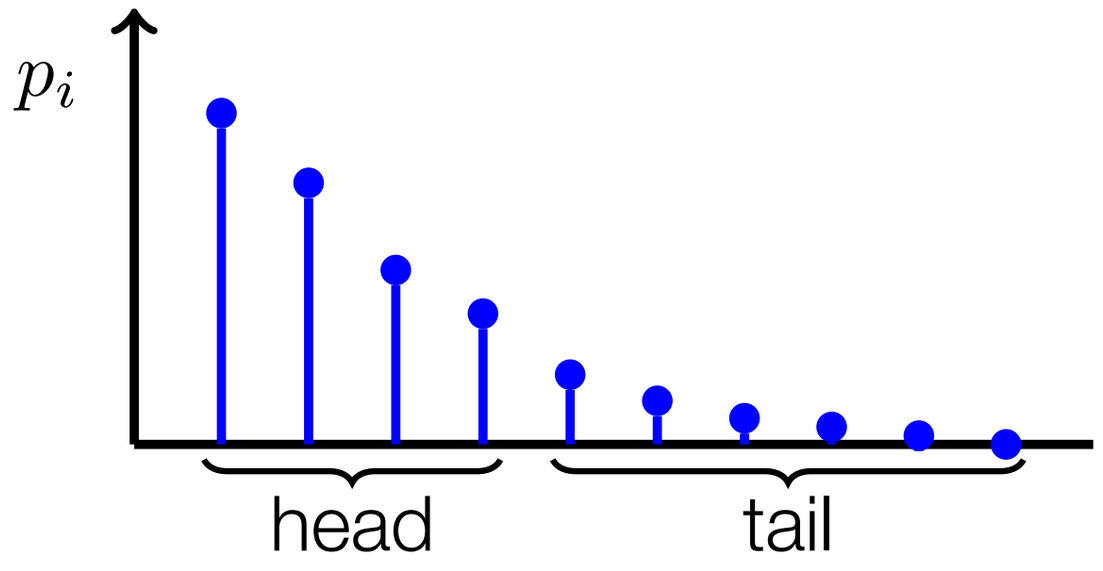
How Can Approximation Help?

Natural relaxation: **"Tail" Approximation**

Given a point $p \in \mathbb{R}^d$, find a $q \in \mathcal{C}$ such that

$$\|p - q\|_2 \;\leq\; c_{\mathcal{T}} \cdot \min_{q' \in \mathcal{C}} \|p - q'\|_2$$





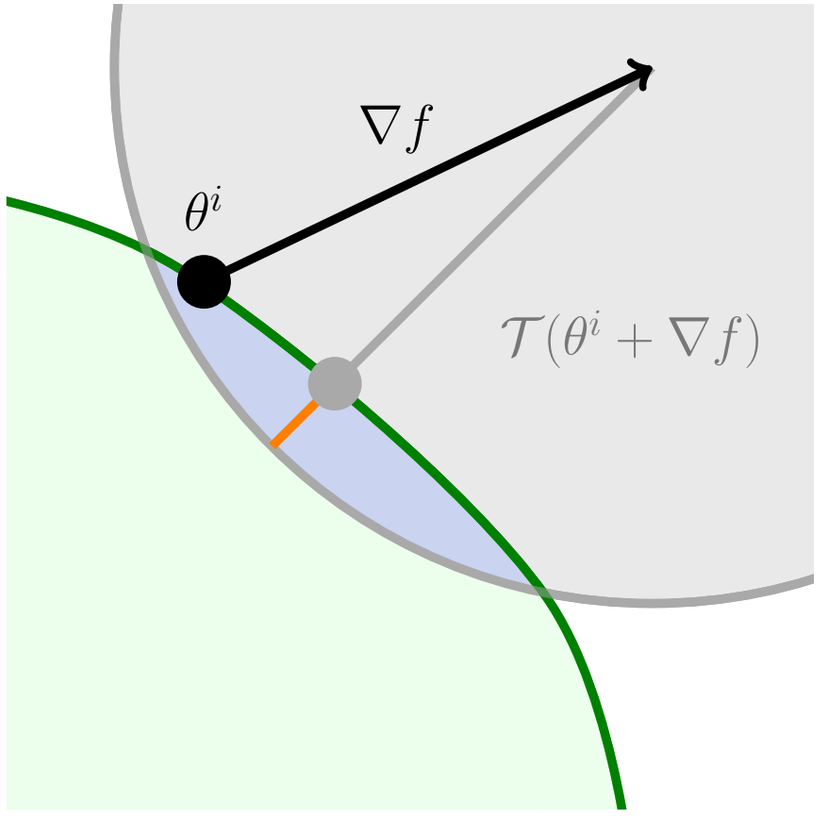
How Can Approximation Help?

Natural relaxation: **“Tail” Approximation**

Given a point $p \in \mathbb{R}^d$, find a $q \in \mathcal{C}$ such that

$$\|p - q\|_2 \;\leq\; c_{\mathcal{T}} \cdot \min_{q' \in \mathcal{C}} \|p - q'\|_2$$

Gradient descent with tail approximation fails for any $c_T > 1$!



