Combinatorial Hodge Theory, Conservation Laws, and You

Chase Dowling, Baosen Zhang

University of Washington, Electrical Engineering

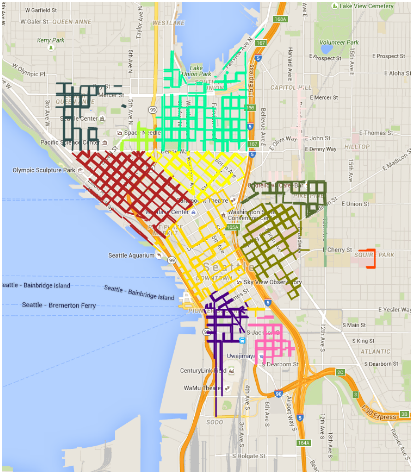
March 19, 2019

Outline

Circling the block for parking

Google Maps can get us there, but where do we park once we get there? How does the city effectively allocate parking supply?

Lots of moving pieces: human factors, non-linearities, low-probability events, new technologies — how can we reliably simplify?



Problem Statement

One of our first questions: where are cars more likely and less likely to find parking, and how does the net flow of traffic searching for parking contribute to congestion?

In other words, where the *sources* and *sinks* for parking-related congestion?

Motivation

In general, answering questions with network or graph-based data incurs burdensome computational complexity.

In current work, we have large amounts of Seattle Department of Transportation data regarding parking and congestion.

Presuming traffic flow is non-linear, how can we simplify, say, optimizing for congestion?

Motivation

Solutions w.r.t data:

- Dimension reduction
  - FFT (graph spectra), graphlets, graphons
- Make simplifying assumptions
  - Linearize edge or vertex relationships, e.g. DC power flow approximation
- Appeal to physical laws
  - Cycle or vertex constraints, e.g. we know flow must be conserved at a vertex
- Build a quantum computer
  - Some assembly required

Motivation

Solutions w.r.t data:

- Dimension reduction
  - FFT (graph spectra), graphlets, graphons
- Make simplifying assumptions
  - Linearize edge or vertex relationships, e.g. DC power flow approximation
- **Appeal to physical laws**
  - **Cycle or vertex constraints, e.g. we know flow must be conserved at a vertex**
- Build a quantum computer
  - Some assembly required

- **Appeal to physical laws**
  - **Cycle or vertex constraints, e.g. we know flow must be conserved at a vertex**

Turns out this is a very general solution for many kinds of physical, infrastructural networks subject to physical laws.

Namely, combinatorial Hodge Theory generalizes familiar graph-theoretic results to simplifying notions from vector calculus like conservation of energy (i.e. curl is 0 everywhere).

Contextual Background

Combinatorial Hodge theory is a mouthful.

First notable application to address *Condorcet's Paradox* as part of a Netflix Prize entry[1].

Condorcet's Paradox says that, given three items A, B, and C where a group of people are told to state their preference over, it is possible to end up in a situation where:

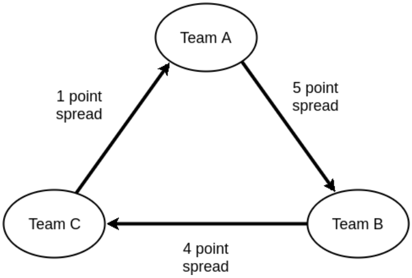
$$A \to B \to C \to A$$

after aggregating preferences.

[1] Jiang, Xiaoye, et al. "Statistical ranking and combinatorial Hodge theory." Mathematical Programming 127.1 (2011): 203-244.

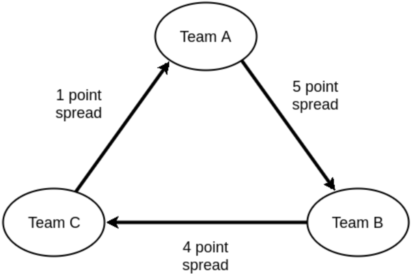
Contextual Background

Sums of flows in cycles should be "conserved" in the sense that $A \to B \to C \to A$ doesn't occur as it does here.



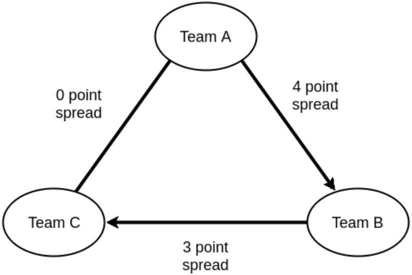
Contextual Background

[Jiang 2011] showed how to uniquely (up to orientation) rectify Condorcet's Paradox–i.e. what is the "closest" ordering or ranking of A, B, and C to the original data that makes sense.



