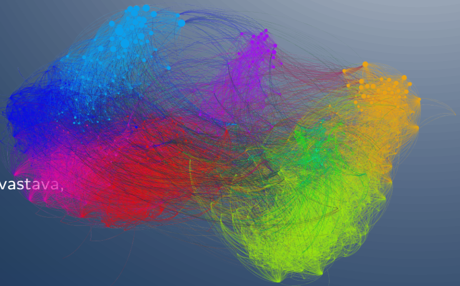
Graphs in Machine Learning

Spectral Graph Sparsifiers in ML

Ridge Sparsifiers and Applications

Michal Valko

Inria & ENS Paris-Saclay, MVA



Partially based on material by: Rob Fergus, Nikhil Srivastava,
Yiannis Koutis, Joshua Batson, Daniel Spielman

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2$$

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2 = \left\|\mathbf{L}_G^{-1/2}\mathbf{b}_e\right\|^2$$

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2 = \left\|\mathbf{L}_G^{-1/2}\mathbf{b}_e\right\|^2 = \mathbf{b}_e^\mathsf{T}\mathbf{L}_G^{-1}\mathbf{b}_e$$

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2 = \left\|\mathbf{L}_G^{-1/2}\mathbf{b}_e\right\|^2 = \mathbf{b}_e^\mathsf{T}\mathbf{L}_G^{-1}\mathbf{b}_e$$

Random walk intuition: inverse of number of alternative paths

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2 = \left\|\mathbf{L}_G^{-1/2}\mathbf{b}_e\right\|^2 = \mathbf{b}_e^\mathsf{T}\mathbf{L}_G^{-1}\mathbf{b}_e = R_{\mathsf{eff}}(e)$$

Random walk intuition: inverse of number of alternative paths

Electrical intuition: We want to find an electrically similar H and the importance of the edge is its effective resistance $R_{\mathsf{eff}}(e)$.

$R_{\mathsf{eff}}(e)$ is the potential difference between the nodes when injecting a unit current

Spectral Graph Sparsification: Intuition

What it this rescaling $\mathbf{v}_e = \mathbf{L}_G^{-1/2}\mathbf{b}_e$ doing to the norm?

$$\|\mathbf{v}_e\|^2 = \left\|\mathbf{L}_G^{-1/2}\mathbf{b}_e\right\|^2 = \mathbf{b}_e^\intercal \mathbf{L}_G^{-1}\mathbf{b}_e = R_{\text{eff}}(e)$$

Random walk intuition: inverse of number of alternative paths

Electrical intuition: We want to find an electrically similar H and the importance of the edge is its effective resistance $R_{\text{eff}}(e)$.

$R_{\text{eff}}(e)$ is the potential difference between the nodes when injecting a unit current

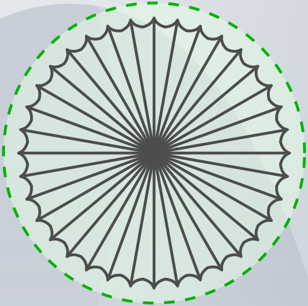
Edges with higher R_{eff} are more **electrically significant**!

Spectral Graph Sparsification: Intuition

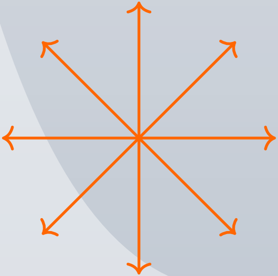
What are we doing by choosing H?

Spectral Graph Sparsification: Intuition

What are we doing by choosing H?



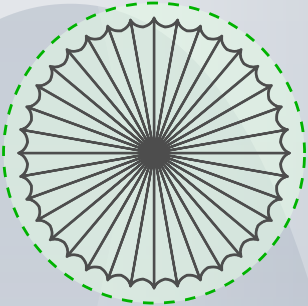
m vectors in $\mathbb{R}^n$



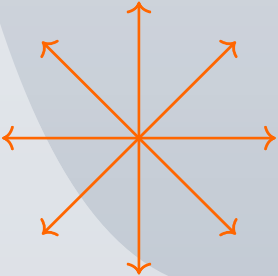
$\tilde{O}(n)$ vectors in $\mathbb{R}^n$

Spectral Graph Sparsification: Intuition

What are we doing by choosing H?



m vectors in $\mathbb{R}^n$



$\tilde{O}(n)$ vectors in $\mathbb{R}^n$

We take a subset of these $\mathbf{e}_e$s and scale them!