Simple Fix: More Approximation

Complementary guarantee: **“Head” Approximation**

Given a point $g \in \mathbb{R}^d$, find a $q \in \mathcal{C} \cap \mathbb{S}^{d-1}$ such that

$$\langle g, q \rangle \;\geq\; c_{\mathcal{H}} \cdot \max_{q' \in \mathcal{C} \cap \mathbb{S}^{d-1}} \langle g, q' \rangle \;.$$

Simple Fix: More Approximation

Complementary guarantee: **"Head" Approximation**

Given a point $g \in \mathbb{R}^d$, find a $q \in \mathcal{C} \cap \mathbb{S}^{d-1}$ such that

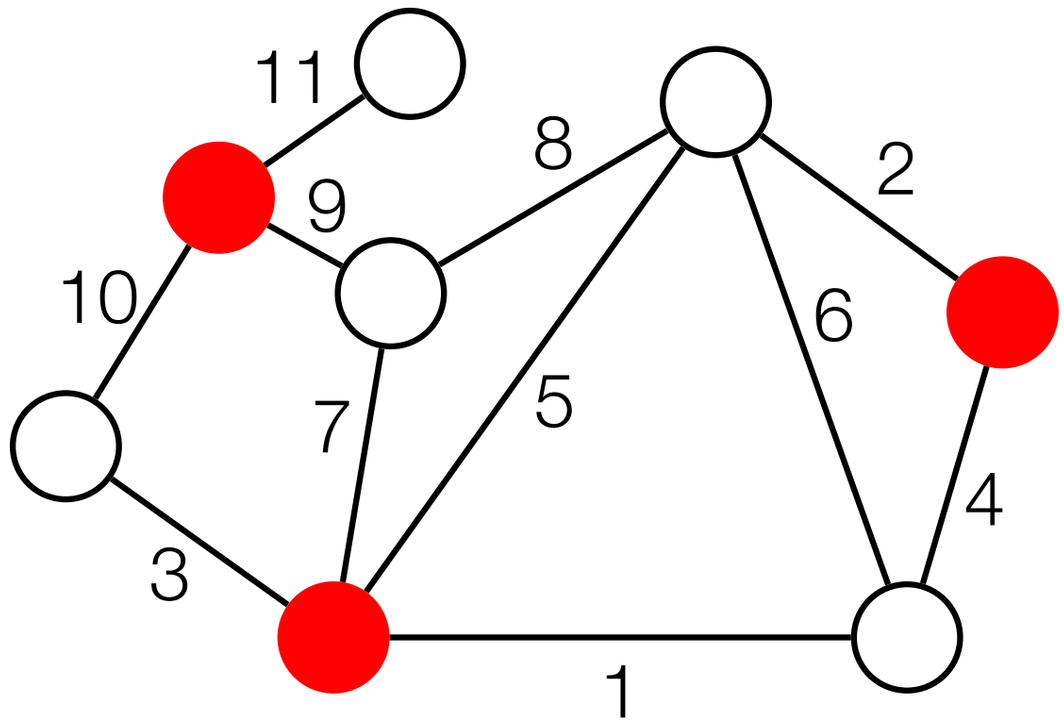
$$\langle g, q \rangle \;\geq\; c_{\mathcal{H}} \cdot \max_{q' \in \mathcal{C} \cap \mathbb{S}^{d-1}} \langle g, q' \rangle \;.$$

Head + Tail approximation combined provably succeeds.

- First analyzed in [Hegde-Indyk-Schmidt'14]
- Generalized in [Cohen-Hegde-Jegelka-Schmidt'??]

