Todo list

Contents

BadGer RamPaGe: Multi-Dimensional Balanced Graph Partitioning via Gradient Descent

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

We give fast and scalable practical algorithms for the problem of partitioining large graphs into components of roughly the same size/weight according to multiple user-specified weight functions. This problem, referred to as *multi-dimensional balanced graph partitioning* (see Section 2 for formal definitions) arises in critical infrastructure applications which involve storage and processing of large graphs, including social networks. High-quality partitions help optimize load balancing in query processing, etc. While a large body of work exists offering practical solutions for the one-dimensional version of the problem [KK95, DGRW12, UB13, TGRV14, ABM16, DKK+16, MLLS17, KKP+17] (see also a recent survey [BMS+16]), as well as on theoretical foundations of graph partitioning [KNS09, AFK+14, MM14], literature on principled and scalable approaches for the multi-dimensional case is quite sparse []. In particular, if the weight functions are unrelated to each other, one can easily construct examples when no feasible solution exists that satisfies all balance constraints even for two weight functions.

elaborate

Talk about why multi-dimensional is particularly important vs. one-dimensional

cite a bunch of theory papers

Say that in general no feasible solution might exist at all.

Our contributions Let $G(V, E)$ be an n-vertex graph whose adjacency matrix is A. We introduce a family of algorithms for the multi-dimensional balanced graph partitioning problem by using the *projected gradient descent method* on a standard relaxation which involves maximizing an n-dimensional non-convex quadratic function $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$ subject to a constraint $\mathbf{x} \in K$ for some convex body K defined by the weight functions [1]. See Section 3 for the exact description of the relaxation.

Maybe need a citation here to justify this is standard

While applying projected gradient descent to solve non-convex optimization problems subject to convex constraints is a well-studied approach in non-linear optimization [Ber99] (Section 2.3) and machine learning [JK17] (Section 6.6), one has to overcome several technical challenges to make it applicable to the multi-dimensional graph partitoning problem: 1) projection step is computationally expensive, 2) abundance of saddle points slows down convergence.

discuss how constrained is different from unconstrained

We show how to address the first challenge by designing ultra-efficient projection step algorithms tailored to the standard non-convex relaxation of the multi-dimensional balanced graph partitioning problem. Computationally the problem of finding the closest point in the For balance according to one weight function our projection algorithm runs in time $O(n)$ where n is the number of vertices

[1] While second-order methods could potentially give better performance in terms of partition quality, due to the large scale of our instances such methods are infeasible.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

in the graph. For two weight functions we show how to implement projection in time $O(n \log^2 n)$. For k weight functions the time is .

TBD

In order to address the second challenge we use small perturbations to each intermediate point, where the perturbation vectors are sampled from a scaled n-dimensional Gaussian distribution. We refer to this algorithm as BADGER RAMPAGE (Algorithm 1). We show how the magnitude of Gaussian noise affects convergence properties of BADGER RAMPAGE by helping it escape from saddle points [] .

cite some literature on randomized PGD

cite literature on escaping from saddle points using SGD, etc.

Our experimental results show that BADGER RAMPAGE can be scaled to graphs with billions of vertices and hundreds of billions of edges. BADGER RAMPAGE outperforms... .

write this

Previous work

- Discuss all papers on balanced graph partitioning again [KK95, DGRW12, UB13, TGRV14, ABM16, DKK+16, MLLS17, KKP+17], survey [BMS+16].
- Discuss all the standard non-convex optimization subject to convex constraints (CNOPT) literature again [Ber99] (Section 2.3), [JK17] (Section 6.6).
- Discuss recent papers which use first-order methods for CNOPT and show how to escape saddle points in general and specific situations. Cite noisy SGD [GHJY15], some polynomial time algortihm which converges to third-order local optimum [AG16], matrix completion [GLM16], second-order methods [SQW15].
- Discuss papers which use PGD for constrained non-convex optimization problems arising from graph partitioning [LRS+10]..

  Find more papers to cite here

- Discuss why we can't use off the shelf QP solvers, like OSQP from Steven Boyd and others (they wouldn't scale to billions of vertices) [SBG+17].

  seems like they are only looking at the convex case, their number of non-zeros is at most 10^8 it seems

Our techniques

2 Preliminaries

We study multi-dimensional graph partitioning problems. The basic one-dimensional unweighted graph partitioning problem is as follows:

Definition 2.1 ($(1 \pm \epsilon)$-BALANCED k-PARTITION)**.** *Given an input graph $G(V, E)$, an integer k and a parameter $\epsilon > 0$ the goal is to find a partition of the vertex set V into k sets $V_1, \ldots, V_k$ such that $|V_i| = \frac{(1\pm\epsilon)|V|}{k}$ for all $i \in [k]$. Among all such partitions the goal is to find one that maximizes the number of edges whose both endpoints are contained within some part of the partition.*

The more general weighted d-dimensional version is defined by a collection of d weight functions $w_1, \ldots, w_d$ where each $w_i \colon V \to \mathbb{R}^+$ is a real-valued weight function. For a set $S \subseteq V$ we use notation $w_i(S) \equiv \sum_{v \in S} w_i(v)$. For example, the unweighted case above corresponds to $d = 1$ and $w_1(v) = 1$ for all $v \in V$.