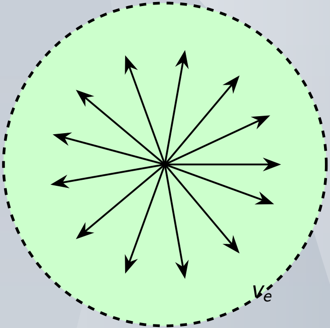
Spectral Graph Sparsification: Intuition

What kind of scaling go we want?

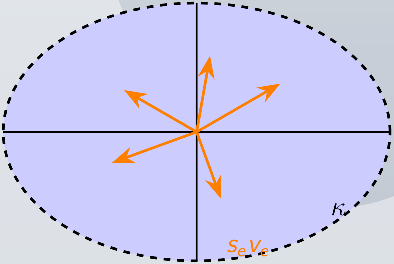
Spectral Graph Sparsification: Intuition

What kind of scaling go we want?

m vectors in $\mathbf{R}^n$



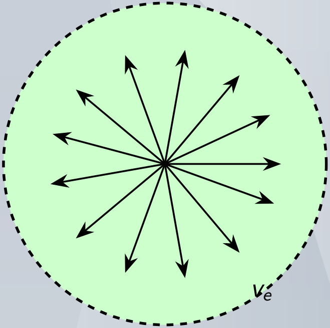
$\mathbf{O}^{\sim}(n)$ vectors in $\mathbf{R}^n$



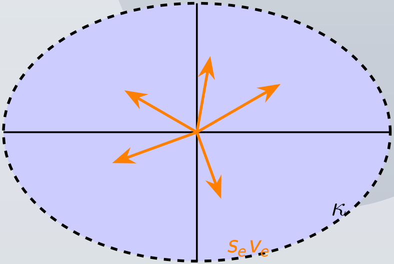
Spectral Graph Sparsification: Intuition

What kind of scaling go we want?

$\mathbf{m}$ vectors in $\mathbf{R}^n$



$\mathbf{O}^{\sim}(n)$ vectors in $\mathbf{R}^n$

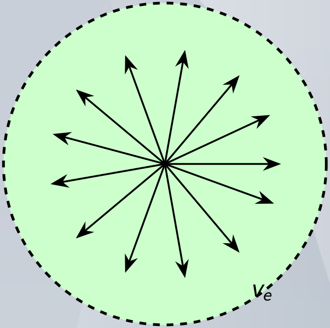


Such that the blue ellipsoid looks like identity!

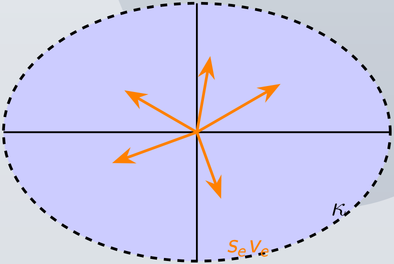
Spectral Graph Sparsification: Intuition

What kind of scaling go we want?

$\mathbf{m}$ vectors in $\mathbf{R}^n$



$\mathbf{O}^{\sim}(n)$ vectors in $\mathbf{R}^n$



Such that the blue ellipsoid looks like identity!

the blue eigenvalues are between 1 and κ

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Sample $n \log n / \varepsilon^2$ with replacement $p_i \propto \|\mathbf{v}_e\|^2$ (resistances)

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Sample $n \log n / \varepsilon^2$ with replacement $p_i \propto \|\mathbf{v}_e\|^2$ (resistances)
Reweigh $s_i = \frac{1}{p_i}$ unbiased $\mathbb{E}[\sum_i s_i \mathbf{v}_i \mathbf{v}_i] = \sum_e s_e p_e \mathbf{v}_e \mathbf{v}_e = \sum_e \mathbf{v}_e \mathbf{v}_e$

