Other applications of spectral methods

Hugues Déprés - Étienne Vareille

3 décembre 2019

Plan

1. Electrical circuits
   - Definitions
   - Computations
2. Applications
   - Cuts and Flows
   - The Matrix Exponential and Random Walks

Reminder : Laplacians matrices

Definition

A laplacian matrix L of an undirected graph can be written $L = D - A$ where D is the degree matrix of the graph and A the adjacency matrix.

Proposition

Let B be the incidence matrix (of dimension $|E| \times |V|$) of any orientation of an undirected graph, it's laplacian matrix L is equal to $B^\top B$.

Electrical circuits

Definition

To each graph connected G, we associate the vectors :

- the current in each edge $i \in \mathbb{R}^m$
- the voltage in each vertex $v \in \mathbb{R}^n$ (up to a constant)
- the external current in each vertex $c_{\text{ext}} \in \mathbb{R}^n$

They verify the followings relations :

- Kirchoff's law : $B^\top i = c_{\text{ext}}$.
- Ohm's law : $Bv = i$
- Steady state $\langle c_{\text{ext}}, 1 \rangle = 0$

Electrical circuits

Definition

To each graph connected G, we associate the vectors :

- the current in each edge $i \in \mathbb{R}^m$
- the voltage in each vertex $v \in \mathbb{R}^n$ (up to a constant)
- the external current in each vertex $c_{\text{ext}} \in \mathbb{R}^n$

They verify the followings relations :

- Kirchoff's law : $B^\top i = c_{\text{ext}}$.
- Ohm's law : $Bv = i$
- Steady state $\langle c_{\text{ext}}, 1\rangle = 0$

Proposition

We have $B^\top Bv = Lv = c_{\text{ext}}$

Electrical circuits

Definition

Let G be a graph, L its laplacian, and $g = (i, j)$ be an edge, the effective resistance $R_{\text{eff}}(e)$ is defined by
$R_{\text{eff}}(g) = (e_i - e_j)^\top L^+(e_i - e_j) = b_g^\top L^+ b_g$.

Electrical circuits

Definition

Let G be a graph, L its laplacian, and $g = (i, j)$ be an edge, the effective resistance $R_{\text{eff}}(e)$ is defined by
$R_{\text{eff}}(g) = (e_i - e_j)^\top L^+ (e_i - e_j) = b_g^\top L^+ b_g$.

Definition

We define $\Pi(f, g) = b_f^\top L^+ b_g$, such that $\Pi = BL^+ B^\top$

Properties of Π

Proposition

Π is symmetric and is a projection matrix, i.e. $\Pi^2 = \Pi$.

Properties of Π

Proposition

Π is symmetric and is a projection matrix, i.e. $\Pi^2 = \Pi$.

Proposition

The eigenvalues of Π are all 0 or 1.

Energy of an electrical flow

We assume that we input one unit of current at s and output one at t, then the flow is defined by $f^* = BL^+(e_s - e_t)$.

Definition

The energy of a flow is defined to be the sum of the squares of the flow on each edge.

Proposition

We have $E(f^*) = (e_s - e_t)^\top L^+(e_s - e_t)$.

Energy of an electrical flow

We assume that we input one unit of current at s and output one at t, then the flow is defined by $f^* = BL^+(e_s - e_t)$.

Definition

The energy of a flow is defined to be the sum of the squares of the flow on each edge.

Proposition

We have $E(f^*) = (e_s - e_t)^\top L^+(e_s - e_t)$.

Proposition

We have that f^* is the s,t-flow that minimize the energy consumption.

Reminder

There is a $\tilde{O}(m)$ algorithm LSolve, wich given L, b, ϵ returns x satisfying $||x - L^+b||_L \leq \varepsilon||L^+b||_L$.

Theorem

There is a $\tilde{O}(m\log(\frac{1}{\varepsilon}))$ algorithm which for a graph G, an epsilon $\epsilon > 0$, and vertices s, t finds $\tilde{v} \in \mathbb{R}^V$ and $\tilde{f} \in \mathbb{R}^E$ such that :

- $||\tilde{v} - v||_\infty \leq \varepsilon$
- $||\tilde{f} - f||_\infty \leq \varepsilon$
- $|\sum_e f_e^2 - \tilde{f}_e^2| \leq \varepsilon$

Computing effective resistance

Definition

We denote by w_i the vector BL^+e_i, in such a way that $R_e = ||BL^+(e_i - e_j)||^2 = ||w_i - w_j||^2$.