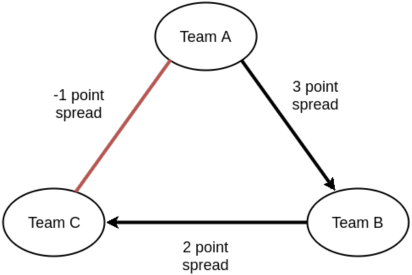
Contextual Background

[Jiang 2011] showed how to uniquely (up to orientation) rectify Condorcet's Paradox–i.e. what is the "closest" ordering or ranking of A, B, and C to the original data that makes sense.



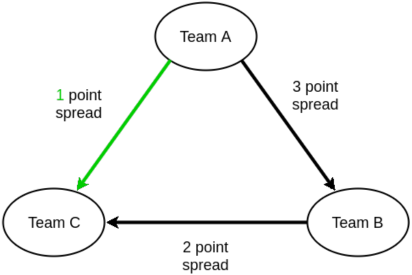
Contextual Background

[Jiang 2011] showed how to uniquely (up to orientation) rectify Condorcet's Paradox–i.e. what is the "closest" ordering or ranking of A, B, and C to the original data that makes sense.



Contextual Background

[Jiang 2011] showed how to uniquely (up to orientation) rectify Condorcet's Paradox–i.e. what is the "closest" ordering or ranking of A, B, and C to the original data that makes sense.



Relevant Work

More fundamental work supporting [Jiang 2011]—generalizing vector calculus to graphs with Hodge Theory

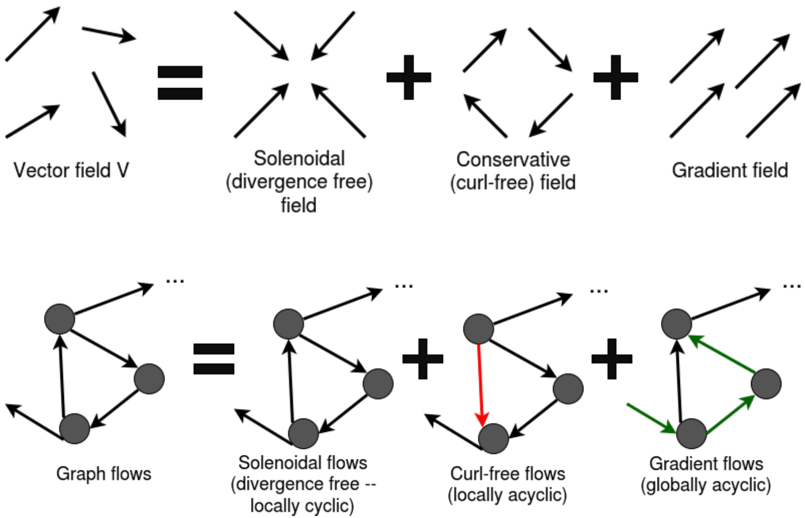
- Friedman, Joel. "Computing Betti numbers via combinatorial Laplacians." Algorithmica 21.4 (1998): 331-346.
- Hirani, Anil N., Kaushik Kalyanaraman, and Seth Watts. "Least squares ranking on graphs." arXiv preprint arXiv:1011.1716 (2010).
- Lim, Lek-Heng. "Hodge Laplacians on Graphs." (2015)

Hodge Theory (In a nutshell)

Combinatorial hodge theory let's me extend the Fundamental Theorem of Vector Calculus (Helmholtz Decomposition) to combinatorial structures like graphs.

This means I can uniquely tease out from flow data the pieces that satisfy conservation laws (cycle or vertex-wise), and the pieces that do not.

Hodge Theory (In a nutshell)



Hodge Theory (In a nutshell)

Why bother?

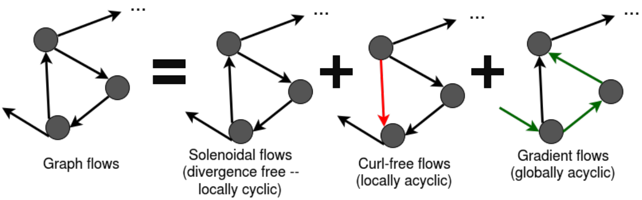
Non-linear flow in graphs can be difficult to optimize according to cost or capacity.