Definition 2.2 (MULTI-DIMENSIONAL WEIGHTED $(1 \pm \epsilon)$-BALANCED k-PARTITION)**.** *Given an input graph $G(V, E)$, an integer k and a parameter $\epsilon > 0$ the goal is to find a partition of the vertex set V into k sets $V_1, \ldots, V_k$ such that for each $j \in [d]$ it holds that $w_j(V_i) = \frac{(1\pm\epsilon)w_j(V)}{k}$ for all $i \in [k]$. Among all such partitions the goal is to find one that maximizes the number of edges whose both endpoints are contained within some part of the partition.*

3 BADGER RAMPAGE: Randomized Projected Gradient Descent Algorithm

The standard integer quadratic program for the weighted balanced graph 2-partitioning problem is:

$$\begin{aligned}
&\text{Maximize:} && \frac{1}{2}\sum_{(i,j)\in E}(x_i x_j + 1) && \\
&\text{Subject to:} && \left|\sum_{j=1}^{n} w_i(j)x_j\right| \leq \epsilon \sum_{j=1}^{n} w_i(j) && \forall i \in [d] \\
& && x_i \in \{-1,1\} && \forall i \in V
\end{aligned}$$

Dropping the additive term in the objective and relaxing the integrality constraints we have a relaxation:

$$\begin{aligned}
&\text{Maximize:} && \mathbf{x}^T A\mathbf{x} && \\
&\text{Subject to:} && \left|\sum_{j=1}^{n} w_i(j)x_j\right| \leq \epsilon \sum_{j=1}^{n} w_i(j) && \forall i \in [d] \\
& && x_i \in [-1,1] && \forall i \in V
\end{aligned}$$

Denoting $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$ we have the gradient $\nabla f(\mathbf{x}) = A\mathbf{x}$ and Hessian $H_f = A$.

We propose the following general algorithm for the multi-dimensional weighted balanced graph partitioning problem. Let $\mathcal{B}_\infty = \{\mathbf{x} \in \mathbb{R}^n | \forall i \colon \mathbf{x}_i \in [-1,1]\}$. For $i \in [d]$ let $\mathcal{S}^i_\epsilon = \{\mathbf{x} \in \mathbb{R}^n || \sum_{j=1}^n w_i(j)\mathbf{x}_j| \leq \epsilon \sum_{j=1}^n w_i(j)\}$.

cite Recht et al. NIPS'11 paper suggested by Kostya

Algorithm 1: BADGER RAMPAGE (d-Dimensional Balanced Graph 2-Partitioning via Randomized Projected Gradient Descent)

input : Graph $G(V,E)$, integer k, real value $\epsilon \in [0,1]$, weight functions $w_1, \ldots, w_d \colon V \to \mathbb{R}^+$
output : $(1 \pm \epsilon)$-balanced partition w.r.t $w_1, \ldots, w_d$ of V into (V_1, V_2).

```
x_0 = 0, t = 0
do
    x'_t = x_t + η_t N(0,1)
    y_{t+1} = (I + γ_t A) x'_t
    x_{t+1} = arg min_{x∈K} ||y_{t+1} − x||_2, where K = B_∞ ∩_{j=1}^d S^j_ε
    t = t + 1
while ||x_t − x_{t+1}||_2 > θ;
```

4 Projection step

cite some paper which focus on efficincy of projection step in PGD

4.1 Approximate projection for $d = 1$

Formally, we have the following optimization problem. Given a fixed vector $\mathbf{y} \in \mathbb{R}^n$ and allowed imbalance ϵ:

$$\begin{aligned}
&\text{Minimize:} && f(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}\|_2^2 \\
&\text{Subject to:} && g_i = x_i^2 - 1 \leq 0 \\
& && h_+ = \sum_{i=1}^{n} x_i - \epsilon \leq 0 \\
& && h_- = -\sum_{i=1}^{n} x_i - \epsilon \leq 0
\end{aligned}$$

By KKT:

$$(\mathbf{y}-\mathbf{x}) = \sum_{i=1}^{n} \mu_i x_i \mathbf{e}_i + (\mu_+ - \mu_-) \sum_{i=1}^{n} \mathbf{e}_i.$$

I.e. for each coordinate we have $y_i - x_i = \mu_i x_i + \mu_+ - \mu_-$ where and $\mu_i, \mu_+, \mu_- \geq 0$. Complementary slackness gives $\mu_i(x_i^2 - 1) = 0$, i.e. for each i either $|x_i| = 1$ or $\mu_i = 0$. Moreover, we have additional slackness constraints:

- $\mu_+(\sum_{i=1}^n x_i - \epsilon) = 0$.
- $\mu_-(-\sum_{i=1}^n x_i - \epsilon) = 0$.

Consider 3 cases:

1. $\sum x_i = \epsilon$. In this case $\mu_- = 0$, and the first slackness constraint equals $y_i - x_i = \mu_i x_i + \mu_+$. Now this problem equals to exact projection, described in the previous section, with $\lambda = \mu_+$ and $\sum h_i = \sum y_i - \epsilon$.
2. $\sum x_i = -\epsilon$. This case is similar to the previous one with $\lambda = -\mu_-$ and $\sum h_i = \sum y_i + \epsilon$.
3. $\mu_+ = \mu_- = 0$. The first slackness constraint equals $y_i - x_i = \mu_i x_i$. This case is just projection on the hypercube, with restriction $\sum h_i \in (\sum y_i - \epsilon; \sum y_i + \epsilon)$.

In all cases we have $y_i - x_i = \mu_i x_i + \lambda$, and possible values of λ are disjoint. Since $\sum h_i$ is increasing, only one of this options can be satisfied. For example, if case one is satisfied, then for $\lambda < 0$ $\sum h_i(\lambda) = \sum y_i + \epsilon$. Since $\sum h_i(\lambda)$ is increasing, for $\sum h_i(0) \geq \sum h_i(\lambda) = \sum y_i + \epsilon$, and therefore $\sum h_i(0) \notin (\sum y_i - \epsilon; \sum y_i + \epsilon)$.