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Sample $n \log n / \varepsilon^2$ with replacement $p_i \propto \|\mathbf{v}_e\|^2$ (resistances)
Reweigh $s_i = \frac{1}{p_i}$ unbiased $\mathbb{E}[\sum_i s_i \mathbf{v}_i \mathbf{v}_i] = \sum_e s_e p_e \mathbf{v}_e \mathbf{v}_e = \sum_e \mathbf{v}_e \mathbf{v}_e$

Does this work?

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Sample $n \log n/\varepsilon^2$ with replacement $p_i \propto \|\mathbf{v}_e\|^2$ (resistances)
Reweigh $s_i = \frac{1}{p_i}$ unbiased $\mathbb{E}[\sum_i s_i \mathbf{v}_i \mathbf{v}_i] = \sum_e s_e p_e \mathbf{v}_e \mathbf{v}_e = \sum_e \mathbf{v}_e \mathbf{v}_e$

Does this work?

Application of Matrix Chernoff Bound by Rudelson (1999)

$$1 - \varepsilon \prec \lambda \left(\sum_e s_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T} \right) \prec 1 + \varepsilon$$

Spectral Graph Sparsification

Todo: Given $\mathbf{I} = \sum_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T}$, find a sparse reweighting.

Randomized algorithm that finds $\mathbf{s}$:

Sample $n \log n/\varepsilon^2$ with replacement $p_i \propto \|\mathbf{v}_e\|^2$ (resistances)
Reweigh $s_i = \frac{1}{p_i}$ unbiased $\mathbb{E}[\sum_i s_i \mathbf{v}_i \mathbf{v}_i] = \sum_e s_e p_e \mathbf{v}_e \mathbf{v}_e = \sum_e \mathbf{v}_e \mathbf{v}_e$

Does this work?

Application of Matrix Chernoff Bound by Rudelson (1999)

$$1 - \varepsilon \prec \lambda \left(\sum_e s_e \mathbf{v}_e \mathbf{v}_e^\mathsf{T} \right) \prec 1 + \varepsilon$$

finer bounds now available

Ridge Spectral Graph Sparsification: Intuition

Definition

γ-effective resistance: $r_e(\gamma) = \mathbf{b}_e^\mathsf{T}(\mathbf{L}_G + \gamma\mathbf{I})^{-1}\mathbf{b}_e$

Effective dim.: $\mathrm{d}_{\text{eff}}(\gamma) = \sum_e r_e(\gamma) = \sum_{i=1}^{n} \frac{\lambda_i(\mathbf{L}_G)}{\lambda_i(\mathbf{L}_G)+\gamma} \leq n$

Ridge Spectral Graph Sparsification: Intuition

Definition

γ-**effective resistance:** $r_e(\gamma) = \mathbf{b}_e^\mathsf{T}(\mathbf{L}_G + \gamma\mathbf{I})^{-1}\mathbf{b}_e$

Effective dim.: $\mathrm{d}_{\text{eff}}(\gamma) = \sum_e r_e(\gamma) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \frac{\lambda_\mathbf{i}(\mathbf{L_G})}{\lambda_\mathbf{i}(\mathbf{L_G})+\gamma} \leq n$

Interpretation as inverse of alternative paths (mostly) lost

Ridge Spectral Graph Sparsification: Intuition

Definition

γ**-effective resistance:** $r_e(\gamma) = \mathbf{b}_e^{\mathsf{T}}(\mathbf{L}_G + \gamma \mathbf{I})^{-1}\mathbf{b}_e$

Effective dim.: $\mathrm{d}_{\text{eff}}(\gamma) = \sum_e r_e(\gamma) = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \frac{\lambda_{\mathbf{i}}(\mathbf{L}_{\mathbf{G}})}{\lambda_{\mathbf{i}}(\mathbf{L}_{\mathbf{G}})+\gamma} \leq n$

