Everything I ever needed to know about geometry I learned from crepes

Crepe Recipes

- **Convex hull:** Finds a convex space that contains all recipe data points
- Connections to: Voronoi, Delaunay Triangulation, Upper Envelope via projections and duality
- Can reduce convex hull to $O(1/\varepsilon)$ vertices if can tolerate distance errors of ε
- Can compute dynamically: $O(n \log n)$ time to add n points

Neighbors & Clusters

- **Voronoi Diagram:** Enables quickly finding nearest neighbor among known recipes
- Delaunay Triangulation: Enables clustering of recipes. Clustering = maximize minimum distance between clusters
- Each recipe is a point in $\mathbb{R}^n$

Duality

- Convex Hull and Voronoi Diagram/Delaunay Triangulation are connected via graph duality and also projective duality.
- A key duality is transformation between points and hyperplanes
- Duality can also solve finding: a linear classifier, a Ham Sandwich, discrepancy, etc.

Crepes on the Cheap

- **Linear Programming:** Finds a “valid” - in the feasible convex space - crepe recipe with minimum cost

- For n constraints and O(1) variables, can solve LP in O(n) expected time with Seidel’s alg. using backwards induction idea.

- Can solve general **LP** within ε factor using MWU. Works even when constraints are concave instead of linear!

Projections onto low dimensional subspaces

Compressing Crepes

- Goal: Project recipes from high (n) dimensional space to low
- **Johnson-Lindenstrauss**
  - Preserves pair-wise distances (and angles) of polynomial points up to ε multiplicative error; $O(\log n/\varepsilon^2)$ dimensions
  - Can compute online
- **Singular Value Decomposition:**
  - Minimizes average distance with original points
  - Must compute offline

Compressing Crepes

- **Johnson-Lindenstrauss:** 1) Learning (ϵ-nets); 2) Reducing state (streaming)
- **Singular Value Decomposition**
  - Data: Compression; Reducing noise
  - Functions: Best linear fit (smallest eigenvector)
  - Graphs: Finding “dense” subgraphs (between recipes and items?)

Learning and Game Theory

Learning crepes

- Recipes are either good are bad
- **Winnow:** Learns a linear classifier
- **Boosting**: Non-linear learning via ensembles of “weak” linear classifiers

Winnow and Boosting use MW

Rock, Scissors, Crepe

- Alice and Bob are on Iron Chef.
- Each chooses a recipe
- A payoff matrix gives the probability that Alice wins for any given choices of Alice and Bob.
- What is Alice’s best randomized strategy if there are spies who learn her probabilities?
- MWU and **fictitious play** give the Nash equilibrium for this game

Topology

Shopping

- Alice and Bob coordinate purchases of butter, milk, flour and eggs
- Communicate via texts, but don't always check phone.
- Can they decide on a partition of the items?

Shopping Coordination

- **Carrier map** describes
  - Communication model
  - Desired mapping from input to output simplicial complex
- **Simplicial mapping** describe outputs of Alice and Bob based on all information they've received
- "Reasonable" maps preserve connectivity between the input and output shapes.
- Can determine what distributed computation is possible based on whether or not this connectivity is maintained.

Viva Le Crepe