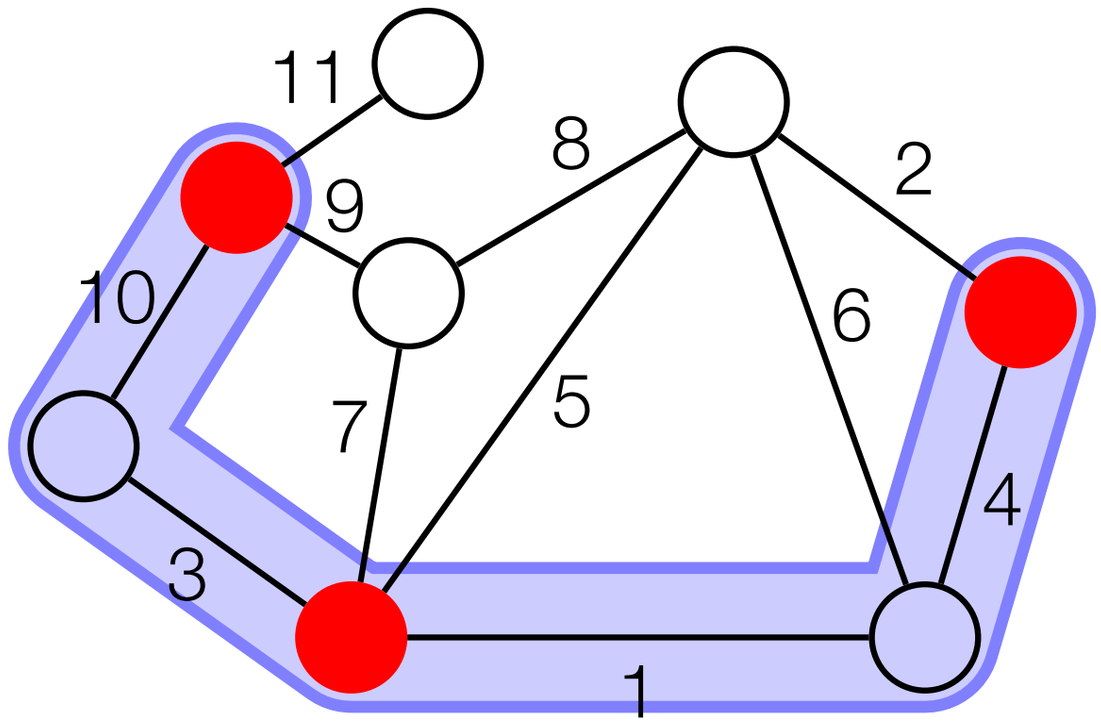
Prize-Collecting Steiner Tree

Generalization of the **Steiner tree** problem



Prize-Collecting Steiner Tree

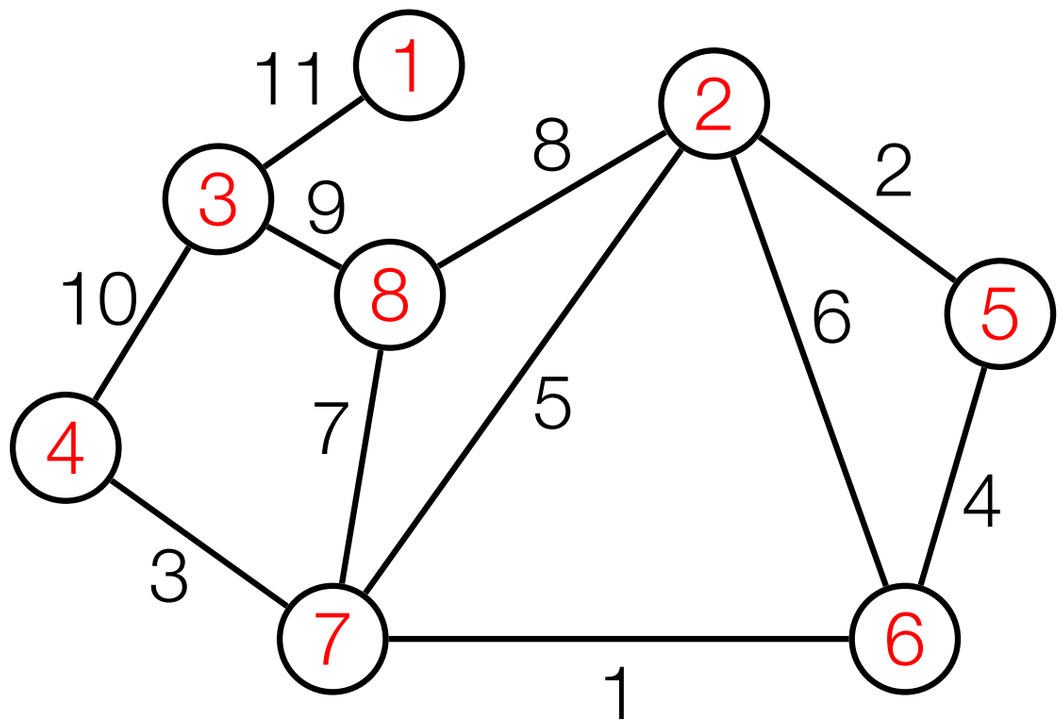
Generalization of the **Steiner tree** problem



Prize-Collecting Steiner Tree

Generalization of the **Steiner tree** problem

Every node has a given prize π.

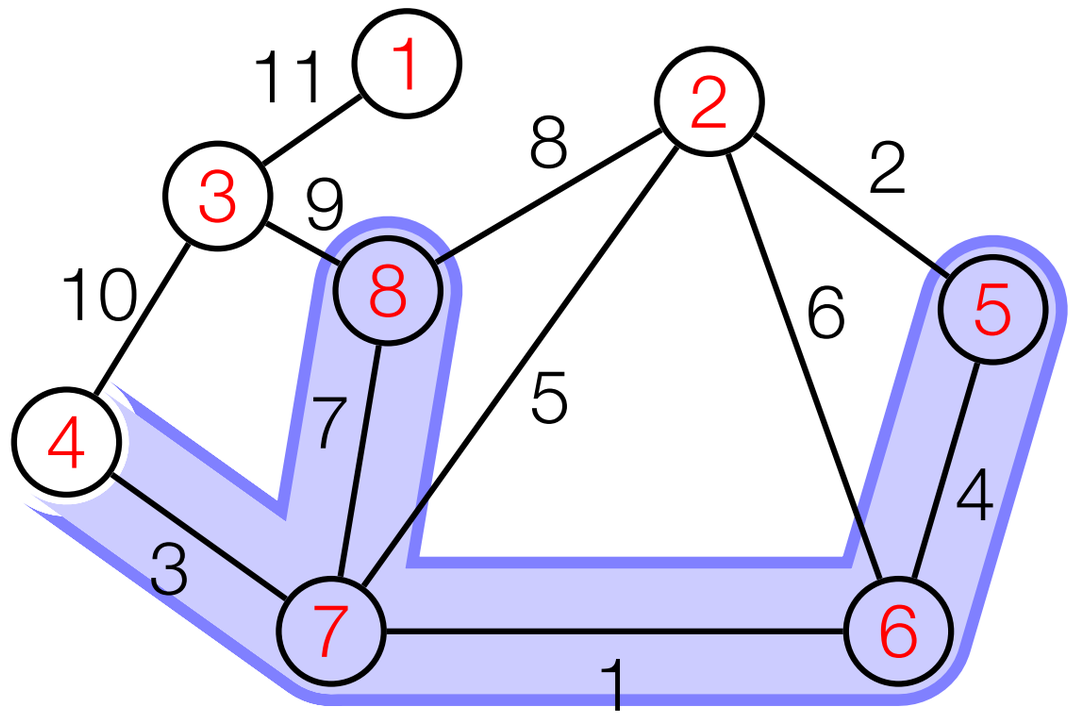


Goal: find a tree T minimizing $\sum_{v \notin V_T} \pi(v) + \sum_{e \in E_T} c(e)$

Prize-Collecting Steiner Tree

Generalization of the **Steiner tree** problem

Every node has a given prize π.