Therefore, to find lambda we can use the following algorithm:

- Try the case $\lambda = 0$. If $\sum h_i(0) \in (\sum y_i - \epsilon; \sum y_i + \epsilon)$, then return the corresponding $\mathbf{x}$. Otherwise select the part for binary search:
  - If $h(0) \geq \sum y_i + \epsilon$, search λ in $(-\infty; 0]$ with $\sum h_i(\lambda) = \sum y_i + \epsilon$.
  - If $h(0) \leq \sum y_i - \epsilon$, search λ in $[0; +\infty)$ with $\sum h_i(\lambda) = \sum y_i - \epsilon$.

4.2 Exact projection for $d = 2$

If we want to project on the intersection of two hyperplanes: $\sum_i w_i x_i = 0$ and $\sum_i w'_i x_i = 0$ then we can do this as follows. Using parameters λ and λ' as above we can still use the following algorithm setting $\gamma_i = \lambda w_i + \lambda' w'_i$:

change this to support approximate projection

1. $(y_i \geq 1 + \gamma_i)$. Set $x_i = 1$.
2. $(y_i \in (-1 + \gamma_i, 1 + \gamma_i))$. Set $x_i = y_i - \gamma_i$.
3. $(y_i \leq -1 + \gamma_i)$. Set $x_i = -1$.

Now we have two balance functions: $h = \sum w_i x_i$ and $h' = \sum w'_i x_i$. The change in h after projection is expressed as:

$$\sum_i w_i y_i - \sum_i w_i x_i = \sum_{i\colon y_i \geq 1+\gamma_i} w_i(y_i - 1) + \sum_{i\colon y_i \in (-1+\gamma_i, 1+\gamma_i)} w_i \gamma_i + \sum_{i\colon y_i \leq 1-\gamma_i} w_i(1 + y_i) = \sum_i h_i(\lambda, \lambda'),$$

where each h_i is the following function:

$$h_i(\lambda, \lambda') = \begin{cases} w_i(y_i - 1) & \text{if } \lambda w_i + \lambda' w'_i < y_i - 1 \\ w_i(\lambda w_i + \lambda' w'_i) & \text{if } \lambda w_i + \lambda' w'_i \in [y_i - 1, y_i + 1] \\ w_i(y_i + 1) & \text{if } \lambda w_i + \lambda' w'_i > y_i + 1 \end{cases}$$

Analogously, the difference between h' can be expressed as $\sum h'_i$, where

$$h'_i(\lambda, \lambda') = \begin{cases} w'_i(y_i - 1) & \text{if } \lambda w_i + \lambda' w'_i < y_i - 1 \\ w'_i(\lambda w_i + \lambda' w'_i) & \text{if } \lambda w_i + \lambda' w'_i \in [y_i - 1, y_i + 1] \\ w'_i(y_i + 1) & \text{if } \lambda w_i + \lambda' w'_i > y_i + 1 \end{cases}$$

Denote $h(\lambda, \lambda') = \sum h_i(\lambda, \lambda')$ and $h'(\lambda, \lambda') = \sum h'_i(\lambda, \lambda')$. We want to find λ and λ' such that $h(\lambda, \lambda') = \sum w_i y_i$ and $h'(\lambda, \lambda') = \sum w'_i y_i$. We will show that we can use binary search for λ with binary search for λ' inside.

Theorem 4.1. *Consider the situation when λ is fixed. Denote the* maximum *λ' such that $h(\lambda, \lambda') = \sum w_i y_i$ as $root(\lambda)$. The same way denote the* maximum *λ' such that $h'(\lambda, \lambda') = \sum w'_i y_i$ as $root'(\lambda)$. Then*

$$\lim_{\lambda \to +\infty} (root(\lambda) - root'(\lambda)) \cdot \lim_{\lambda \to -\infty} (root(\lambda) - root'(\lambda)) \leq 0$$

Our goal is to find λ such that $root(\lambda) - root'(\lambda) = 0$, meaning that there exist λ', such that $h(\lambda, \lambda') = \sum w_i y_i$ and $h'(\lambda, \lambda') = \sum w'_i y_i$. Since function $dif(\lambda) = root(\lambda) - root'(\lambda)$ is continuous (since it's piecewise-linear and continuous near borders??) and we can find points with different signs, we can find its root using binary search. (TODO: I think I know two pointers solution for this case).

Note that there can be several λ', corresponding to one λ. By selecting maximum value $root(\lambda)$ becomes unique. Now we will prove the theorem.

Proof. For the proof we will use a geometric approach. We consider a two-dimensional plane (λ, λ') and the following regions: $\lambda w_i + \lambda' w'_i$ for all i. We will show that when $\lambda \to +\infty$, $root(\lambda)$ form a line, lying in some region and parallel to its borders.

First, note that there are only finite number of intersections between regions. Non-empty intersections can be of two following types:

1. Unbounded intersections. Each region corresponds to an area between two parallel lines, two regions are *parallel* if their border lines are parallel. Then non-empty intersection of the regions is unbounded if all the regions are parallel.

2. Bounded intersection. If some of regions are not parallel, the intersection is bounded.

We consider the case when there are no unbounded intersections of more than one region. (TODO: consider another case.) By monotonicity and piecewise-linearity of h and h' $root$ and $root'$ are also piecewise-linear functions. Since region borders are lines, for large enough λ $root(\lambda)$ entirely lies in one region or between them (TODO: show this).

Consider function h. Sort all regions by its angle $k_i = -\frac{w'_i}{w_i}$. Change numeration of coordinates $\{ h_i \}$ in such way that k_i are decreasing. Then the following is true. When point (λ, λ') belongs to i-th region, for large enough λ:

- For all h_j where $j < i$ the third case is satisfied (since (λ, λ') is above this region), namely $\lambda w_i + \lambda' w'_i > y_i + 1$ and $h_i(\lambda, \lambda') = w_i(y_i + 1)$.

