Combinatorial Approach to Data Mining

Yury Lifshits
Caltech
http://yury.name

MIT, 29 November 2007

Based on joint work with Navin Goyal, Benjamin Hoffmann, Dirk Nowotka, Hinrich Schütze and Shengyu Zhang
Nearest neighbors
Preprocess a set $S$ such that given any $q$
the closest point in $S$ to $q$ can be found quickly

Near-duplicates
Find all pairs of objects with distance
below some threshold in subquadratic time

Navigability design
Construct a graph such that local routing
is leading to target in logarithmic number of steps

Clustering
Split a set to $k$ parts minimizing in-cluster distances
Nearest neighbors
Preprocess a set $S$ such that given any $q$
the closest point in $S$ to $q$ can be found quickly

Near-duplicates
Find all pairs of objects with distance
below some threshold in subquadratic time

Navigability design
Construct a graph such that local routing
is leading to target in logarithmic number of steps

Clustering
Split a set to $k$ parts minimizing in-cluster distances

Today: distances are not given,
triangle inequality is not satisfied
Outline

1. Combinatorial Framework
2. Results: New Algorithms
3. One Proof: Visibility Graph
4. Open Problems
1

Combinatorial Framework
Comparison Oracle

- Dataset \( p_1, \ldots, p_n \)
- Objects and distance (or similarity) function are NOT given
- Instead, there is a comparison oracle answering queries of the form:

  **Who is closer to \( A \): \( B \) or \( C \)?**
Disorder Inequality

Sort all objects by their similarity to $p$:

$$\text{rank}_p(r)$$

$$\text{rank}_p(s)$$
Disorder Inequality

Sort all objects by their similarity to $p$:

Then by similarity to $r$:

$$\forall p, r, s: \text{rank}_p(r) \leq \text{D} \left( \text{rank}_p(r) + \text{rank}_p(s) \right)$$
Disorder Inequality

Sort all objects by their similarity to $p$:

Then by similarity to $r$:

Dataset has disorder $D$ if

$$\forall p, r, s : \quad rank_r(s) \leq D(rank_p(r) + rank_p(s))$$
Combinatorial Framework

\[ = \]

Comparison oracle
Who is closer to A: B or C?

+ 

Disorder inequality
\[ \text{rank}_r(s) \leq D(\text{rank}_p(r) + \text{rank}_p(s)) \]
Combinatorial Framework: Pro & Contra

Advantages:

- Does not require triangle inequality for distances
- Applicable to any data model and any similarity function
- Require only comparative training information
- Sensitive to “local density” of a dataset
Advantages:

- Does not require triangle inequality for distances
- Applicable to any data model and any similarity function
- Require only comparative training information
- Sensitive to “local density” of a dataset

Limitation: worst-case form of disorder inequality
Combinatorial Ball

\[ B(x, r) = \{ y : \text{rank}_x(y) < r \} \]

In other words, it is a subset of dataset \( S \): the object \( x \) itself and \( r - 1 \) its nearest neighbors.

\[ B(x, 10) \]
A subset $R \subseteq S$ is called a **combinatorial $r$-net** iff the following two properties holds:

**Covering:** $\forall y \in S, \exists x \in R, \text{ s.t. } \text{rank}_x(y) < r.$

**Separation:** $\forall x_i, x_j \in R, \text{ rank}_x(x_j) \geq r \text{ OR rank}_x(x_i) \geq r.$
Combinatorial Net

A subset \( R \subseteq S \) is called a **combinatorial \( r \)-net** iff the following two properties holds:

**Covering:** \( \forall y \in S, \exists x \in R, \text{ s.t. } \operatorname{rank}_x(y) < r \).

**Separation:** \( \forall x_i, x_j \in R, \operatorname{rank}_{x_i}(x_j) \geq r \text{ OR } \operatorname{rank}_{x_j}(x_i) \geq r \).

How to construct a combinatorial net?
What upper bound on its size can we guarantee?
Disorder vs. Others

- If expansion rate is $c$, disorder constant is at most $c^2$
- Doubling dimension and disorder dimension are incomparable
- Disorder inequality implies combinatorial form of “doubling effect”
Results:
Combinatorial Algorithms
Combinatorial nets:
For every $0 \leq i \leq \log n$, construct a $\frac{n}{2^i}$-net
Basic Data Structure

**Combinatorial nets:**
For every $0 \leq i \leq \log n$, construct a $\frac{n}{2^i}$-net

**Pointers, pointers, pointers:**
- **Direct & inverted indices:** links between centers and members of their balls
- **Cousin links:** for every center keep pointers to close centers on the same level
- **Navigation links:** for every center keep pointers to close centers on the next level
Theorem

