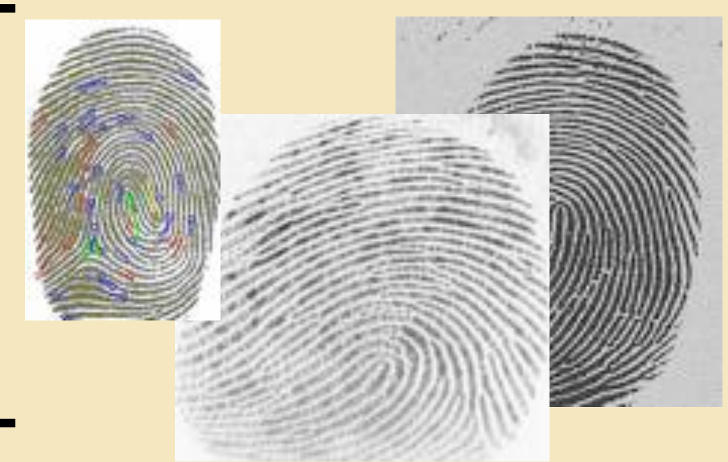
Fast nearest neighbor retrieval for bregman divergences

Lawrence Cayton
UC San Diego

Nearest neighbor search

query: (very large) database:

$q =$  find best match in $X =$ 

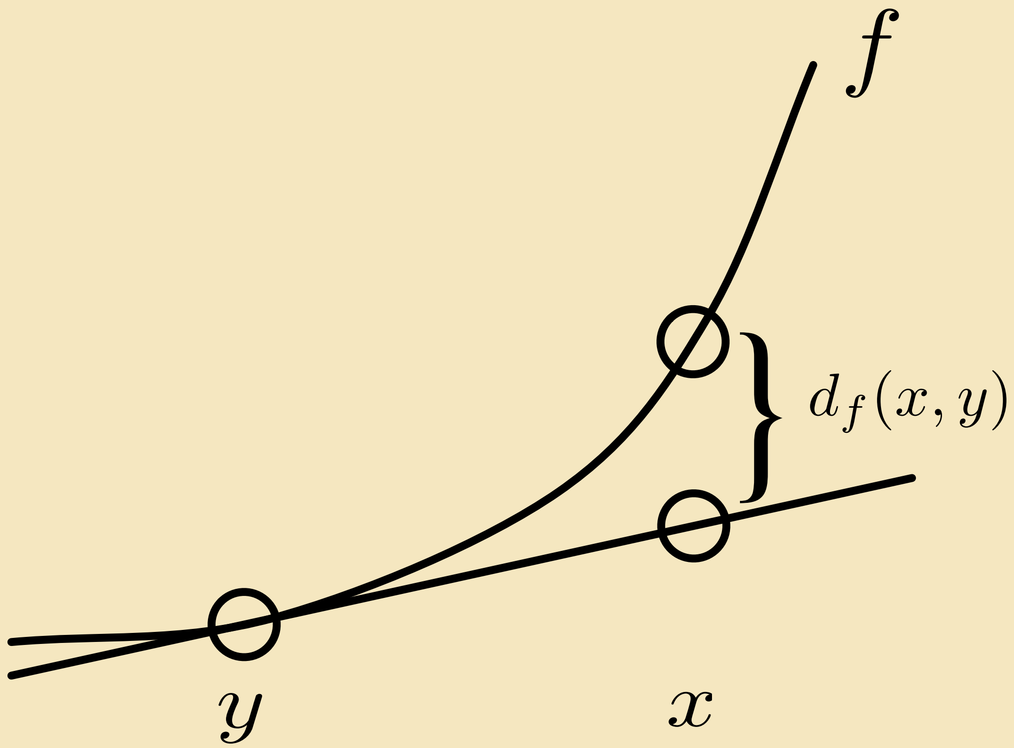
- NN methods ubiquitous, but expensive
- Many NN data structures designed to reduce the complexity, mostly for metrics
- In learning, vision, text, use many non-metric measures; a prominent example is the KL-divergence.

This work: a data structure designed for bregman divergences.

Bregman divergence def

For strictly convex $f : \mathbb{R}^d \to \mathbb{R}$,

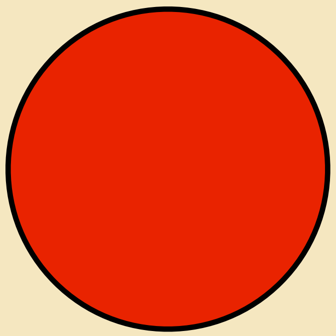
$$d_f(x, y) \equiv f(x) - f(y) - \langle \nabla f(y), x - y \rangle$$



Bregman divergence examples

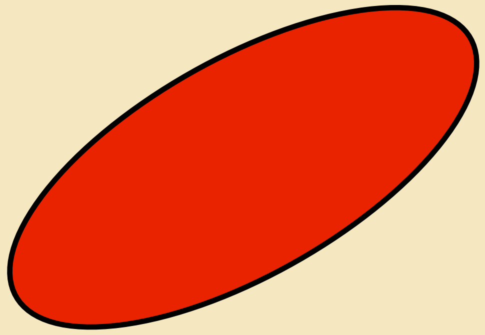
ℓ_2^2

$$d_f(x, y) = \frac{1}{2}\|x - y\|_2^2$$



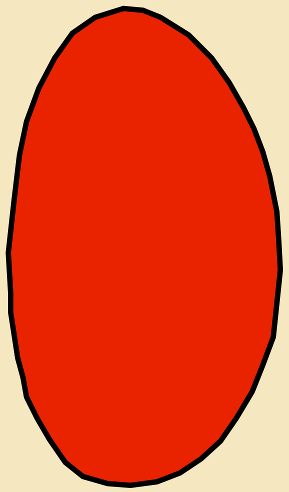
Mahalanobis ($Q \succ 0$)