- For all h_j where $j > i$ the first case is satisfied (since (λ, λ') is below this region), namely $\lambda w_i + \lambda' w'_i < y_i - 1$ and $h_i(\lambda, \lambda') = w_i(y_i - 1)$.

- For i-th region itself we know that $\lambda w_i + \lambda' w'_i \in [y_i - 1; y_i + 1]$. Therefore, $\lambda w_i + \lambda' w'_i = y_i + c$, where $c \in [-1; 1]$. $h_i(\lambda, \lambda') = w_i(y_i + c)$

Computing $h(\lambda, \lambda')$ gives us

$$h(\lambda, \lambda') = \sum_{j<i} w_i(y_i + 1) + w_i(y_i + c) + \sum_{j>i} w_i(y_i - 1) =$$
$$= \sum_i w_i y_i + \sum_{j<i} w_i + w_i c - \sum_{j>i} w_i$$

In case when (λ, λ') lies between $(i-1)$-th and i-th regions, there is no $w_i c$ term and

$$h(\lambda, \lambda') = \sum_i w_i y_i + \sum_{j<i} w_i - \sum_{j \geq i} w_i$$

So, the value of h doesn't change between two regions, which also follows from h_i definition (it is constant on each side of i-th region). Also note that it matches the value of h on the borders of $(i-1)$-th and i-th region, because values of c are 1 and -1 respectively.

Now we can find $root(\lambda)$ when $\lambda \to \infty$. We need to find i and c such that

$$\sum_i w_i y_i = h(\lambda, \lambda') = \sum_i w_i y_i + \sum_{j<i} w_i + w_i c - \sum_{j>i} w_i \iff \sum_{j<i} w_i + w_i c - \sum_{j>i} w_i = 0$$

Note that since all w_i are non-negative, there is an only way to split such sum, but there can be two representations when $c \in \{-1; 1\}$ (i.e. when (λ, λ') is between regions). In such case the maximum value of λ' is achieved when $c = -1$ (note that the maximum value is achieved on the left border of the right region, which has greater index). Denote such i and c as i_+ and c_+ respectively. (TODO: handle the case when (λ, λ') is not between regions, but on the left or on the right of all of them.) (TODO: handle the case when some regions are horizontal or vertical.) (TODO: draw some pictures.)

Consider $root(\lambda)$ when $\lambda \to -\infty$. By the similar reasoning we achieve the following equation for i and c:

$$-\sum_{j<i} w_i + w_i c + \sum_{j>i} w_i = 0 \iff \sum_{j<i} w_i - w_i c - \sum_{j>i} w_i = 0$$

As can be seen, $i = i_+$ and $c = -c_+$ satisfy this equation. If $c_+ \in (-1; 1)$ then pair $(i_-, c_-) = (i_+, -c_+)$ is a solution, corresponding to the unique λ'. Otherwise $c_+ = -1 \iff -c_+ = 1$ and we should assign $i_- = i - 1$ and $c_- = -1$ (note that the maximum value is achieved on the left border of the right region, which has smaller index).

By the same reasoning for h' we have to solve the following equations

$$\sum_{j<i} w_i + w_i c - \sum_{j>i} w_i = 0$$
$$-\sum_{j<i} w_i + w_i c + \sum_{j>i} w_i = 0$$

Denote the solution of the first equation as (i'_+, c'_+). Then the solution of the second equation is

$$(i'_-, c'_-) \begin{cases} (i'_+, -c'_+) & \text{if } c'_+ \in (-1; 1) \\ (i'_+ - 1, -1) & \text{if } c'_+ = -1 \end{cases}$$

If $i_+ = i'_+$ and $c_+ = c'_+$) then the theorem is proved, since $\lim_{\lambda \to +\infty}(root(\lambda) - root'(\lambda)) = 0$. Otherwise, w.l.o.g. assume that $i_+ < i'_+$ or ($i_+ = i'_+$ and $c_+ < c'_+$)). In this case $\lim_{\lambda \to +\infty}(root(\lambda) - root'(\lambda)) < 0$, since less pair (i, c) corresponds to less λ' when $\lambda \to \infty$.

Our goal is to show that $\lim_{\lambda \to -\infty}(root(\lambda) - root'(\lambda)) \geq 0$, and namely that $i_- < i'_-$ or ($i_- = i'_-$ and $c_- \geq c'_-$). We need to consider the following cases:

- $i_+ < i'_+$ and $c'_+ = -1$. Then $i'_- = i'_+ - 1 \geq i_-$ and $c'_- = -1 \leq c_-$.
- $i_+ < i'_+$ and $c'_+ \in (-1; 1)$. Then $i'_- = i'_+ > i_-$.
- $i_+ = i'_+$ and $c'_+ = -1$. Impossible, since c_+ must be less than c'_+.
- $i_+ = i'_+$, $c_+ = -1$ and $c'_+ \in (-1; 1)$. Then $i'_- = i'_+ > i_+ - 1 = i_-$.
- $i_+ = i'_+$, $c_+, c'_+ \in (-1; 1)$. Then $i'_- = i_-$ and $c'_- < c_-$.