Interpretation as inverse of alternative paths (mostly) lost

Most existing graph algorithms inapplicable Kyng et al., 2017

Most existing RLA algorithms too slow Cohen et al., 2017

Ridge Spectral Graph Sparsification: Intuition

Definition

γ-effective resistance: $r_e(\gamma) = \mathbf{b}_e^\mathsf{T}(\mathbf{L}_G + \gamma\mathbf{I})^{-1}\mathbf{b}_e$

Effective dim.: $\mathrm{d}_{\text{eff}}(\gamma) = \sum_e r_e(\gamma) = \sum_{\mathbf{i}=1}^{\mathbf{n}} \frac{\lambda_\mathbf{i}(\mathbf{L_G})}{\lambda_\mathbf{i}(\mathbf{L_G})+\gamma} \leq n$

Interpretation as inverse of alternative paths (mostly) lost

Most existing graph algorithms inapplicable Kyng et al., 2017

Most existing RLA algorithms too slow Cohen et al., 2017

Adapt SOA algorithm for kernel matrix approximation

SQUEAK, Calandriello et al., 2017

Spectral Graph Sparsification

We want to make this algorithm fast.

Spectral Graph Sparsification

We want to make this algorithm fast.

How can we compute the effective resistances?

Spectral Graph Sparsification

We want to make this algorithm fast.

How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^\mathsf{T}\widehat{\mathbf{x}}$

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^\mathsf{T}\widehat{\mathbf{x}}$

Gaussian Elimination $\qquad \mathcal{O}(n^3)$

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^{\mathsf{T}}\widehat{\mathbf{x}}$

Gaussian Elimination	$\mathcal{O}(n^3)$
Fast Matrix Multiplication	$\mathcal{O}(n^{2.37})$

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^\mathsf{T}\widehat{\mathbf{x}}$

Gaussian Elimination	$\mathcal{O}(n^3)$
Fast Matrix Multiplication	$\mathcal{O}(n^{2.37})$
Spielman & Teng (2004)	$\mathcal{O}(m\log^{30} n)$

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^\mathsf{T}\widehat{\mathbf{x}}$

Gaussian Elimination	$\mathcal{O}(n^3)$
Fast Matrix Multiplication	$\mathcal{O}(n^{2.37})$
Spielman & Teng (2004)	$\mathcal{O}(m\log^{30} n)$
Koutis, Miller, and Peng (2010)	$\mathcal{O}(m\log n)$

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^\mathsf{T}\widehat{\mathbf{x}}$

Gaussian Elimination	$\mathcal{O}(n^3)$
Fast Matrix Multiplication	$\mathcal{O}(n^{2.37})$
Spielman & Teng (2004)	$\mathcal{O}(m \log^{30} n)$
Koutis, Miller, and Peng (2010)	$\mathcal{O}(m \log n)$

- Fast solvers for SDD systems:

Spectral Graph Sparsification

We want to make this algorithm fast.
How can we compute the effective resistances?

Solve a linear system $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_G\mathbf{x} - \mathbf{b}_e\|$ and then $R_{\text{eff}} = \mathbf{b}_e^{\mathsf{T}}\widehat{\mathbf{x}}$

Gaussian Elimination	$\mathcal{O}(n^3)$
Fast Matrix Multiplication	$\mathcal{O}(n^{2.37})$
Spielman & Teng (2004)	$\mathcal{O}(m\log^{30} n)$
Koutis, Miller, and Peng (2010)	$\mathcal{O}(m\log n)$

- Fast solvers for SDD systems:
  - ↳ use sparsification internally

all the way until you hit the turtles

Spectral Graph Sparsification

Chicken and egg problem

We need R_{eff} to compute a sparsifier H ↰
↳ We need a sparsifier H to compute R_{eff}

Spectral Graph Sparsification

Chicken and egg problem

We need R_{eff} to compute a sparsifier H ↰
↳ We need a sparsifier H to compute R_{eff}