$$d_f(x, y) = \frac{1}{2}(x - y)^\top Q(x - y)$$



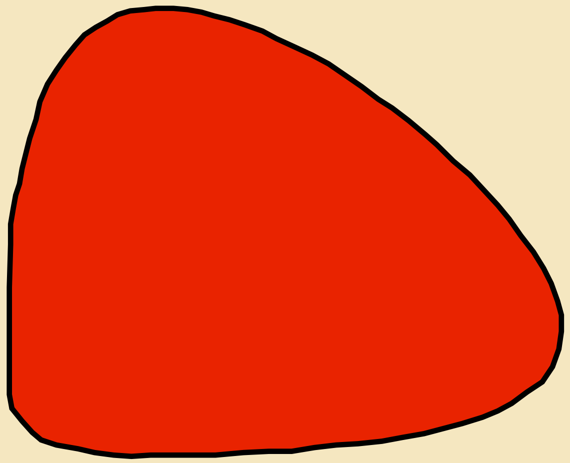
KL-divergence

$$d_f(x, y) = \sum x_i \log \frac{x_i}{y_i}$$



Itakura-Saito

$$d_f(x, y) = \sum \left(\frac{x_i}{y_i} - \log \frac{x_i}{y_i} - 1\right)$$



Bregman divergences VS metrics

Metrics:

non-negativity	symmetry	triangle inequality
$d(x, y) \geq 0$	$d(x, y) = d(y, x)$	$d(x, y) + d(y, z) \geq d(x, z)$

Bregman divergences VS metrics

Metrics:

non-negativity

$d(x,y) \geq 0$

symmetry

$d(x,y) = d(y,x)$

triangle inequality

$d(x,y) + d(y,z) \geq d(x,z)$

Bregman divergences:

non-negativity

$d_f(x,y) \geq 0$

~~symmetry~~

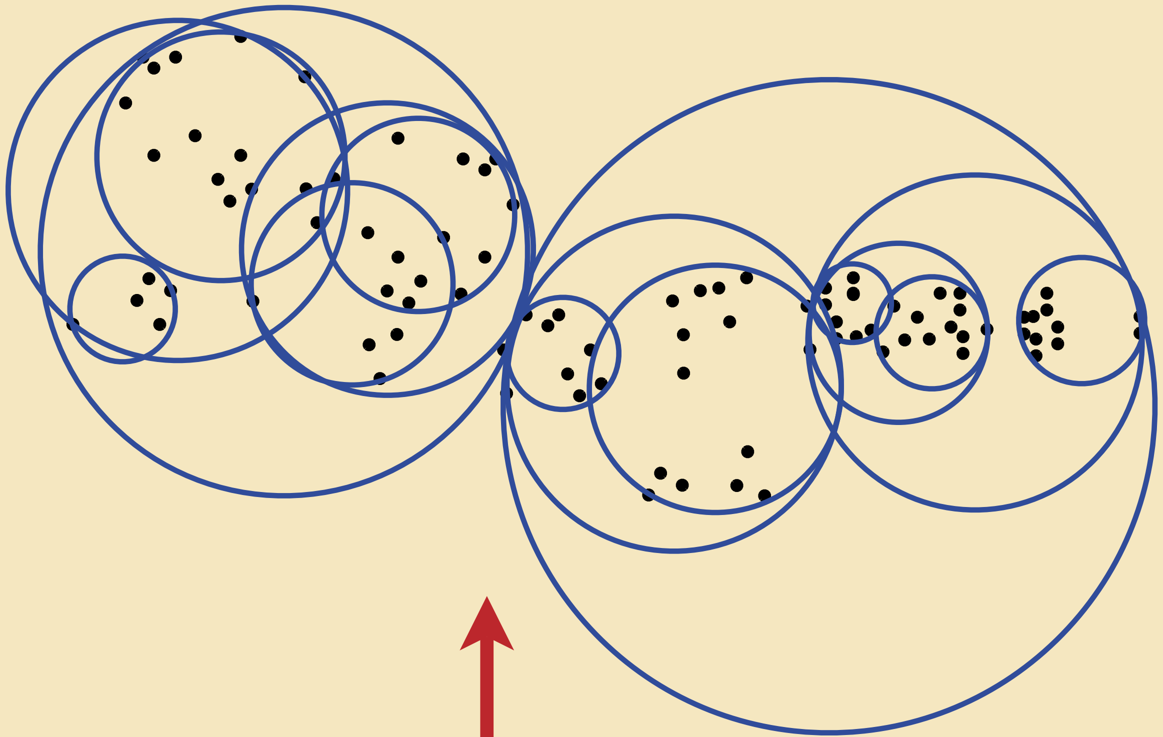
~~$d_f(x,y) = d_f(y,x)$~~

~~triangle inequality~~

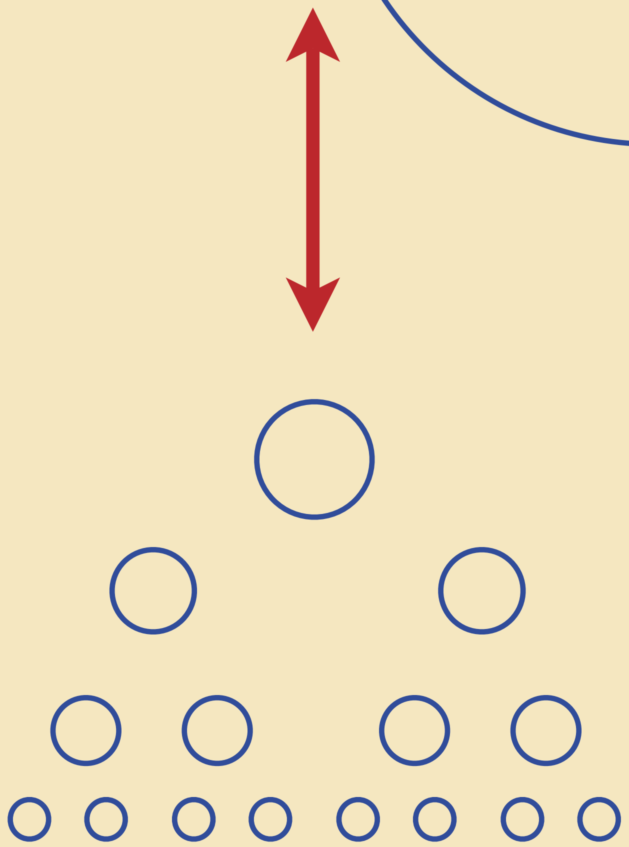
~~$d_f(x,y) + d_f(y,z) \geq d_f(x,z)$~~

Review: tree-based NN retrieval

e.g. kd-trees, metric trees, many many variants



Hierarchical space decomposition



Search via branch and bound exploration

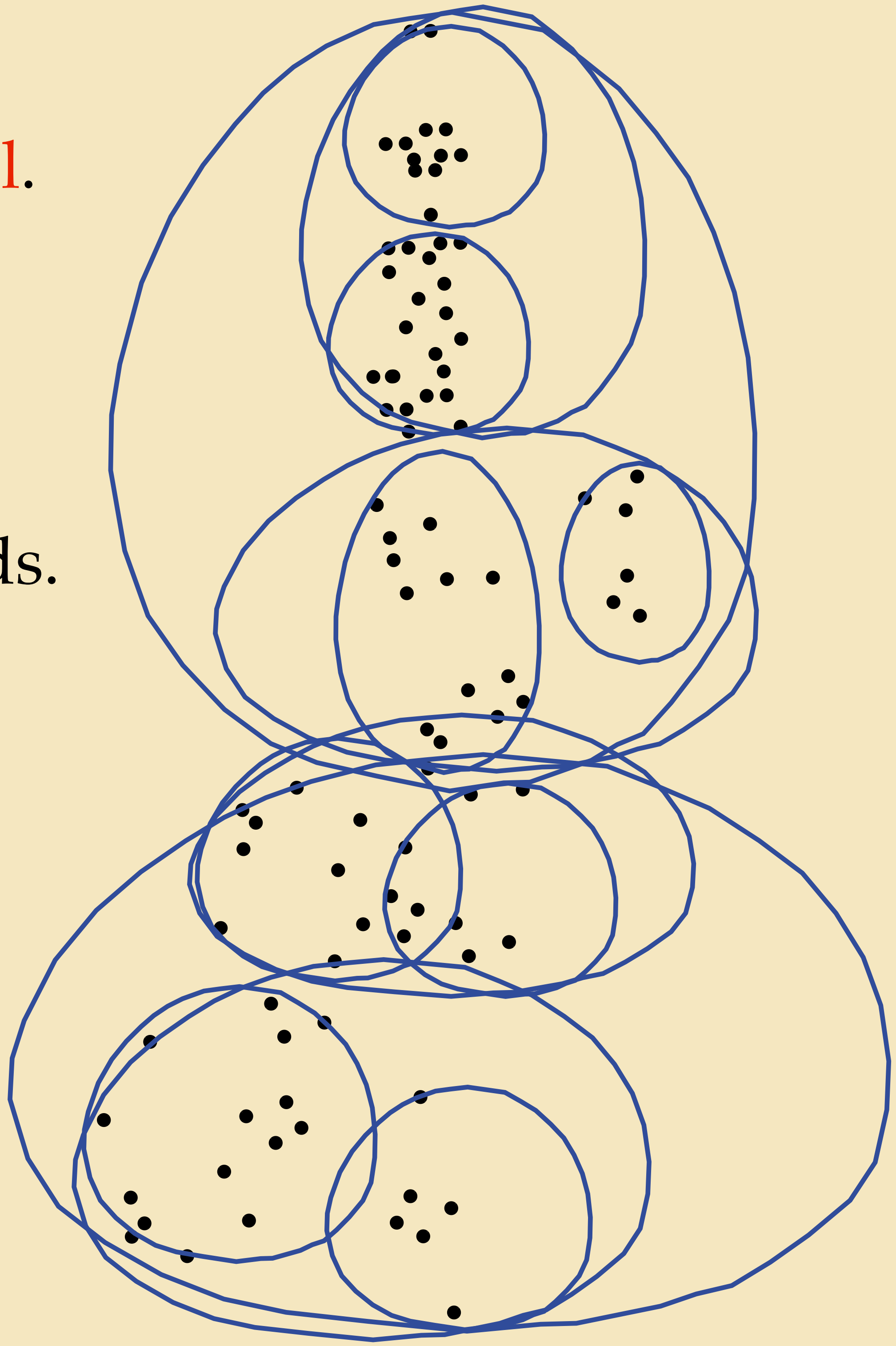
Bregman ball trees

- Fundamental geometric unit: bregman ball.

$$B(\mu, R) \equiv \{x \;\; : \;\; d_f(x, \mu) \leq R\}$$

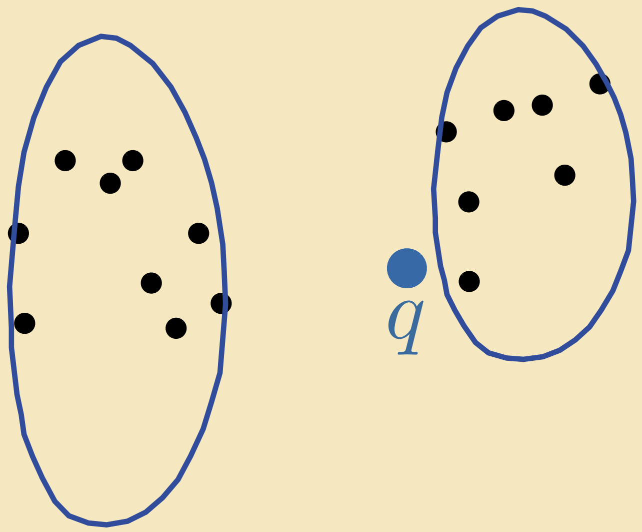
- Need a reasonable build heuristic.
- Can't use the triangle inequality for bounds.
- Need to handle asymmetry of divergence.

(Not covered here -- see paper)



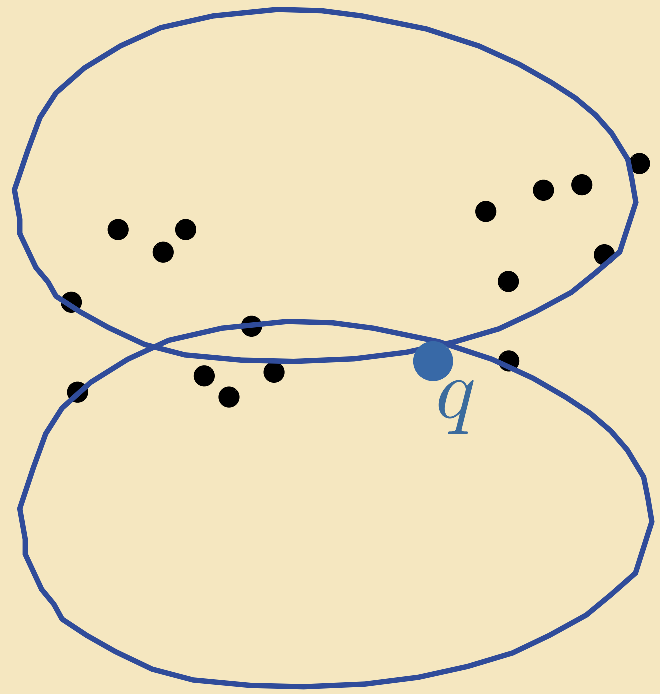
bbtree -- build

Intuition: at each level, want balls that are well separated & compact.