5 Towards convergence analysis

Lemma 5.1 (Bertsekas, Section 2.3.2). *If $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $0 < \gamma < 2/L$ then*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \geq \left(\frac{1}{\gamma} - \frac{L}{2}\right) \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2$$

Proof. We use the following Descent Lemma:

Proposition 5.2 (Descent Lemma). *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and let $\mathbf{x}$ and $\mathbf{y}$ be two vectors in $\mathbb{R}^n$. If for all $t \in [0,1]$ it holds that $\|\nabla f(\mathbf{x}+t\mathbf{y}) - \nabla f(\mathbf{x})\| \le Lt\|\mathbf{y}\|$ where L is a constant then:*

$$f(\mathbf{x}+\mathbf{y}) \ge f(\mathbf{x}) + \mathbf{y}^T\nabla f(\mathbf{x}) - \frac{L}{2}\|\mathbf{y}\|_2^2$$

Proof. Let t be a scalar and let $g(t) = f(\mathbf{x}+t\mathbf{y})$. By the chain rule $(dg/dt)(t) = \mathbf{y}^T\nabla f(\mathbf{x}+t\mathbf{y})$. Then:

$$\begin{aligned}
f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) &= g(1) - g(0)\\
&= \int_0^1 \frac{dg}{dt}(g)dt\\
&= \int_0^1 \mathbf{y}^T\nabla f(\mathbf{x}+t\mathbf{y})dt\\
&\ge \int_0^1 \mathbf{y}^T\nabla f(\mathbf{x})dt - \left|\int_0^1 \mathbf{y}^T(\nabla f(\mathbf{x}+t\mathbf{y}) - \nabla f(\mathbf{x}))dt\right|\\
&\ge \int_0^1 \mathbf{y}^T\nabla f(\mathbf{x})dt - \int_0^1 \|\mathbf{y}\|_2\|\nabla f(\mathbf{x}+t\mathbf{y}) - \nabla f(\mathbf{x})\|_2 dt\\
&\ge \mathbf{y}^T\nabla f(\mathbf{x}) - \|\mathbf{y}\|_2\int_0^1 Lt\|\mathbf{y}\|_2 dt\\
&= \mathbf{y}^T\nabla f(\mathbf{x}) - \frac{L}{2}\|\mathbf{y}\|_2^2
\end{aligned}$$

∎

By the Descent Lemma we have:

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \ge \nabla f(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t) - \frac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2.$$

Note that since $K = \mathcal{S}_0 \cap \mathcal{B}_\infty$ is a convex body and $\mathbf{x}_{k+1}$ is a projection of $\mathbf{y}_{t+1}$ on K for every $\mathbf{x} \in K$ it holds that:

$$(\mathbf{y}_{t+1} - \mathbf{x}_{t+1})(\mathbf{x} - \mathbf{x}_{t+1}) \le 0.$$

Applying this to $\mathbf{x} = \mathbf{x}_t$ and using the fact that $\mathbf{y}_{t+1} = \mathbf{x}_t + \gamma\nabla f(\mathbf{x}_t)$ we have:

$$(\mathbf{x}_t + \gamma\nabla f(\mathbf{x}_t) - \mathbf{x}_{t+1})(\mathbf{x}_t - \mathbf{x}_{t+1}) \le 0,$$

which implies that $\nabla f(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t) \ge \frac{1}{\gamma}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2$. Hence we have:

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \ge \left(\frac{1}{\gamma} - \frac{L}{2}\right)\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2$$

5.1 First step

Let's analyze the first step. We show the following theorem:

Theorem 5.3. *Let $\lambda_{max} = \max(|\lambda_1|, |\lambda_n|)$ then if $\gamma = 1/\lambda_{max}$ then $\mathbb{E}[f(\mathbf{x}_1)] \ge \frac{\eta^2|E|}{2\lambda_{max}}$ (assuming we don't have to round the coordinates).*

Fix this!

Consider three points $\mathbf{x}_0, \mathbf{x}_1$ and $\mathbf{y}_1$. These three points lie in a hyperplane H' which is orthogonal to the hyperplane $H = \{\mathbf{x}\colon \langle \mathbf{1}, \mathbf{x}\rangle = 0\}$, where $\mathbf{1} = (\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})$. Let $\mathbf{x}_0'$ be the projection of $\mathbf{x}_0$ on H. Then $\mathbf{x}_0'$ also lies in H'.

We have $\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2 = \|\mathbf{x}_0 - \mathbf{x}_0'\|_2^2 + \|\mathbf{x}_0' - \mathbf{x}_1\|_2^2$. Let $\mathbf{y}_1'$ be the projection of $\mathbf{x}_0$ on the line through $\mathbf{y}_1$ and $\mathbf{x}_1$. Then $\|\mathbf{x}_0 - \mathbf{y}_1\|_2^2 = \|\mathbf{x}_0 - \mathbf{y}_1'\|_2^2 + \|\mathbf{y}_1' - \mathbf{y}_1\|_2^2$. Since $\|\mathbf{x}_0 - \mathbf{y}_1'\|_2^2 = \|\mathbf{x}_0' - \mathbf{x}_1\|_2^2$ we have:

$$\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2 = \|\mathbf{x}_0 - \mathbf{x}_0'\|_2^2 + \|\mathbf{x}_0 - \mathbf{y}_1\|_2^2 - \|\mathbf{y}_1' - \mathbf{y}_1\|_2^2.$$

We now take expectations and make use of the Lemma 5.4 which is proved below:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2\right] &= \eta^2\left(1 + \gamma^2\|A\|_F^2 - \gamma^2\sum_{i=1}^{n}\lambda_i^2\langle \mathbf{1}, v_i\rangle^2\right)\\ &\geq \eta^2(1 + 2\gamma^2|E| - \gamma^2\max(\lambda_1^2, \lambda_n^2))\end{aligned}$$

By Lemma 5.1 using the fact that for our function $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$ we have $L \leq \max(|\lambda_1|, |\lambda_n|)$ we obtain:

$$f(\mathbf{x}_1) - f(\mathbf{x}_0) \geq \left(\frac{1}{\gamma} - \frac{\max(|\lambda_1|, |\lambda_n|)}{2}\right)\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2.$$

Setting $\gamma = 1/\max(|\lambda_1|, |\lambda_n|)$ and taking expectations we have:

$$\mathbb{E}[f(\mathbf{x}_1) - f(\mathbf{x}_0)] \geq \frac{\eta^2|E|}{2\max(|\lambda_1|, |\lambda_n|)}$$

Finally note that:

$$\mathbb{E}[f(\mathbf{x}_0)] = \mathbb{E}[\mathbf{x}_0 A\mathbf{x}_0] = \eta^2\sum_{i=1}^{n}\lambda_i = \eta^2 tr(A) = 0,$$

and hence the proof of the theorem follows.