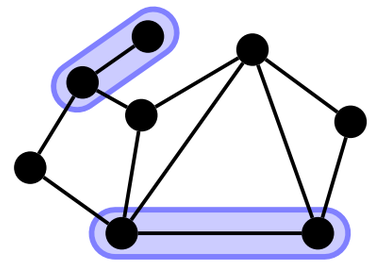


Goal: find a tree T minimizing $\sum_{v \notin V_T} \pi(v) + \sum_{e \in E_T} c(e)$

Connection to Graph Sparsity

Starting point: Goemans-Williamson approximation scheme

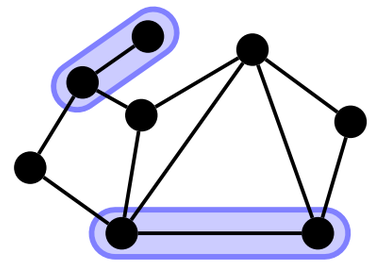
- Generalize to Steiner **forests**



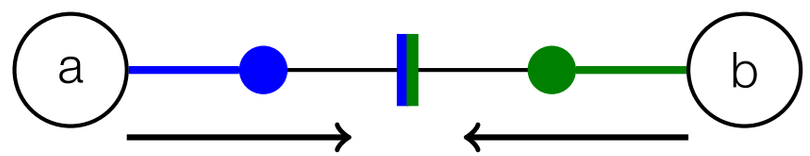
Connection to Graph Sparsity

Starting point: Goemans-Williamson approximation scheme

- Generalize to Steiner **forests**



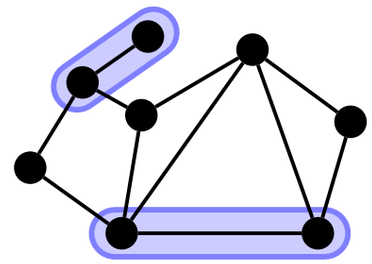
- Develop a **nearly-linear time** and **practical** algorithm



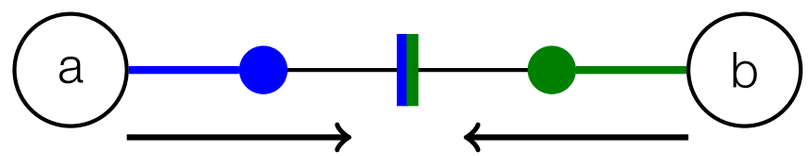
Connection to Graph Sparsity

Starting point: Goemans-Williamson approximation scheme

- Generalize to Steiner **forests**



- Develop a **nearly-linear time** and **practical** algorithm



- **Reduce** graph sparsity projection to a sequence of PCSF instances

$$\sum_{v \notin V_T} \pi(v) + \lambda \cdot \sum_{e \in E_T} c(e)$$

Estimation Guarantee

Setup: $$y = X\theta^* + e$$

If X is a restricted isometry for graph-sparse vectors, projected gradient descent converges in $O(\log \|\theta^*\| / \|e\|)$ iterations and finds an estimate $\hat{\theta}$ such that:

$$\|\hat{\theta} - \theta^*\| \leq O(\|e\|)$$

Estimation Guarantee

Setup: $y = X\theta^* + e$

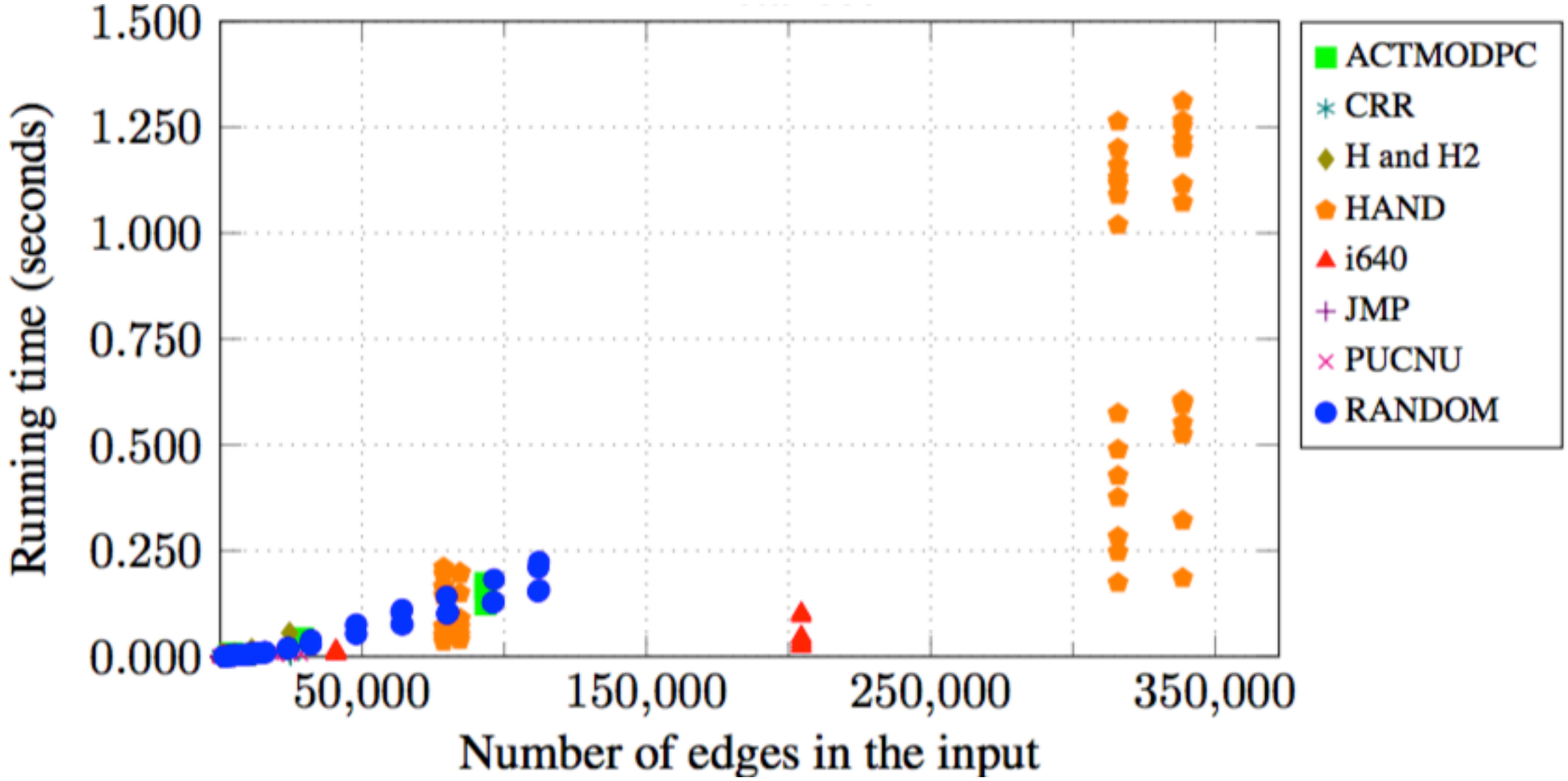
If X is a restricted isometry for graph-sparse vectors, projected gradient descent converges in $O(\log \|\theta^*\| / \|e\|)$ iterations and finds an estimate $\hat{\theta}$ such that:

$$\|\hat{\theta} - \theta^*\| \leq O(\|e\|)$$

Same guarantee for approximate and exact projections.

Experiments: Prize Collecting Steiner Forest

Running time on instance of the 2014 DIMACS challenge.

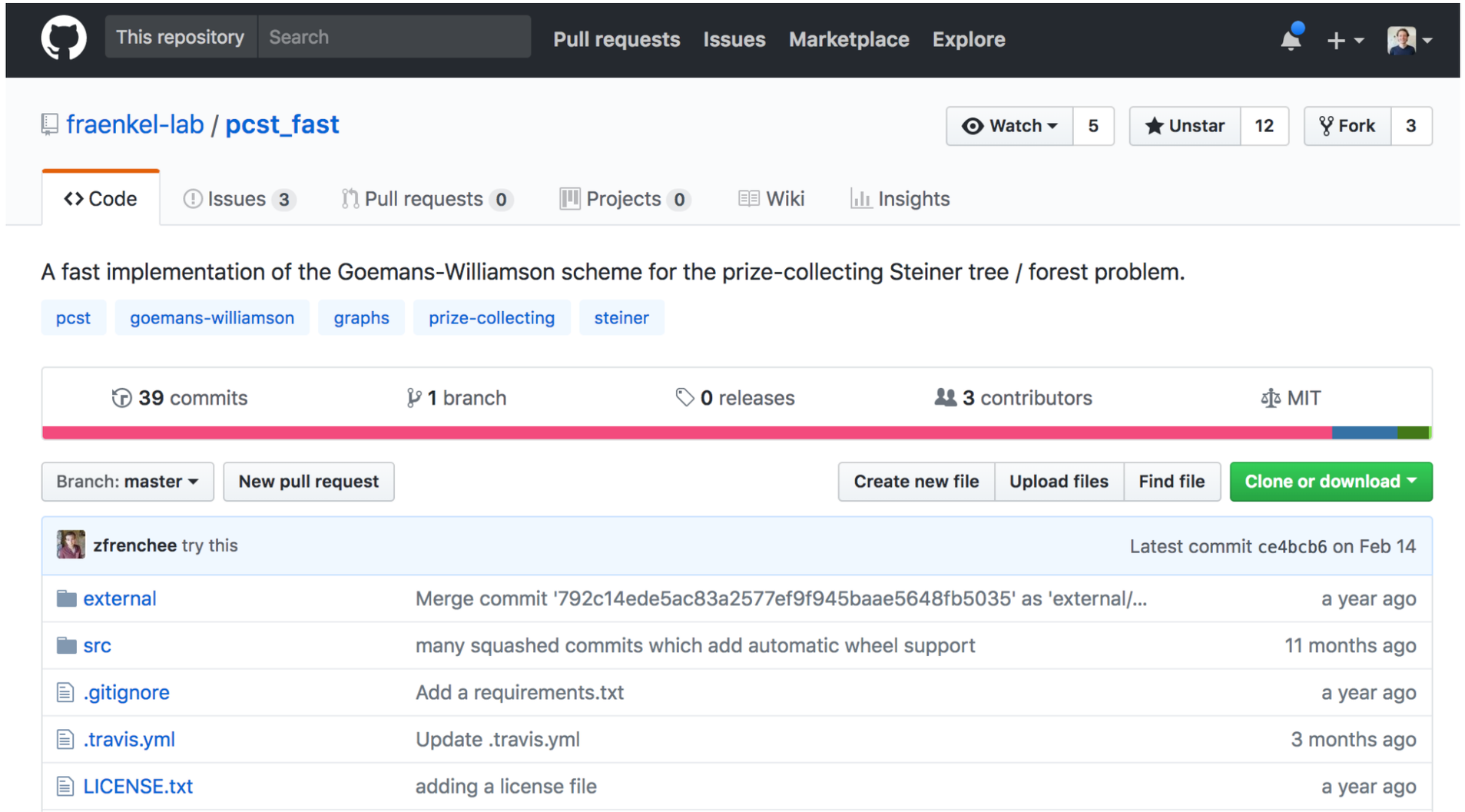


→ **Nearly-linear** scaling with number of edges.