Theorem

There exist a constant C such that, when $k \geq \frac{C\log(n)}{\epsilon^2}$, and A is a matrix of dimension $k \times m$ with coefficients choosen randomly among $\{\frac{-1}{\sqrt{k}}, \frac{1}{\sqrt{k}}\}$, that with probability $1 - \frac{1}{n}$, forall $1 \leq i, j \leq n$ we have $(1-\epsilon)||w_i - w_j||^2 \leq ||Aw_i - A - w_j||^2 \leq (1+\epsilon)||w_i - w_j||^2$.

Computing effective resistance

Algorithm

- (1) Let A be a $k \times n$ matrix with random coefficients $\pm\frac{1}{\sqrt{k}}$ and $k = O(\frac{\log(n)}{\varepsilon^2})$.
- (2) Compute $Y = AB$. This takes $2m \times O(\frac{\log(n)}{\varepsilon^2}) + m = \tilde{O}(\frac{m}{\varepsilon^2})$ times since B has $2m$ entries.
- (3) Let $y_i^\top$, for $1 \leq i \leq k$ be the rows of Y, and compute $\tilde{z}_i = LSOLVE(L, y_i, \delta)$.

Theorem

There is an $\tilde{O}(\frac{m}{\varepsilon^2})$ time algorithm wich computes an $O(\frac{\log(n)}{\varepsilon^2}) \times n$ matrix $\tilde{Z}$ such that with probability at least $1 - \frac{1}{n}$:
$\forall i, j \in V, (1-\varepsilon)R_{i,j} \leq ||\tilde{Z}(e_i - e_j)||^2 \leq (1+\varepsilon)R_{i,j}$

Plan

1. Electrical circuits
   - Definitions
   - Computations
2. Applications
   - Cuts and Flows
   - The Matrix Exponential and Random Walks

Goal of the subsection

- Solve max-flow and min cut to electrical analogue.
- Find an $1 - \epsilon$ approximation of max flow in $\tilde{O}(m^{\frac{3}{2}} poly(\frac{1}{\epsilon}))$ time.

Reminder : combinatorial flows

Definition of combinatorial flow

For an undirected graph $G = (V, E)$, a source $s \in V$ and a sink $t \in V$, with edge capacity $c_e \geq 0$ for each edge, $\boldsymbol{f} = (f_e)_E$ is a flow :

- $\forall e \in E, |f_e| \leq c_e$
- $\forall v \in V - \{s, t\}, \boldsymbol{f}^T B \boldsymbol{e}_v = 0$
- $\boldsymbol{f}^T B \boldsymbol{e}_s + \boldsymbol{f}^T B \boldsymbol{e}_t = 0$

where B is the incidence matrix of G where all edges are oriented arbitrarily.

Max-flow problem

Find $\boldsymbol{f}$ that maximizes $|\boldsymbol{f}^T B \boldsymbol{e}_s|$.

Combinatorial versus Electrical Flows

We have defined the electrical flow from s to t with current F as :

$$\boldsymbol{f} = R^{-1} B L^{+} F(\boldsymbol{e}_s - \boldsymbol{e}_t)$$

This can be computed in time $\tilde{O}(m \log \frac{1}{\epsilon})$ time for precision ϵ.

Energy of flows

Let $\boldsymbol{f}^* = R^{-1} B L^{+}(\boldsymbol{e}_s - \boldsymbol{e}_t)$. For any combinatorial flow $\boldsymbol{g}$ from s to t such that $\forall e, |g_e| \leq 1$,

$$E_{\boldsymbol{r}}(\boldsymbol{f}^*) = \sum_{e \in E} r_e (f_e^*)^2 \leq \sum_{e \in E} r_e g_e^2 \leq \sum_{e \in E} r_e$$

Algorithm ELECFLOW

Algorithm 12.1 ELECFLOW

Input: $G(V, E)$, source s, sink t, a target flow value F and $0 < \varepsilon < 1$

Output: Either an s,t-flow $\mathbf{f}$ of value at least $(1 - O(\varepsilon))F$ or FAIL indicating that $F > F^\star$

```
w_e^0 ← 1 for all e ∈ E
ρ ← 2 sqrt(m/ε)
T ← ρ log m / ε^2
for t = 0 → T − 1 do
    ∀e ∈ E, r_e^t ← w_e^t + (ε/3m) Σ_e w_e^t
    f^t  =def  R_t^{-1} B L_t^+ F (e_s − e_t)
    if E_{r^t}(f^t) > (1 + ε/3) Σ_e r_e^t then
        return FAIL
    else
        ∀e ∈ E, w_e^{t+1} ← w_e^t (1 + ε|f_e^t| / ρ)
    end if
end for
return f  =def  ((1−ε)/(1+2ε)) · (1/T) · Σ_{t=0}^{T−1} f^t
```