It remains to prove Lemma 5.4.

Lemma 5.4. *If $A = \sum_{i=1}^{n}\lambda_i v_i v_i^T$ is the eigendecomposition of A where v_i's form an orthonormal basis then:*

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}_0'\|_2^2] &= \eta^2\\ \mathbb{E}[\|\mathbf{x}_0 - \mathbf{y}_1\|_2^2] &= \eta^2\gamma^2\|A\|_F^2\\ \mathbb{E}[\|\mathbf{y}_1' - \mathbf{y}_1\|_2^2] &= \eta^2\gamma^2\sum_{i=1}^{n}\lambda_i^2\langle \mathbf{1}, v_i\rangle^2\end{aligned}$$

Proof. We have $\mathbb{E}[\|\mathbf{x}_0 - \mathbf{x}_0'\|_2^2] = \mathbb{E}[\langle \mathbf{1}, \mathbf{x}_0\rangle^2] = \eta^2$, where the second equality follows by rotational symmetry of the Gaussian distribution.

We have:

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}_0 - \mathbf{y}_1\|_2^2] &= \mathbb{E}[\|\gamma A\mathbf{x}_0\|_2^2]\\ &= \mathbb{E}\left[\gamma^2\mathbf{x}_0^T A^2\mathbf{x}_0\right]\\ &= \mathbb{E}\left[\gamma^2\mathbf{x}_0^T\left(\sum_{i=1}^{n}\lambda_i^2 v_i v_i^T\right)\mathbf{x}_0\right]\\ &= \gamma^2\sum_{i=1}^{n}\lambda_i^2\mathbb{E}\left[\langle v_i, \mathbf{x}_0\rangle^2\right]\\ &= \gamma^2\eta^2\|A\|_F^2,\end{aligned}$$

where in the last equality we use the fact that since each v_i is a unit vector $\mathbb{E}[\langle v_i, \mathbf{x}_0\rangle] = \eta^2$ by the rotaional symmetry of the Gaussian distribution.

Finally, we have:

$$\begin{aligned}
\mathbb{E}[\|\mathbf{y}'_1 - \mathbf{y}_1\|_2^2] &= \mathbb{E}[\langle \mathbf{1}, \mathbf{y}_1 - \mathbf{x}_0\rangle^2] \\
&= \mathbb{E}[\langle \mathbf{1}, \gamma A \mathbf{x}_0\rangle^2] \\
&= \gamma^2 \mathbb{E}\left[\left(\mathbf{1}^T\left(\sum_{i=1}^n \lambda_i v_i v_i^T\right)\mathbf{x}_0\right)^2\right] \\
&= \gamma^2 \mathbb{E}\left[\left(\sum_{i=1}^n \lambda_i \langle \mathbf{1}, v_i\rangle \langle v_i, \mathbf{x}_0\rangle\right)^2\right] \\
&= \gamma^2\left(\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j \langle \mathbf{1}, v_i\rangle\langle \mathbf{1}, v_j\rangle \mathbb{E}\left[\langle v_i, \mathbf{x}_0\rangle\langle v_j, \mathbf{x}_0\rangle\right]\right)
\end{aligned}$$

Note that since v_i and v_j are orthogonal for $i \neq j$ we have $\mathbb{E}\left[\langle v_i, \mathbf{x}_0\rangle\langle v_j, \mathbf{x}_0\rangle\right] = 0$. For $i = j$ we have $\mathbb{E}[\langle v_i, \mathbf{x}_0\rangle^2] = \eta^2$ as before. Hence we have:

$$\mathbb{E}[\|\mathbf{y}'_1 - \mathbf{y}_1\|_2^2] = \eta^2\gamma^2 \sum_{i=1}^n \lambda_i^2 \langle \mathbf{1}, v_i\rangle^2$$

5.2 t-th step

We will assume that noise is added in every step, i.e. the algorithm at every step looks as follows:

1. Given input $\mathbf{x}_t$ from the pervious step let $\mathbf{x}'_t = \mathbf{x}_t + \eta N(0,1)$.
2. Let $\mathbf{y}_{t+1} = \mathbf{x}'_t + \gamma A \mathbf{x}'_t$
3. Set $\mathbf{x}_{t+1} = \arg min_{\mathbf{x} \in \mathcal{S}_0 \cap \mathcal{B}_\infty} \|\mathbf{y}_{t+1} - \mathbf{x}\|$.

We need an analog of Lemma 5.4.