Experiments: Prize collecting Steiner Forest

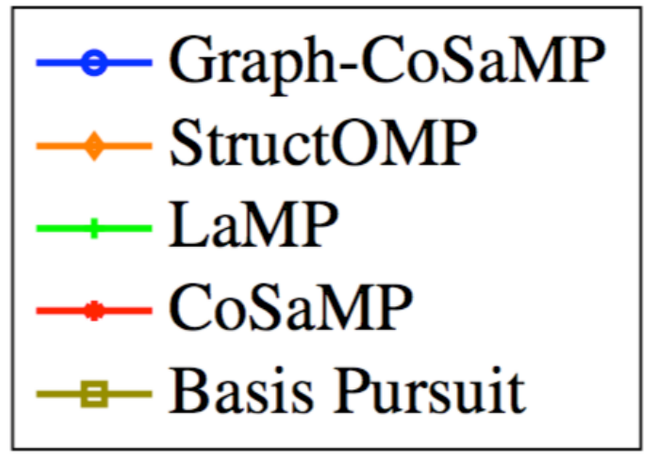
Algorithm	Solution value	Time
Hegde, Indyk, Schmidt	5601.8	**0.2**
mozartballs	5058.5	122.1
staynerd	**5057.4**	143.0
polito	5130.6	160.1
scipjack	5903.5	1052.3
Althaus-Blumenstock	5831.9	2113.5
KTS	5129.0	4842.1
heinz-dc	5695.5	6420.0