$w_e^0 \leftarrow 1$ for all $e \in E$

$\rho \leftarrow 2\sqrt{\frac{m}{\varepsilon}}$

$T \leftarrow \frac{\rho \log m}{\varepsilon^2}$

for $t = 0 \to T - 1$ **do**

  $\forall e \in E,\ r_e^t \leftarrow w_e^t + \frac{\varepsilon}{3m} \sum_e w_e^t$

  $\mathbf{f}^t \stackrel{\text{def}}{=} R_t^{-1} B L_t^+ F(\mathbf{e}_s - \mathbf{e}_t)$

  if $E_{\mathbf{r}^t}(\mathbf{f}^t) > (1 + \frac{\varepsilon}{3}) \sum_e r_e^t$ **then**

    return FAIL

  else

    $\forall e \in E,\ w_e^{t+1} \leftarrow w_e^t (1 + \frac{\varepsilon |f_e^t|}{\rho})$

  end if

end for

return $\mathbf{f} \stackrel{\text{def}}{=} \frac{(1-\varepsilon)}{(1+2\varepsilon)} \cdot \frac{1}{T} \cdot \sum_{t=0}^{T-1} \mathbf{f}^t$

Analysis of ELECFLOW

Let F^* be the maximal s-t-flow value for a graph G.
We need to guarantee :

- When algorithm fails, $F > F^*$.
- When it outputs $\boldsymbol{f}$, the flow value from s to t is at least $(1 - O(\epsilon))F$.
- Capacity constraints are respected.

The first two points are easy to prove.

Analysis of ELECFLOW

Proof of the third point :

Lemma

If $E_{\boldsymbol{r}^t}(\boldsymbol{f}^t) \leq (1+\frac{\epsilon}{3})\sum_e r_e^t$ then

- $\max_e |f_e^t| \leq 2\sqrt{\frac{m}{\epsilon}}$
- $\sum w_e^t |f_e^t| \leq (1+\epsilon)\sum w_e^t$

Lemma

$$\sum_e w_e^t \leq m \exp\left(\frac{\epsilon(1-\epsilon)T}{\rho}\right)$$

Analysis of ELECFLOW

Lemma

For $T \geq \frac{\rho \log m}{\epsilon}$, the capacity constraint is respected.

Theorem

If F^* is the maximal flow value of graph G, ELECFLOW outputs a flow of value at least $(1-\epsilon)F^*$ in time $\tilde{O}(m^{\frac{3}{2}} poly(\frac{1}{\epsilon}))$

Generalization of ELECFLOW

For a general capacity vector $\boldsymbol{c}$, the algorithm can be adapted by :

- Replace r_e^t update rule by $r_e^t = \frac{1}{c_e^2}(w_e^t + \frac{\epsilon}{3m}\sum_e w_e^t)$
- Replace all mention of $|f_e^t|$ by $\frac{|f_e^t|}{c_e^2}$

We can also adapt ELECFLOW to obtain an approximation min-cut algorithm.

Goal of the subsection

- For L the Laplacian of graph and a vector $\boldsymbol{v}$, compute $\exp(-tL)\boldsymbol{v}$.
- There is an approximation algorithm with ϵ error running in $O(m \log t \log \frac{1}{\epsilon})$ time.
- Application to continuous random walks

Reminder : the Matrix Exponential

Definition

Let A be a symetric $n \times n$ matrix. The matrix exponential of A is defined as :

$$\exp(A) = \sum_{i=0}^{\infty} \frac{A^i}{i!}$$

Remark : If $A = \sum \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ spectral decomposition, then

$$\exp(A) = \sum_{i=0}^{n} \exp(\lambda_i) \boldsymbol{u}_i \boldsymbol{u}_i^T$$

First approximation : truncate the exponential

We can use $\boldsymbol{u} = \sum_{i=0}^{T} \frac{(-1)^i L^i}{i!} \boldsymbol{v}$ as an approximation for $\exp(-L)\boldsymbol{v}$.
We can compute $\boldsymbol{u}$ in $O(mT)$ time.