Can prune left node

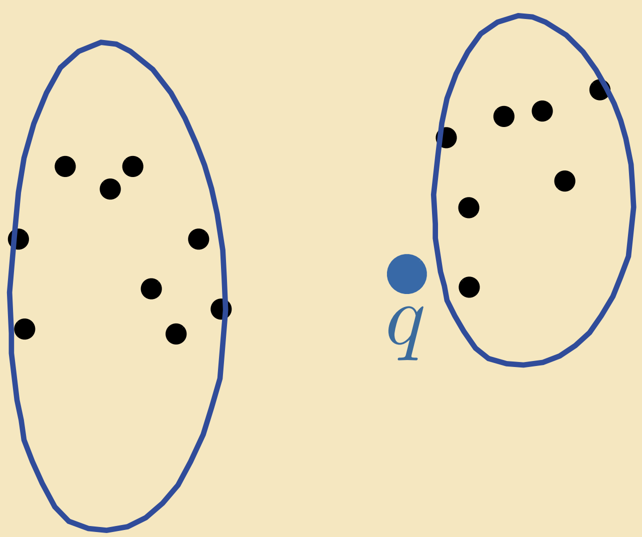
VS



Have to search both

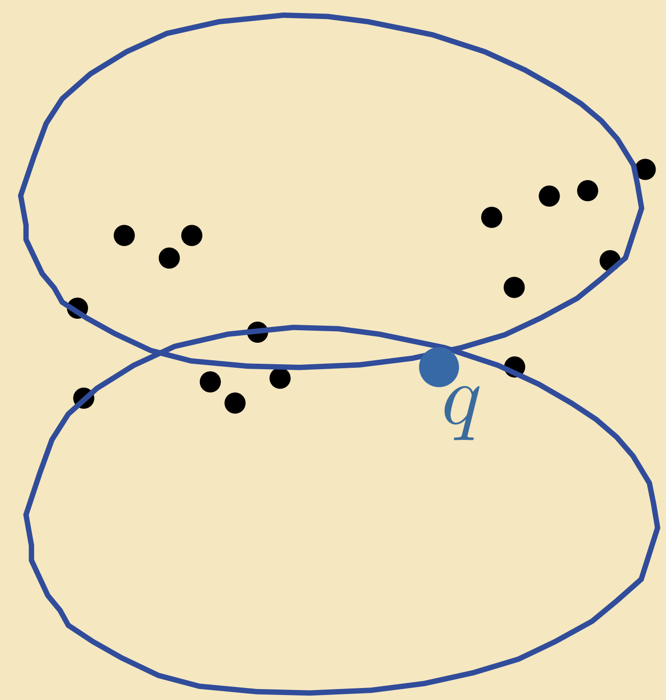
bbtree -- build

Intuition: at each level, want balls that are well separated & compact.



Can prune left node

VS



Have to search both

Build method: Deploy k-means hierarchically (top-down).

(k-means was generalized to bregman divergences in Banerjee *et al.* 2005)

bbtree -- search

Want to find the left NN:

$$\text{argmin}_{x \in X} d_f(x, q)$$

Branch & bound search:

1. Descend tree, choosing child whose mean is closest to q. *Ignore* the sibling.

2. At leaf, compute distances to all points; call the nearest the *candidate* NN x_c.

3. Traverse back up tree; check the ignored nodes. If

$$d_f(x_c, q) > \min_{x \in B(\mu, R)} d_f(x, q)$$

need to explore it.