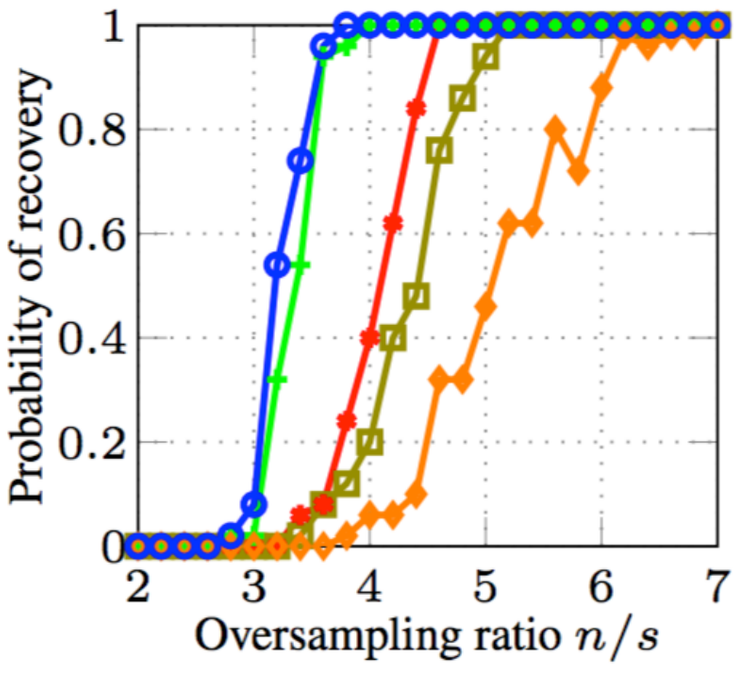
Code for PCSF

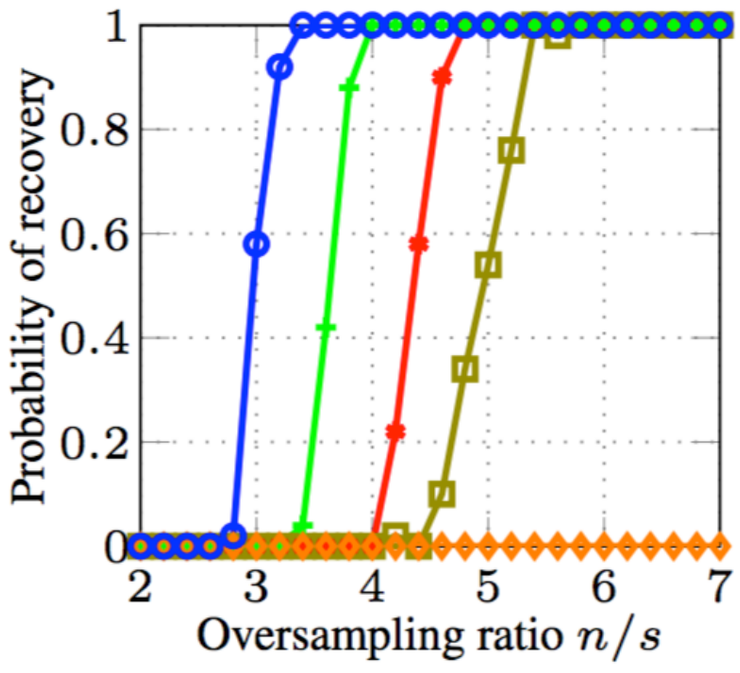


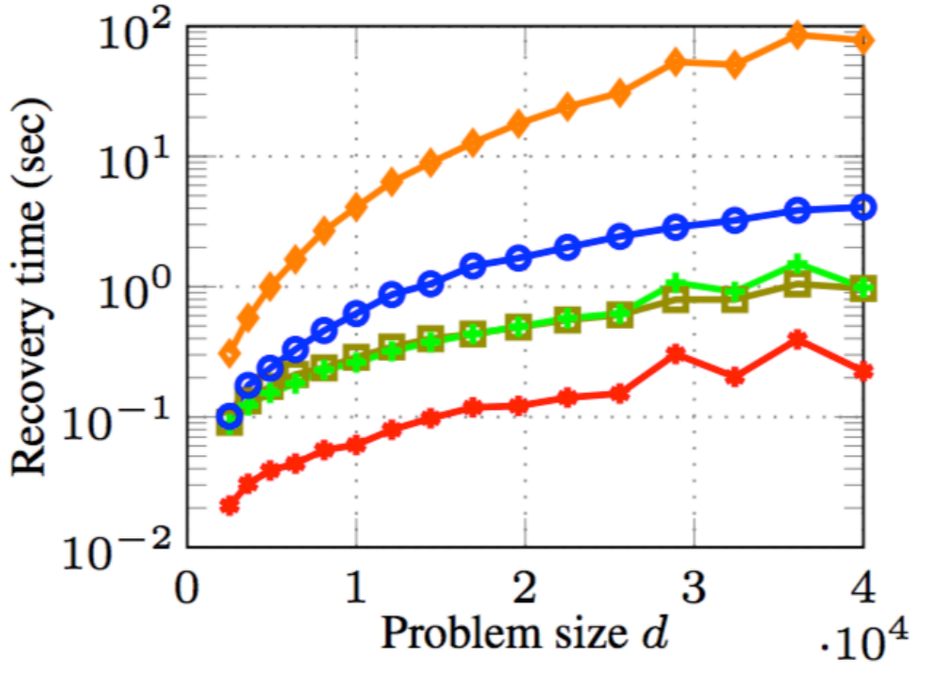
- C++ implementation with Python interface
- Has unit tests
- `pip install pcst_fast`