Lemma 5.5.

$$\mathbb{E}[\|\mathbf{y}_{t+1} - \mathbf{x}'_t\|_2^2] \geq \gamma^2\eta^2\|A\|_F^2.$$

Proof. Let $\mathbf{z} \sim N(0,1)$

$$\begin{aligned}
\mathbb{E}[\|\mathbf{y}_{t+1} - \mathbf{x}'_t\|_2^2] &= \mathbb{E}[\|\gamma A \mathbf{x}'_t\|_2^2] \\
&= \gamma^2 \mathbb{E}[\mathbf{x}_t'^T A^2 \mathbf{x}'_t] \\
&= \gamma^2 \mathbb{E}[(\mathbf{x}_t + \eta \mathbf{z})^T A^2 (\mathbf{x}_t + \eta\mathbf{z})] \\
&= \gamma^2\left(\mathbf{x}_t^T A^2 \mathbf{x}_t + 2\eta \mathbb{E}[\mathbf{z}^T A^2 \mathbf{x}_t] + \eta^2 \mathbb{E}[\mathbf{z}^T A^2 \mathbf{z}]\right)
\end{aligned}$$

We have $\mathbb{E}[\mathbf{z}^T A^2 \mathbf{z}] = \|A\|_F^2$ as in the proof of Lemma 5.4. Furthermore, $\mathbb{E}[\mathbf{z}^T A^2 \mathbf{x}_t] = \mathbb{E}[\langle \mathbf{z}, A^2\mathbf{x}_t\rangle] = 0$ where the second equality follows by the linearity of expectation using the fact that $\mathbb{E}[\mathbf{z}_i] = 0$ for each i.

We have $\mathbf{x}_t^T A^2 \mathbf{x}_t = \sum_{i=1}^n \lambda_i^2 \langle v_i, \mathbf{x}_t\rangle^2 \geq 0$ which completes the proof.

6 Experiments

We design our experiments to understand how well our agorithm behaves on real-world datasets and how it compares to the state-of-the-art approaches. As pointed out in Section **??**, we are not aware of a scalable approach for solving the *multidimensional* balanced partitioning. Hence, we present a comparison of BR with related techniques for *one-dimensional* variant of the problem. For the multi-dimensional variant, discussed in Section **??**, we present...

For our experiments, we use three publicly available social networks and several large subgraphs of the Facebook friendship graph. We utilize the public graphs for which the results of the state-of-the-art minimum-cut paritioning are known. The private datasets serve to demonstrate scalability of our approach and its performance on real-world data. Our dataset is as follows.

- `LiveJournal` is an undirected version of the public social graph (snapshot from 2006) containing 4.8 million vertices and 42.9 million edges [UB13].
- `Twitter` is a public graph of tweets, with about 41 million vertices (twitter accounts) and 2.4 billion edges (denoting followership) [**?**].
- `Friendster` is another public social graph whose minimum-cut partitioning is available [**?**].
- `FB-X` are subgraphs of the Facebook friendship graph, where X indicates the (approximate) number of edges; the data was anonymized before processing.

6.1 One-dimensional partitioning

We evaluate our algorithm, denoted by BR and described in Section **??**, with existing scalable approaches for graph partitioning. Recall that our primary goal is to design and implement a scalable algorithm that can run for very large graphs in ditributed setting. The most relevant works are the label propagation-based approaches by Ugander and Backstrom [UB13] and by Martella at al. [], balanced partitioning via linear embedding by Aydin et al. [], a streaming technique, called Fennel, suggested by Tsourakakis et al. [], and a distributed algorithm called SocialHash by Kabiljo at al. []. We also present results computed by the classical library for graph partitioning, METIS [**?**].

Table 1 compares the percentage of cut edges produced by various

Next we will compare the technique against competing tools. (for $d = 1$, $\varepsilon = 0.03$, $k = 2$). We need the following data:

Graph	BR	SHP	LinEm	Spinner	Fennel	METIS
`Twitter`	7.3% $\varepsilon = 0.02$	8.33% $\varepsilon = 0.01$	7.43% $\varepsilon = 0.03$	15% $\varepsilon = 0.05$	$\mathbf{6.8\%}$ $\varepsilon = 0.1$	11.98% $\varepsilon = 0.03$
`Friendster`	3.73% $\varepsilon = 0.03$	$\mathbf{3.54\%}$ $\varepsilon = 0.01$	11.9% $\varepsilon = 0.03$			

Table 1: bla.

Graph	BR	SHP	machine-hours
`FB-2.5B`	$\mathbf{5.11\%}$	8.75%	1.1
`FB-5.5B`	$\mathbf{4.99\%}$	11.75%	9
`FB-80B`	$\mathbf{5.21\%}$	12.04%	13
`FB-400B`	6.88%	$\mathbf{5.82\%}$	65
`FB-800B`	$\mathbf{5.52\%}$	5.58%	150

Table 2: bla.

– describe distributed – give times

6.2 Multi-dimensional partitioning

Here we describe how the alg works for $d > 1$. For simplicity we pick $d = 2$ and balance on vertices and degrees. Need a plot for:

- LiveJournal graph. Quality of "one-dim-GradientDescent vs iterations", "alternating projection vs iterations", "real projection vs iterations". Another three plots for "Vertex-imbalance vs iterations". Another three plots for "Degree-imbalance vs iterations".
- com-orkut. (If time permits). Do the same for this graph

6.3 Experiments with projections

Dmitry

First, we need to motivate the projection step. We will do it for $d = 1$.

- Consider LiveJournal graph. Compute 6 plots: (i) quality vs iterations, (ii) number of moved vertices vs iterations, (iii) imbalance (max_vertices/avg_vertices) vs iterations. First do it for uniform projection (3 plots), then for binary-search-based one (another 3 plots).
- This one will motivcate the usage of approximate projection. Consider LiveJournal and build a plot "quality vs iterations" for $\varepsilon = 0$ (exact projection), $\varepsilon = 0.01$ (1% imbalance), $\varepsilon = 0.05$, and $\varepsilon = 0.1$.

6.4 Scalability+Distributed computation

We'll do it if we have time and space

Sergey

7 Conclusions

References

[ABM16] Kevin Aydin, MohammadHossein Bateni, and Vahab S. Mirrokni. Distributed balanced partitioning via linear embedding. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, pages 387–396, 2016.

[AFK⁺14] Amihood Amir, Jessica Ficler, Robert Krauthgamer, Liam Roditty, and Oren Sar Shalom. Multiply balanced k -partitioning. In *LATIN 2014: Theoretical Informatics - 11th Latin American Symposium, Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings*, pages 586–597, 2014.