For example, in power grids, technical solutions to optimizing non-linear AC-power flow are hard to extrapolate to arbitrary grid (graph) topologies

Hodge Theory (In a nutshell)

New tools like the Hodge *decomposition* (basis for HodgeRank, which we demonstrate in the parking case) can be used to simplify these non-linear flow problems.

HodgeRank utilizes the gradient flows of the decomposition to rank vertices by how much flow is absorbed or generated—like more general s-t flow graph



Application

We'll use HodgeRank to determine the relative sources and sinks of the flow of parking-related congestion on surface streets in downtown Seattle.

Parking-Related Congestion

We developed a parking-traffic simulator (as a network of specialized queues) using Seattle Department of Transportation (SDOT) data.

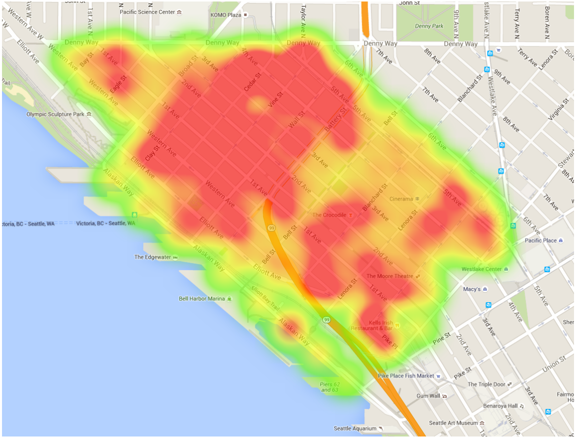
Ratliff, Lillian J., et al. "To Observe or Not to Observe: Queuing Game Framework for Urban Parking." arXiv preprint arXiv:1603.08995 (2016).

Parking-Related Congestion

We'll simulate traffic caused by people searching for the coveted curbside parking space.

- Cars looking for parking behind flow between vertices on edges
- Cars arriving to a vertex to find parking contribute to the source/sink potential
- Cars "served" by a parking space become sinks
- We don't need the individual vertex potential functions per se, only need edge flows

SDOT Data

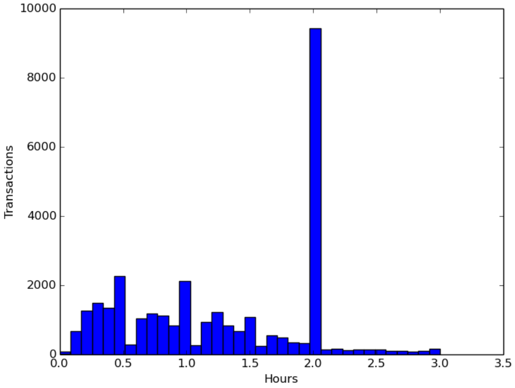


Downtown Seattle neighborhood of Belltown, average streetside-parking utilization during the first quarter of 2015

Just about everyone parks for 2 hours.

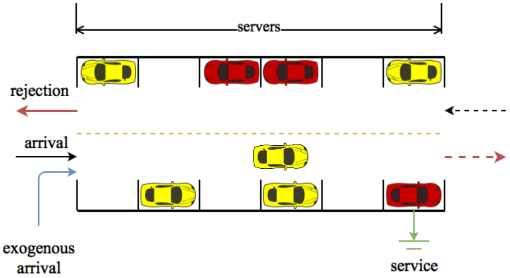
SDOT Data

Number of transactions for paid parking duration during first quarter of 2015

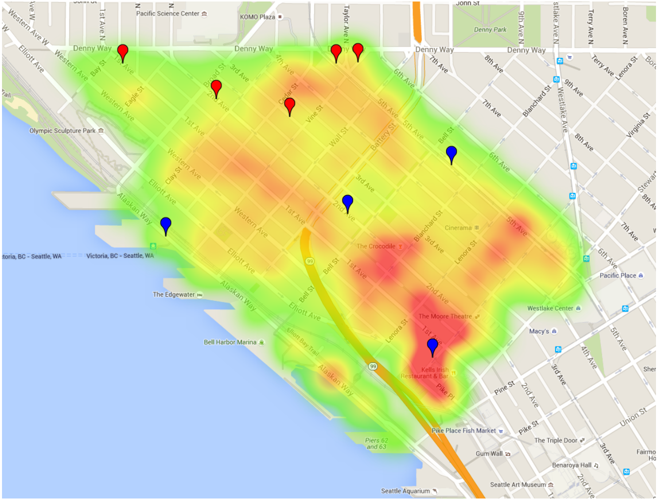


Simulator: Blockface as a Queue

- Interblockface and exogenous arrivals
- Exit after service (parking)
- Reject from queue if servers full
- Number of servers based on parking data
- Reject to intersection adjacent blockface
- Service times based on parking data
- Servers (both sides of street) lumped together
- Queues networked together according to road topology