dist to candidate NN

dist to bregman ball

Computing the bound

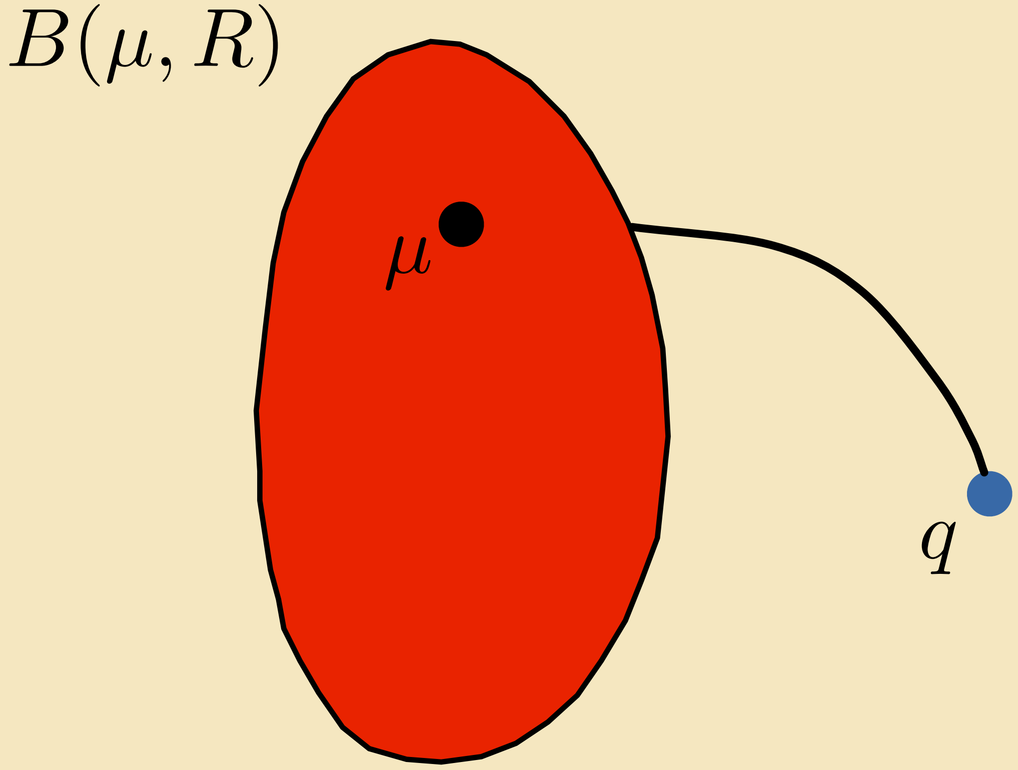
Need to check if

$$d_f(x_c, q) > \min_{x \in B(\mu, R)} d_f(x, q)$$

Computing the bound

Need to check if

$$d_f(x_c, q) > \boxed{\min_{x \in B(\mu,R)} d_f(x, q)}$$



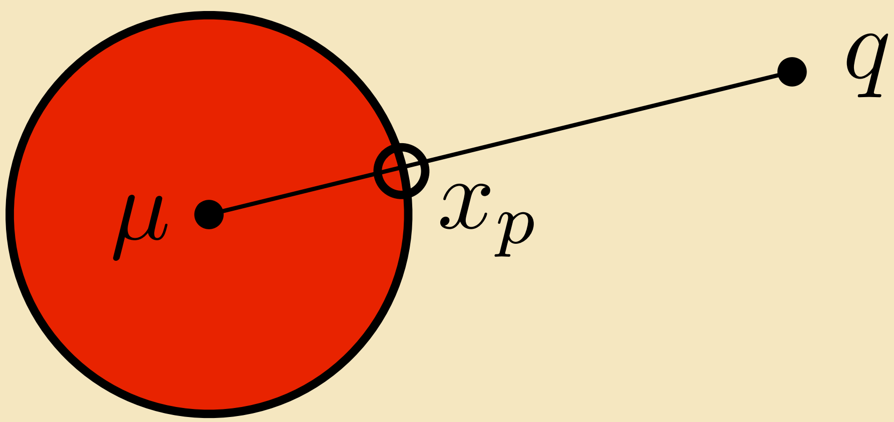
The *bregman projection onto a bregman ball*

Convex, but need to compute it in time comparable to evaluating an analytic expression

The ℓ_2^2 case

$$\begin{aligned} \min_x \quad & \tfrac{1}{2}\|x - q\|^2 \\ \text{subject to:} \quad & \tfrac{1}{2}\|x - \mu\|^2 \leq R \end{aligned}$$

Can compute projection analytically:



$$x_p = \theta\mu + (1 - \theta)q$$

$$\text{where } \theta = \frac{\sqrt{2R}}{\|q - \mu\|}$$

Easy because

x_p is on line between μ and q

The general case

$$\begin{aligned} \min_{x} \quad & d_f(x, q) \\ \text{subject to:} \quad & d_f(x, \mu) \leq R \end{aligned}$$

Something similar holds..

The general case

$$\min_x \quad d_f(x, q)$$
$$\text{subject to:} \quad d_f(x, \mu) \leq R$$

Something similar holds..

Claim 1: $\nabla f(x_p) = \theta \nabla f(\mu) + (1 - \theta) \nabla f(q)$.

→ The ℓ_2^2 relationship is a special case since $\nabla f(x) = x$.

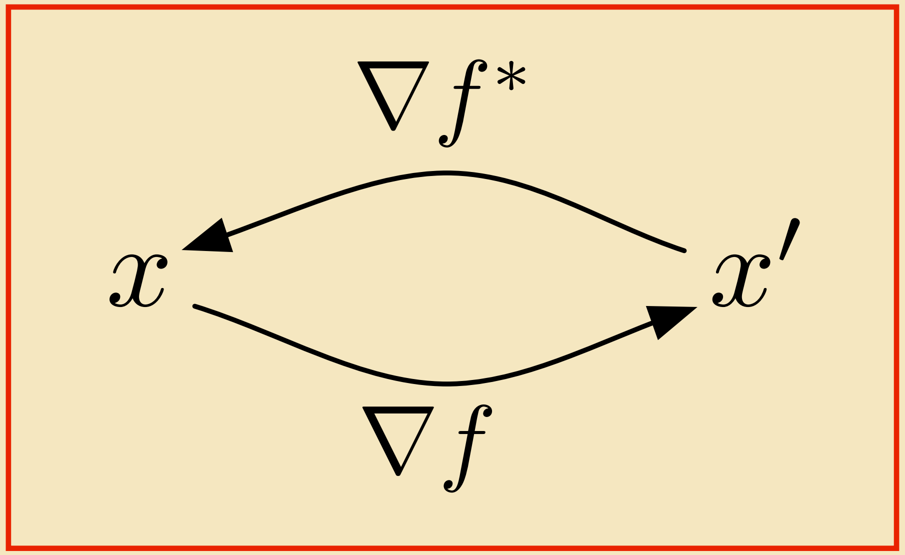
Nearly as useful....

Since f strictly convex,

∇f is one-to-one.