Combinatorial nets can be constructed in $O(D^7 n \log^2 n)$ time
Nearest Neighbor Search

Assume $S \cup \{q\}$ has disorder constant $D$

**Theorem**

There is a deterministic and exact algorithm for nearest neighbor search:

- **Preprocessing:** $O(D^7 n \log^2 n)$
- **Search:** $O(D^4 \log n)$
Nearest Neighbor Search

Assume $S \cup \{q\}$ has disorder constant $D$

**Theorem**

There is a deterministic and exact algorithm for nearest neighbor search:

- **Preprocessing**: $\mathcal{O}(D^7 n \log^2 n)$
- **Search**: $\mathcal{O}(D^4 \log n)$

**Variations:**

- $\mathcal{O}(n)$ size of data structure, still $\text{poly}(D) \log n$ search
- Randomized algorithm, $\mathcal{O}(D \log n)$ search
Navigability Design

**Local routing in a graph:**
Given target description and the current node $p$, a message is forwarded via one of the out-going edges from $p$. 

Design task:
Given a collection of points $S = \{p_1, \ldots, p_n\}$ construct a low-degree graph and rules for local decisions such that given a start $p \in S$ and a target $q$ the nearest neighbor of $q$ in $S$ can be reached in a small number of steps.
Navigability Design

**Local routing in a graph:**
Given target description and the current node \( p \)
a message is forwarded via one of the out-going edges from \( p \)

**Design task:**
Given a collection of points \( S = \{ p_1, \ldots, p_n \} \)
construct a low-degree graph and rules for local decisions
such that given a start \( p \in S \) and a target \( q \)
the nearest neighbor of \( q \) in \( S \)
can be reached in a small number of steps
Theorem

Any dataset \( S \) has a **visibility graph**:

- \( \text{poly}(D)n \log^2 n \) construction time
- \( \mathcal{O}(D^4 \log n) \) out-degrees
- Naïve greedy routing **deterministically** reaches exact nearest neighbor of \( q \) in at most \( \log n \) steps
Near-Duplicates

Assume, comparison oracle can also tell us whether $\sigma(x, y) > T$ for some similarity threshold $T$

**Theorem**

All pairs with over-$T$ similarity can be found deterministically in time

$$\text{poly}(D)(n \log^2 n + |\text{Output}|)$$
Clustering

Combinatorial objective function for $k$-clustering:

Minimize $\sum_{i \in [k]} \sum_{x, y \in C_i} rank_x(y)$
Clustering

Combinatorial objective function for $k$-clustering:

$$\text{Minimize} \quad \sum_{i \in [k]} \sum_{x, y \in C_i} \text{rank}_x(y)$$

**Theorem**

A $32D^3$-approximate clustering can be constructed in time $\text{poly}(D)n \log^2 n$
One Proof: Visibility Graph
Problem Statement

Input:
Dataset \( S = \{p_1, \ldots, p_n\} \)
Represented by comparison oracle
Having disorder constant \( D \)
Problem Statement

**Input:**
Dataset $S = \{p_1, \ldots, p_n\}$
Represented by comparison oracle
Having disorder constant $D$

**Design Task:**
Connect every object with few others
Set local rules for routing
Problem Statement

**Input:**
Dataset \( S = \{p_1, \ldots, p_n\} \)
Represented by comparison oracle
Having disorder constant \( D \)

**Design Task:**
Connect every object with few others
Set local rules for routing

**Routing Requirement:** Given a target point \( q \) and a starting point \( p \in S \) the nearest neighbor of \( q \) in \( S \) should be reached by a few steps in the graph
Greedy Routing

1. Use oracle to compare distances to $q$ from current point $p$ and from all its neighbors in the graph.

2. If $p$ is not the closest one, move to the one which is the closest.

3. Otherwise, STOP and return $p$. 

Also known as local search, hill climbing etc.
Greedy Routing

1. Use oracle to compare distances to $q$ from current point $p$ and from all its neighbors in the graph.

2. If $p$ is not the closest one, move to the one which is the closest.

3. Otherwise, STOP and return $p$.

Also known as local search, hill climbing etc.
Definition of Visibility

A center $c_i$ in the $\frac{n}{2^i}$-net is visible from some object $p$ iff

$$\text{rank}_p(c_i) \leq 3D^2 \frac{n}{2^i}$$
Definition of Visibility

A center $c_i$ in the $\frac{n}{2^i}$-net is visible from some object $p$ iff

$$\text{rank}_p(c_i) \leq 3D^2 \frac{n}{2^i}$$

**Interpretation:** the farther you are the larger radius you need to be visible
Definition of Visibility

A center $c_i$ in the $\frac{n}{2^i}$-net is **visible** from some object $p$ iff

$$\text{rank}_p(c_i) \leq 3D^2 \frac{n}{2^i}$$

**Interpretation:** the farther you are the larger radius you need to be visible
Analysis

Three claims:

- Out-degrees are $O(D^4 \log n)$
- After $i$ steps we reach a point that is at least as close to $q$ as the best center in $\frac{n}{2^i}$-net
- Visibility graph can be constructed in $poly(D)n \log^2 n$ time
Bound on Degrees

Connecting $p$ with centers of $r$-net:

- By construction, centers have ranks at most $3D^2r$ to $p$
- There are disjoint $\frac{r}{2D}$ balls around these centers
- Members of these disjoint balls have $O(D^3)r$ rank to $p$
- Thus, there are at most $O(D^4)$ such centers
After $i$ steps we reach a point that is at least as close to $q$ as the best point in $\frac{n}{2^i}$-net

**Inductive proof.** From $i$ to $i+1$:

- For the best center in $i$-th level $\text{rank}_q(c_i^*) \leq Dr_i$.

Similarly, $c_{i+1}^*$ satisfies $\text{rank}_q(c_{i+1}^*) \leq \frac{Dr_i}{2}$.
After $i$ steps we reach a point that is at least as close to $q$ as the best point in $\frac{n}{2^i}$-net

**Inductive proof.** From $i$ to $i + 1$:

- For the best center in $i$-th level $\text{rank}_q(c_i^*) \leq Dr_i$.

  Similarly, $c_{i+1}^*$ satisfies $\text{rank}_q(c_{i+1}^*) \leq \frac{Dr_i}{2}$

- From inductive conjecture: after $i$ steps in a greedy walk the current point $p^{(i)}$ also has $\text{rank}_q(p^{(i)}) \leq Dr_i$
Fast Convergence

After $i$ steps we reach a point that is at least as close to $q$ as the best point in $\frac{n}{2^i}$-net

**Inductive proof.** From $i$ to $i+1$:

1. For the best center in $i$-th level $\text{rank}_q(c_i^*) \leq Dr_i$. Similarly, $c_{i+1}^*$ satisfies $\text{rank}_q(c_{i+1}^*) \leq \frac{Dr_i}{2}$

2. From inductive conjecture: after $i$ steps in a greedy walk the current point $p^{(i)}$ also has $\text{rank}_q(p^{(i)}) \leq Dr_i$

3. By disorder inequality $p^{(i)}$ is connected to $c_{i+1}^*$

Therefore $p^{(i+1)}$ is at least as good as $c_{i+1}^*$ is
Directions for Further Research

- Other problems in combinatorial framework:
  - Low-distortion embeddings
  - Closest pairs
  - Community discovery
  - Linear arrangement
  - Distance labelling
  - Dimensionality reduction

- What if disorder inequality has exceptions, but holds in average?
- Insertions, deletions, changing metric
- Metric regularizations
- Experiments & implementation
Call for Feedback

- What do you like the most in these results?
- What is the most important question for further studies?
- Relevant literature?

Another talk: YL, "Open Problems TO GO"
Friday Nov 30, 4pm, 56-154, MIT Theory Reading Group
Call for Feedback

- What do you like the most in these results?
- What is the most important question for further studies?
- Relevant literature?
- Are you interested in further discussions? I am around this evening and the whole Friday.
Call for Feedback

- What do you like the most in these results?
- What is the most important question for further studies?
- Relevant literature?
- Are you interested in further discussions? I am around this evening and the whole Friday.

Another talk: YL, “Open Problems TO GO”
Friday Nov 30, 4pm, 56-154, MIT Theory Reading Group
http://yury.name

http://simsearch.yury.name
Tutorial, bibliography, people, links, open problems

Yury Lifshits and Shengyu Zhang
Similarity Search via Combinatorial Nets

Navin Goyal, Yury Lifshits, Hinrich Schütze
Disorder Inequality: A Combinatorial Approach to Nearest Neighbor Search

Benjamin Hoffmann, Yury Lifshits, Dirk Novotka
Maximal Intersection Queries in Randomized Graph Models
Summary

- **Combinatorial framework:** comparison oracle + disorder inequality
- **Near-linear construction** of combinatorial nets
- Nearest neighbor search in **almost logarithmic** time
- **Deterministic** detection of near-duplicates in **subquadratic** time
- **Visibility graph:** small degrees and deterministic convergence in $\log n$ steps
Summary

- **Combinatorial framework:** comparison oracle + disorder inequality
- **Near-linear construction** of combinatorial nets
- Nearest neighbor search in *almost logarithmic* time
- **Deterministic** detection of near-duplicates in subquadratic time
- **Visibility graph:** small degrees and deterministic convergence in $\log n$ steps

Thanks for your attention!
Questions?