[AG16] Animashree Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 81–102, 2016.

[Ber99] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[BMS⁺16] Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In *Algorithm Engineering - Selected Results and Surveys*, pages 117–158. 2016.

[DGRW12] Daniel Delling, Andrew V. Goldberg, Ilya P. Razenshteyn, and Renato Fonseca F. Werneck. Exact combinatorial branch-and-bound for graph bisection. In *Proceedings of the 14th Meeting on Algorithm Engineering & Experiments, ALENEX 2012, The Westin Miyako, Kyoto, Japan, January 16, 2012*, pages 30–44, 2012.

[DKK⁺16] Laxman Dhulipala, Igor Kabiljo, Brian Karrer, Giuseppe Ottaviano, Sergey Pupyrev, and Alon Shalita. Compressing graphs and indexes with recursive graph bisection. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1535–1544, New York, NY, USA, 2016. ACM.

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 797–842, 2015.

[GLM16] Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2973–2981, 2016.

[JK17] P. Jain and P. Kar. Non-convex Optimization for Machine Learning. *ArXiv e-prints*, December 2017.

[KK95] George Karypis and Vipin Kumar. Metis – unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report, 1995.

[KKP⁺17] Igor Kabiljo, Brian Karrer, Mayank Pundir, Sergey Pupyrev, Alon Shalita, Yaroslav Akhremtsev, and Alessandro Presta. Social hash partitioner: A scalable distributed hypergraph partitioner. *PVLDB*, 10(11):1418–1429, 2017.

[KNS09] Robert Krauthgamer, Joseph Naor, and Roy Schwartz. Partitioning graphs into balanced components. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 942–949, 2009.

[LRS⁺10] Jason D. Lee, Ben Recht, Ruslan Salakhutdinov, Nathan Srebro, and Joel A. Tropp. Practical large-scale optimization for max-norm regularization. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1297–1305, 2010.

[MLLS17] Claudio Martella, Dionysios Logothetis, Andreas Loukas, and Georgos Siganos. Spinner: Scalable graph partitioning in the cloud. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 1083–1094, 2017.

[MM14] Konstantin Makarychev and Yury Makarychev. Nonuniform graph partitioning with unrelated weights. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 812–822, 2014.

[SBG+17] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. Osqp: An operator splitting solver for quadratic programs. *arXiv preprint arXiv:1711.08013*, 2017.

[SQW15] Ju Sun, Qing Qu, and John Wright. When are nonconvex problems not scary? *CoRR*, abs/1510.06096, 2015.

[TGRV14] Charalampos E. Tsourakakis, Christos Gkantsidis, Bozidar Radunovic, and Milan Vojnovic. FENNEL: streaming graph partitioning for massive scale graphs. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 333–342, 2014.

[UB13] Johan Ugander and Lars Backstrom. Balanced label propagation for partitioning massive graphs. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 507–516, 2013.

A Submission of papers to NIPS 2018

NIPS requires electronic submissions. The electronic submission site is

`https://cmt.research.microsoft.com/NIPS2018/`

Please read the instructions below carefully and follow them faithfully.

A.1 Style

Papers to be submitted to NIPS 2018 must be prepared according to the instructions presented here. Papers may only be up to eight pages long, including figures. Additional pages *containing only acknowledgments and/or cited references* are allowed. Papers that exceed eight pages of content (ignoring references) will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2018 are the same as since 2007, which allow for $\sim 15\%$ more words in the paper compared to earlier years.

Authors are required to use the NIPS LaTeX style files obtainable at the NIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

A.2 Retrieval of style files

The style files for NIPS and other conference information are available on the World Wide Web at

`http://www.nips.cc/`

The file `nips_2018.pdf` contains these instructions and illustrates the various formatting requirements your NIPS paper must satisfy.

The only supported style file for NIPS 2018 is `nips_2018.sty`, rewritten for LaTeX 2_{ε}. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The LaTeX style file contains three optional arguments: `final`, which creates a camera-ready copy, `preprint`, which creates a preprint for submission to, e.g., arXiv, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

New preprint option for 2018 If you wish to post a preprint of your work online, e.g., on arXiv, using the NIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text "Preprint. Work in progress." in the footer. This version may be distributed as you see fit. Please **do not** use the `final` option, which should **only** be used for papers accepted to NIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `nips_2018.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections B, C, and D below.

B General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow ¼ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section D regarding figures, tables, acknowledgments, and references.

C Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

C.1 Headings: second level

Second-level headings should be in 10-point type.

C.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

D Citations, figures, tables, references

These instructions apply to everyone.

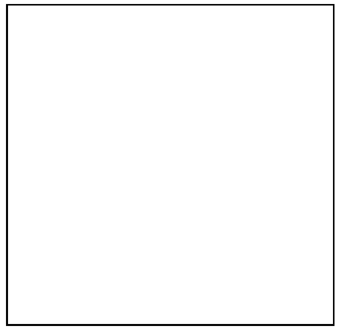

Figure 1: Sample figure caption.

D.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

```
http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf
```

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated...

If you wish to load the `natbib` package with options, you may add the following before loading the `nips_2018` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{nips_2018}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

D.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[2] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[3]

D.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the

[2] Sample of the first footnote.

[3] As in this example.

Table 3: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

D.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 3.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 3.

E Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

F Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

```
\usepackage{amsfonts}
```

followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

```
\newcommand{\RR}{I\!\!R} %real numbers
\newcommand{\Nat}{I\!\!N} %natural numbers
\newcommand{\CC}{I\!\!\!\!C} %complex numbers
```

Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

F.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/`

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can use more than eight pages as long as the additional pages contain *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.