Moreover, its inverse is given by the gradient of

$$f^*(y) \equiv \sup_x \{\langle x, y \rangle - f(x)\}\,.$$

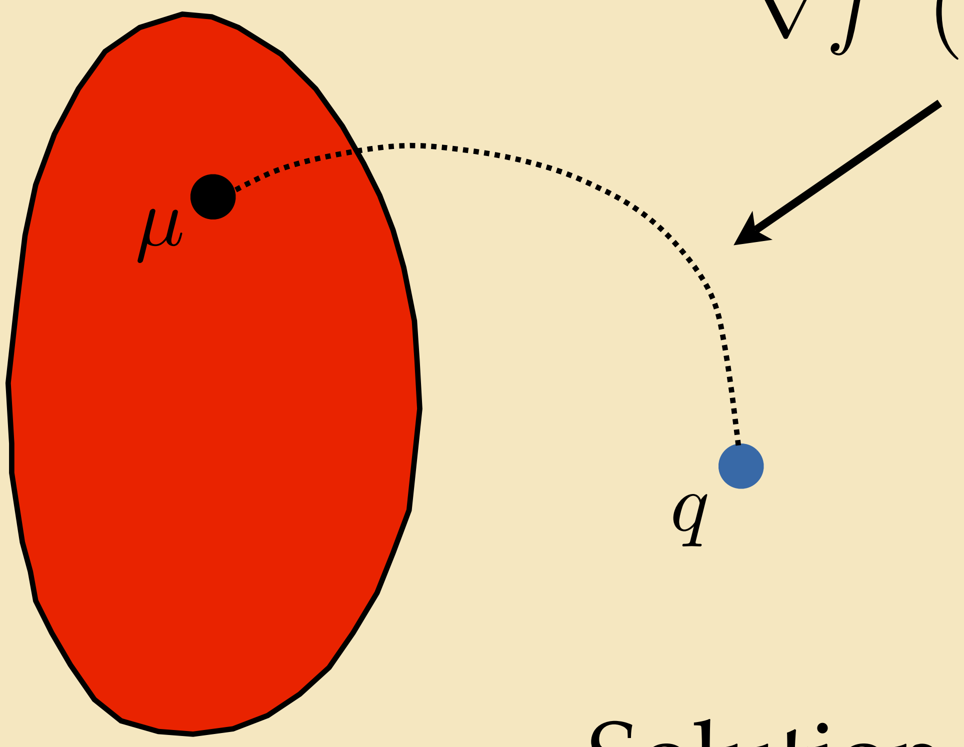


Thus can recover x_p from $\nabla f(x_p)$

Notation:

$$\mu' \equiv \nabla f(\mu)$$
$$q' \equiv \nabla f(q)$$
$$x'_\theta \equiv \theta\mu' + (1-\theta)q'$$

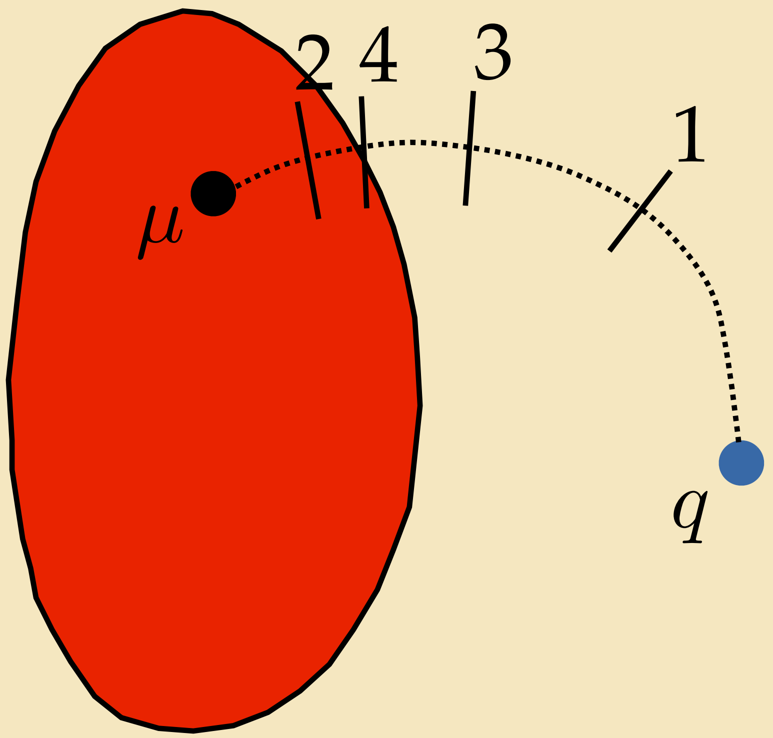
$$\nabla f^*(x'_\theta), \quad \theta \in [0,1]$$



Solution lies on this curve.

Algorithm

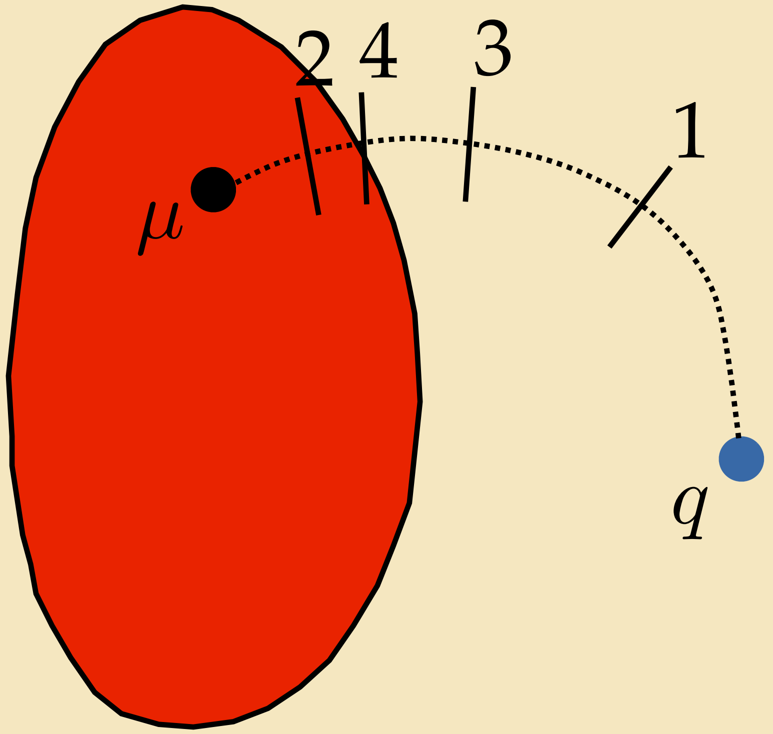
Bisection search on θ for x satisfying $d_f(x, \mu) = R$.



1. $\theta_1 = \frac{1}{2}$
2. $\theta_2 = \frac{1}{4}$
3. $\theta_3 = \frac{3}{8}$
4. $\theta_4 = \frac{5}{16}$

Algorithm

Bisection search on θ for x satisfying $d_f(x, \mu) = R$.



1. $\theta_1 = \frac{1}{2}$
2. $\theta_2 = \frac{1}{4}$
3. $\theta_3 = \frac{3}{8}$
4. $\theta_4 = \frac{5}{16}$

- Can compute a solution to accuracy ϵ in $\log \frac{1}{\epsilon}$ steps.
- Each step requires 1 gradient evaluation and 1 divergence evaluation.

Very fast.

But: Don't actually need an exact solution.