Hodgerank of Simulated Traffic Flow



Blue pins: sources of traffic, while red pins: sinks.

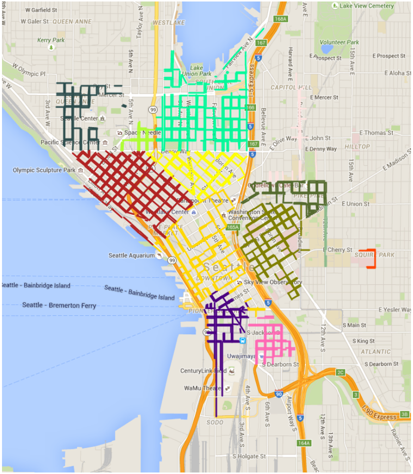
Southeastern blue pin is the northern extent of Pike Market and a major exit for US-99

Control Strategies for Parking Supply

Incorporate time-of-day dependent through-traffic rates

Incorporate exogenous factors such as weather and major events

Reallocate parking according to topologies that relieve congestion on high volume roads



Questions?

Hodge Theory

The Hodge decomposition is a generalization of the fundamental theorem of vector calculus or the Helmholtz decomposition:

Helmholtz Decomposition

Let F be a vector field on bounded $O \subseteq R^3$, twice differentiable, F can be decomposed into a curl-free (conservative) and divergence-free (solenoidal) component:

$$F = -\nabla s + \nabla \times v \qquad (1)$$

Where s is a scalar potential and v a vector potential. A curl-free ($\nabla F = 0$) and divergence-free ($\nabla \times F = 0$) vector field can still be non-zero → solution to Laplace equation $F = -\nabla s$ and $\nabla^2 s = 0$.

Combinatorial Hodge Decomposition

I want to decompose a graph of “vector” flows into curl-free and divergence-free components

Need *div*, *grad*, *curl and all that*

Combinatorial Hodge Decomposition

Relevant coboundary (unweighted L^2) mappings in graph clique complex

$$\begin{aligned} grad &: L^2(V) \to L^2(E) \\ grad\ s(i,j) &= s_j - s_i \end{aligned} \tag{2}$$

$$\begin{aligned} curl &: L^2(E) \to L^2(T) \\ curl\ X(i,j,k) &= X_{ij} + X_{jk} + X_{ki} \end{aligned} \tag{3}$$

$$\begin{aligned} div &: L^2(E) \to L^2(V) \\ div\ X(i) &= \textstyle\sum_j X_{ij} \end{aligned} \tag{4}$$

the gradient operator is the adjoint of the negative divergence operator

Combinatorial Hodge Theory

(Combinatorial Hodge) Helmholtz Decomposition

Let $G = (V, E)$ be a graph and $\mathcal{L}_1$ be its graph Helmholtzian. The space of edge flows, $L^2(E)$ admits an orthogonal decomposition:

$$L^2(E) = im(grad) \oplus ker(\mathcal{L}_1) \oplus im(curl^*) \tag{5}$$

1. $im(grad)$: gradient flows (globally acyclic)
2. $ker(\mathcal{L}_1)$: harmonic flows (locally acyclic)
3. $im(curl^*)$: curl flows (locally cyclic)

1 and 2 form the curl-free component, 2 and 3 form the divergence-free component.

HodgeRank

$$\min_{X \in im(grad)} ||X - Y||_2^2 = \min_{X \in im(grad)} \sum_{(i,j) \in E} (X_{ij} - Y_{ij})^2 \qquad (6)$$

The minimum is calculated by orienting and vectorizing the edge flows Y. The curl-free Hodge ranking, denote it r, on vertex potentials $s \in S$ can be calculated as:

$$r = -(div(E) \otimes grad(S))^{+} grad(S) \qquad (7)$$

$$r = -\mathcal{L}_0^{+} grad(S) \qquad (8)$$