Sampling according to approximate effective resistances
$R_{\text{eff}} \leq \widetilde{R}_{\text{eff}} \leq \alpha R_{\text{eff}}$ give approximate sparsifier $\mathbf{L}_G \preceq \mathbf{L}_H \preceq \alpha\kappa\mathbf{L}_G$

Spectral Graph Sparsification

Chicken and egg problem

We need R_{eff} to compute a sparsifier H ↰
↳ We need a sparsifier H to compute R_{eff}

Sampling according to approximate effective resistances $R_{\text{eff}} \leq \widetilde{R}_{\text{eff}} \leq \alpha R_{\text{eff}}$ give approximate sparsifier $\mathbf{L}_G \preceq \mathbf{L}_H \preceq \alpha\kappa\mathbf{L}_G$

Start with poor approximation $\widetilde{R}_{\text{eff}} \sim 1/n$ and poor sparsifier $n\mathbf{I}$.

Spectral Graph Sparsification

Chicken and egg problem

We need R_{eff} to compute a sparsifier H ↰
↳ We need a sparsifier H to compute R_{eff}

Sampling according to approximate effective resistances $R_{\text{eff}} \leq \widetilde{R}_{\text{eff}} \leq \alpha R_{\text{eff}}$ give approximate sparsifier $\mathbf{L}_G \preceq \mathbf{L}_H \preceq \alpha\kappa\mathbf{L}_G$

Start with poor approximation $\widetilde{R}_{\text{eff}} \sim 1/n$ and poor sparsifier $n\mathbf{I}$.

Use $\widetilde{R}_{\text{eff}}$ to compute an improved approximate sparsifier H ↰
↳ Use the sparsifier H to compute improved approximate $\widetilde{R}_{\text{eff}}$

Spectral Graph Sparsification

Chicken and egg problem

We need R_{eff} to compute a sparsifier H ↰
↳ We need a sparsifier H to compute R_{eff}

Sampling according to approximate effective resistances $R_{\text{eff}} \leq \widetilde{R}_{\text{eff}} \leq \alpha R_{\text{eff}}$ give approximate sparsifier $\mathbf{L}_G \preceq \mathbf{L}_H \preceq \alpha\kappa\mathbf{L}_G$

Start with poor approximation $\widetilde{R}_{\text{eff}} \sim 1/n$ and poor sparsifier $n\mathbf{I}$.

Use $\widetilde{R}_{\text{eff}}$ to compute an improved approximate sparsifier H ↰
↳ Use the sparsifier H to compute improved approximate $\widetilde{R}_{\text{eff}}$

Computing $\widetilde{R}_{\text{eff}}$ using the sparsifier is fast ($m = \boldsymbol{O}(n\log(n))$), and not too many iterations are necessary.

What can I use sparsifiers for?

- Graph linear systems: minimum cut, maximum flow, Laplacian regression, SSL

But what if my problems have no use for spectral guarantees?

Or if my boss does not trust approximation methods

What can I use sparsifiers for?

- Graph linear systems: minimum cut, maximum flow, Laplacian regression, SSL
- More in general, solving Strongly Diagonally Dominant (SDD) linear systems

But what if my problems have no use for spectral guarantees?

Or if my boss does not trust approximation methods

What can I use sparsifiers for?

- Graph linear systems: minimum cut, maximum flow, Laplacian regression, SSL
- More in general, solving Strongly Diagonally Dominant (SDD) linear systems
  ↳ electric circuit, fluid equations, finite elements methods

But what if my problems have no use for spectral guarantees?

Or if my boss does not trust approximation methods

What can I use sparsifiers for?

- Graph linear systems: minimum cut, maximum flow, Laplacian regression, SSL
- More in general, solving Strongly Diagonally Dominant (SDD) linear systems

  ↳ electric circuit, fluid equations, finite elements methods
- Various embeddings: k-means, spectral clustering.

But what if my problems have no use for spectral guarantees?

Or if my boss does not trust approximation methods

Michal Valko

michal.valko@inria.fr

Inria & ENS Paris-Saclay, MVA

https://misovalko.github.io/mva-ml-graphs.html