Only need to determine if: $$d_f(x_c, q) > \min_{x \in B(\mu, R)} d_f(x, q)$$

(x_c is the current candidate NN)

But: Don't actually need an exact solution.

Only need to determine if: $$d_f(x_c, q) > \min_{x \in B(\mu,R)} d_f(x, q)$$

(x_c is the current candidate NN)

i.e. upper and lower bounds suffice

Lower bound: weak duality

$$\begin{aligned} \mathcal{L}(\theta) &\equiv d_f(x_\theta, q) + \frac{\theta}{1-\theta}\Big(d_f(x_\theta, \mu) - R\Big) \\ &\leq \min_{x \in B(\mu,R)} d_f(x, q) \end{aligned}$$

Upper bound: primal

$$d_f(x_\theta, q) \geq \min_{x \in B(\mu,R)} d_f(x, q)$$

for feasible x_θ

Evaluate bounds at each step of bisection to stop early.

Experiments: KL-divergence

Why KL divergence?

- Used extensively to compare histograms (*e.g.* text, vision).
- No (correct) NN schemes out there for it.
- Mahalanobis, ℓ_2^2 can be handled by metric methods.

Data sets

- **rcv-D**: 500k documents from the RCV corpus represented as a D-dimensional distribution over topic (generated using LDA).
- **Corel histograms**: 60k color histograms, 64-dimensional.
- **Semantic space**: 371-dimensional representation of 5000 images (from CBIR literature)
- **SIFT signatures**: 1111-dimensional quantized histogram representation of 10k images from PASCAL 2007 dataset

Approx search experiments

Stop search early (after examining only a few leaves)
-- standard practice with metric, kd-trees, etc.

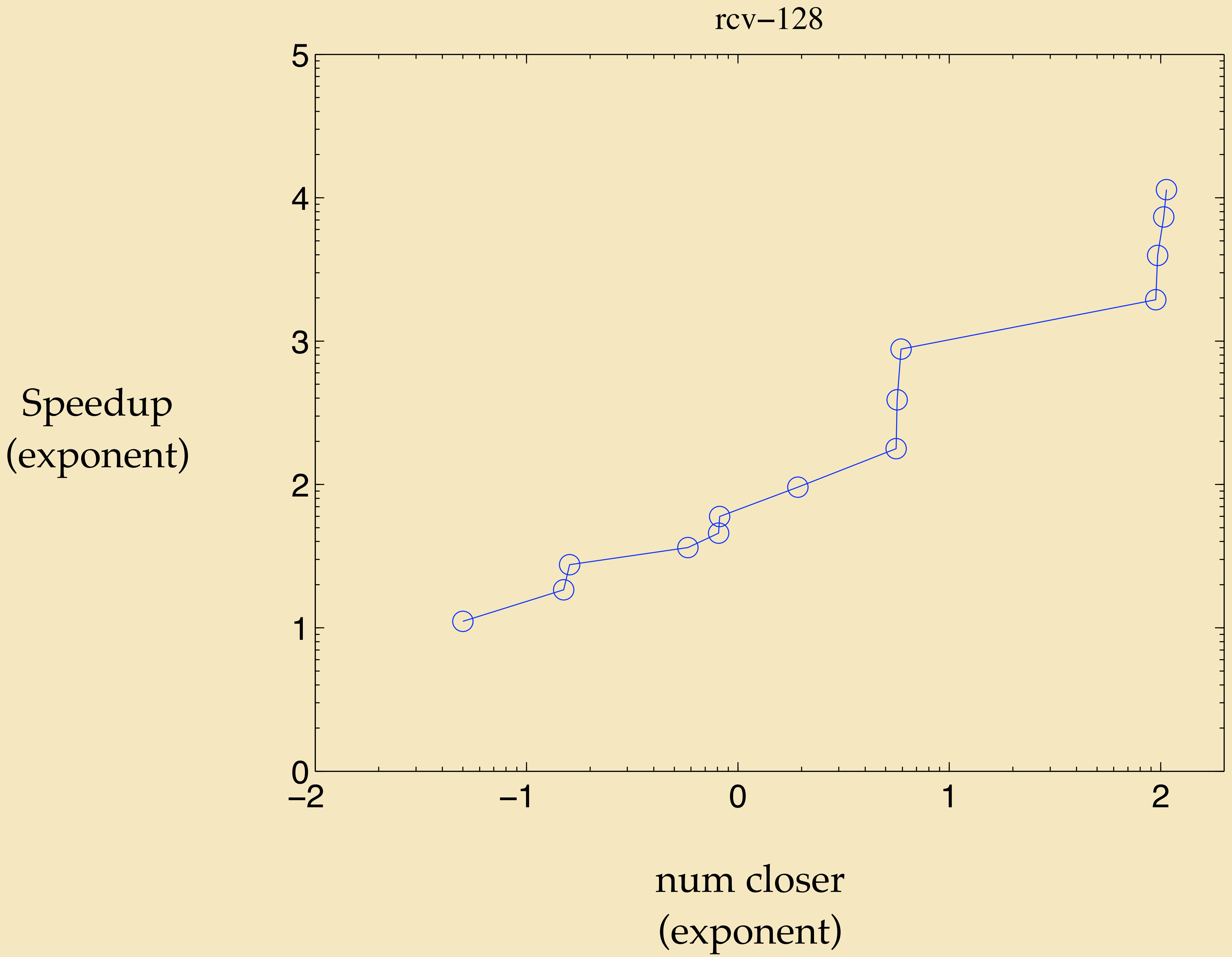
Evaluation

Speedup over brute-force search in execution time.