Compressive Sensing Experiments





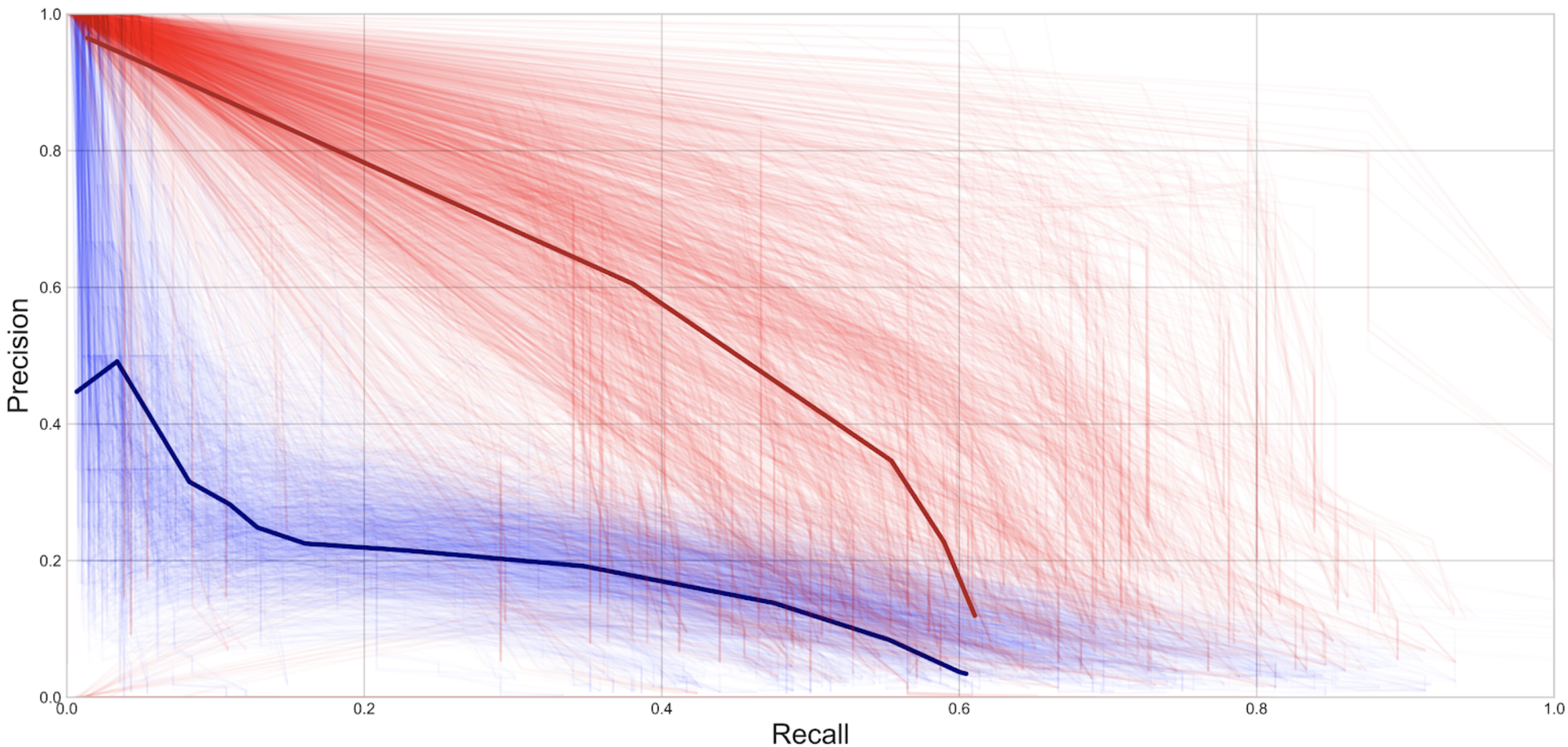




Graph-Sparse Linear Regression

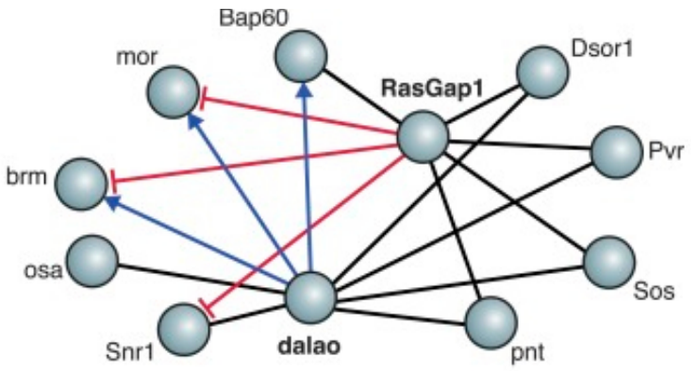
Support recovery on semi-synthetic data.

Graph-sparsity
Lasso



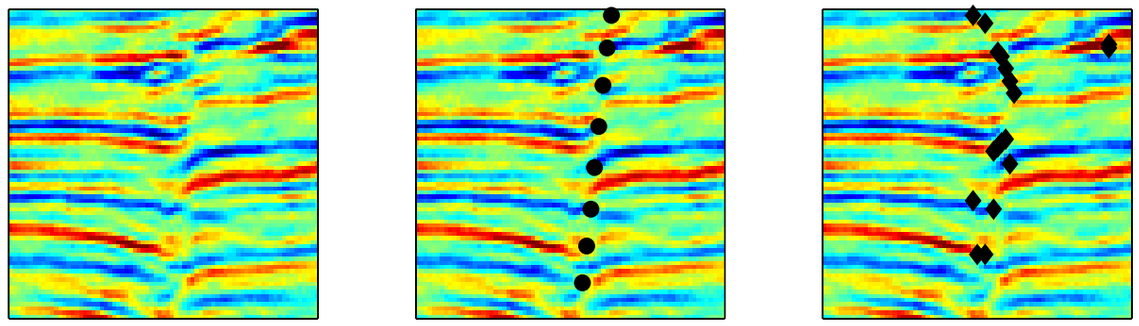
Code on github (but not production-ready yet).

Applications of PCSF



Omics Integrator

Outage Detection
in Power Networks



Computational Geophysics

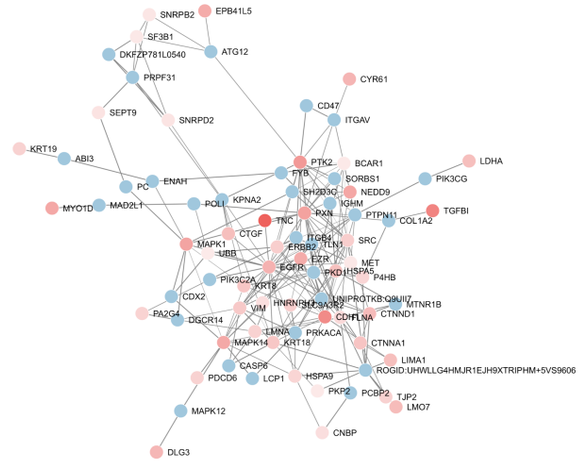
Event Detection
in Social Networks

Computational Biology

Conclusions

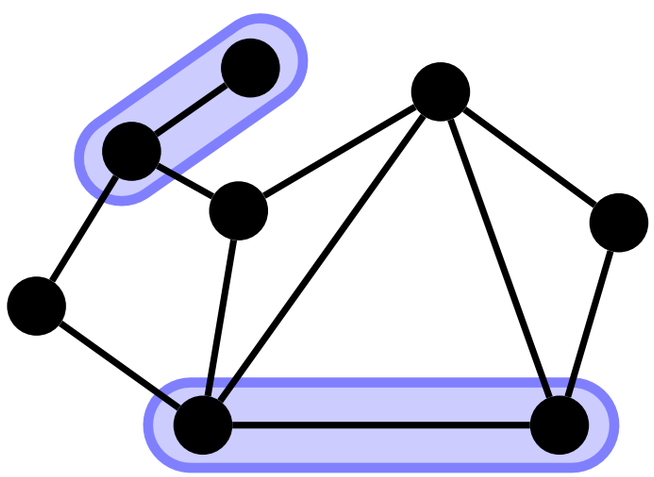
Linear regression with graph constraints

→ Find a predictive subgraph.



Fast, rigorous approximations for PCSF

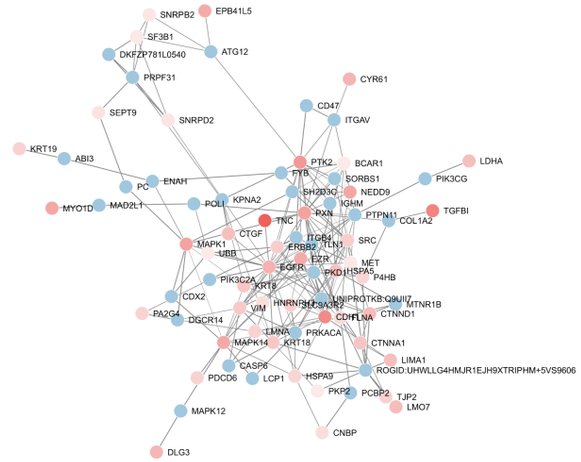
→ Collect large node prizes for little edge cost.



Conclusions

Linear regression with graph constraints

→ Find a predictive subgraph.



Fast, rigorous approximations for PCSF

→ Collect large node prizes for little edge cost.

Future directions

- Applications?
- Other graph constraints in regression?