Theorem (admitted)

For $T \sim ||L|| + \log \frac{1}{\epsilon}$, we have $||\boldsymbol{u} - \exp(-L)\boldsymbol{v}|| \leq \epsilon ||\boldsymbol{v}||$

Problem : dependency in $||L||$.

Rational approximations to the exponential

Bound over the real exponential

There exists constants $c \geq 1$ and k_0 such that, for any integer $k \geq k_0$, there exists a polynomial $P_k(x)$ of degree k such that

$$\sup_{x \in [0,\infty)} \left| \exp(-x) - P_k\left(\frac{1}{1+x/k}\right) \right| \leq ck \times 2^{-k}$$

Corollary

There exists constants $c \geq 1$ and k_0 such that, for any integer $k \geq k_0$, there exists a polynomial $P_k(x)$ of degree k such that for any graph Laplacian L and vector $\boldsymbol{v}$,

$$\left|\left|\exp(-L)\boldsymbol{v} - P_k((I + L/k)^{+})\boldsymbol{v}\right|\right| \leq O(k2^{-k})||\boldsymbol{v}||$$

Solver for SDD matrices

Definition : Symmetric Diagonally Dominant matrix

A matrix A is SDD iff it is symmetric and for all i, $A_{ii} \geq \sum_{j,j\neq i} |A_{ij}|$

Laplacian solver *LSOLVE* can be adapted to SDD matrices.

Theorem (admitted)

Given an $n \times n$ SDD matrix A with m nonzero entries, a vector $\boldsymbol{b}$, and an error parameter $\epsilon > 0$, we can obtain a vector $\boldsymbol{u}$ such that$||\boldsymbol{u} - A^{+}b||_A \leq \epsilon ||A^{+}b||_A$.
Time required : $O(m \log n \log(1/(\epsilon ||A^{+}||)))$

Main result

Theorem

There is an algorithm that, given the graph Laplacian L of a weighted graph with n vertices and m edges, a vector $\boldsymbol{v}$, and a parameter $0 < \delta \leq 1$, outputs a vector $\boldsymbol{u}$ such that

$$||\exp(-L)\boldsymbol{v} - \boldsymbol{u}|| \leq \delta||\boldsymbol{v}||$$

in time $O((m+n)\log(1+||L||)\text{polylog}\frac{1}{\delta})$.

Application to continuous random walks

Discrete random walk process : At each step, we transition to a neighbor of the current vertex.
If we are on vertex v of degree $\Delta(v)$ then we transition to each neighbor with probability $\frac{1}{\Delta(v)}$.

Transition matrix

If initial distribution is $\boldsymbol{v}$, at next step the distribution will be $W\boldsymbol{v}$ with $W = AD^{-1}$.
Iterating, from step 0 to step t, the transition matrix is W^t.

Application to continuous random walks

Continuous time random walk

At time t, from initial distribution $\boldsymbol{v}$, we reach distribution $\tilde{W}(t)\boldsymbol{v}$, with

$$\tilde{W}(t) = \exp(-t(I - W))$$

Remark : $\tilde{W}(t) = \exp(-t)\sum_{i=0}^{\infty} \frac{t^i}{i!} W^i$
Equivalent to a discrete time random walk where the number of steps follow a Poisson law.

Application to continuous random walks

W as normalized laplacian

$W = D^{\frac{1}{2}}(I - \mathcal{L})D^{-\frac{1}{2}}$ with $\mathcal{L}$ the normalized laplacian of G.
Thus, $\tilde{W}(t) = D^{\frac{1}{2}} \exp(-t\mathcal{L})D^{-\frac{1}{2}}$.

Consequence :
We can use laplacian exponentiation to compute an approximation of $\tilde{W}(t)\boldsymbol{v}$ in time $O(m \log(1+t) \text{polylog}(\frac{1}{\delta}))$.

Application to continuous random walks

Theorem : Approximation bound

There is an algorithm that, given an undirected graph G with m edges, a vector $\boldsymbol{v}$,a time $t \geq 0$,and a $\delta > 0$,outputs a vector $\boldsymbol{u}$ such that

$$||\tilde{W}(t)\boldsymbol{v} - \boldsymbol{u}|| \leq \delta \sqrt{\frac{d_{\max}}{d_{\min}}} ||\boldsymbol{v}||$$

. Time taken : $O(m \log(1 + t) \text{polylog}(\frac{1}{\delta}))$. Here, $d_{\max}$is the largest degree of G and $d_{\min}$ the smallest.