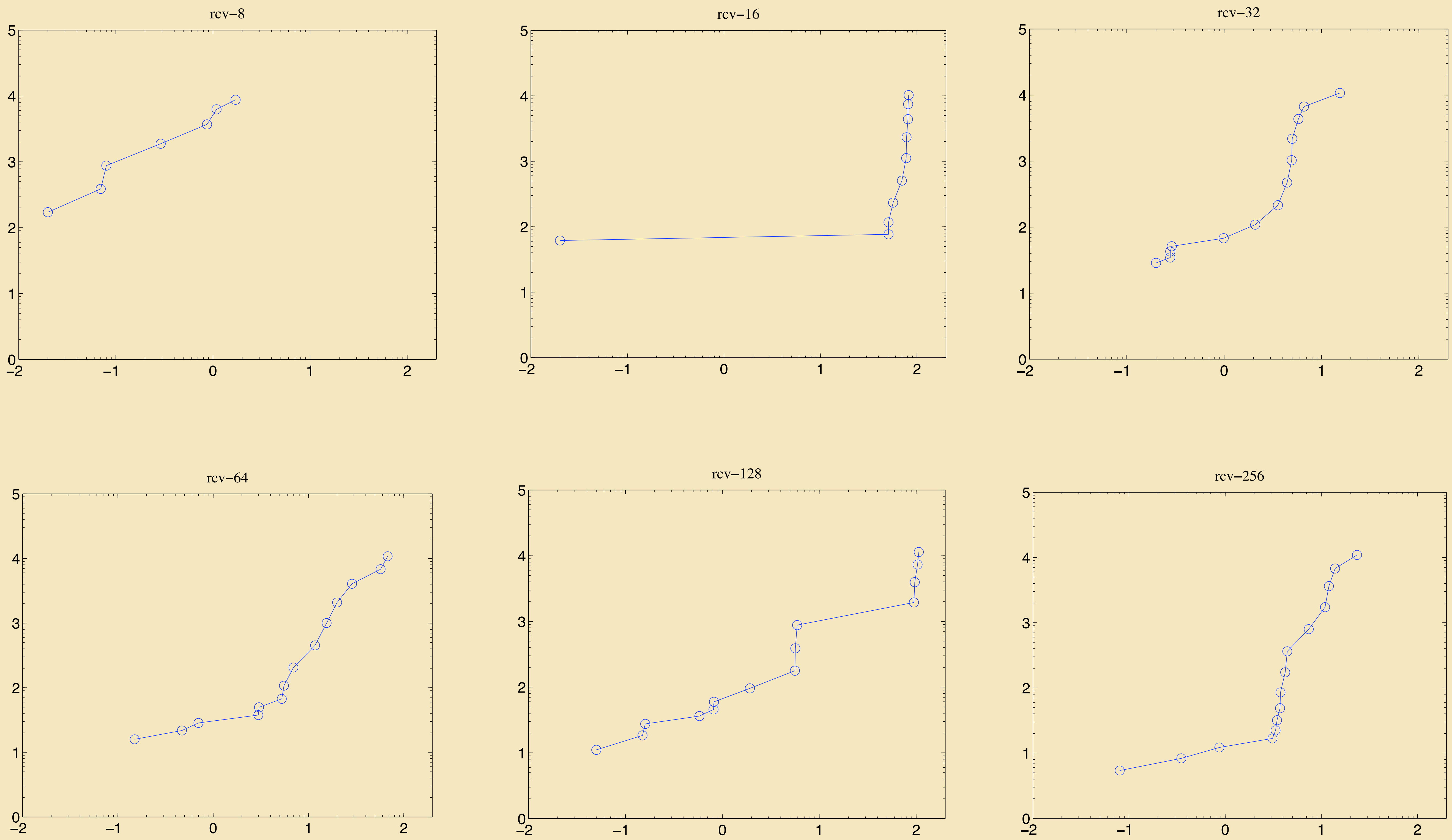
VS

NC for *number closer*: how many closer points are there? *e.g.* if NC=3, the bbtree returned the fourth NN.

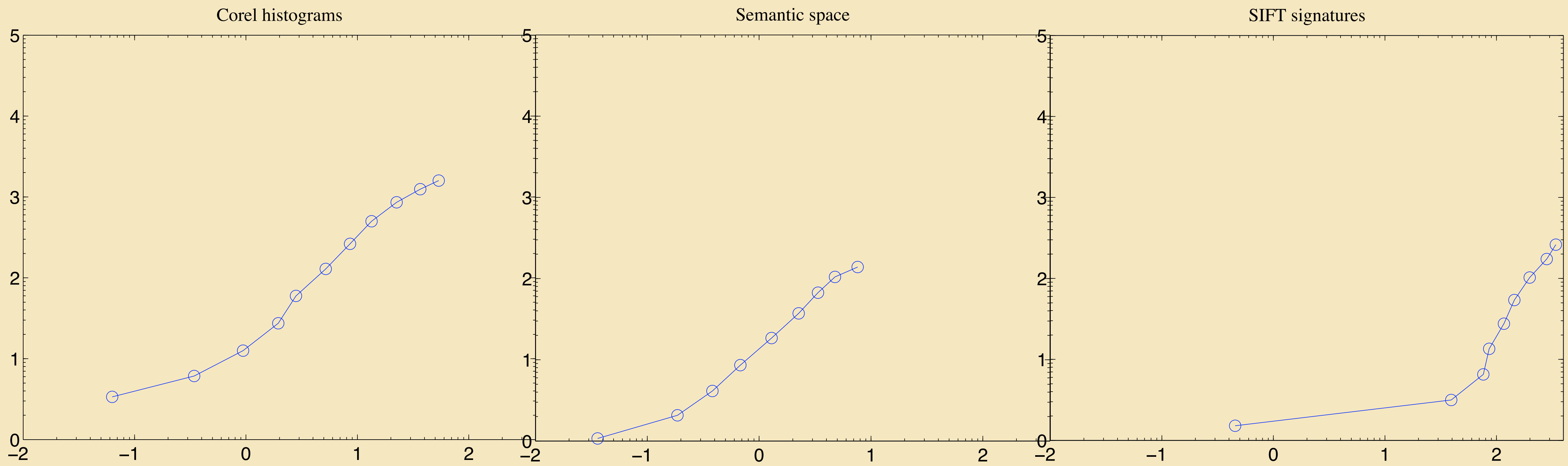
Approximate search



rcv data



corel, semantic space, SIFT



Exact search

dataset	**dimensionality**	**speedup**
rcv-8	8	64.5
rcv-16	16	36.7
rcv-32	32	21.9
rcv-64	64	12.0
corel histograms	64	2.4
rcv-128	128	5.3
rcv-256	256	3.3
semantic space	371	1.0
SIFT signatures	1111	0.9

Thanks..

- Serge Belongie
- Sanjoy Dasgupta
- Charles Elkan
- Carolina Galleguillos
- Daniel Hsu
- Nikhil Rasiwasia
- Lawrence Saul