Statistical Learning and Optimization Based on Comparative Judgments

OSL, Les Houches, January 10, 2013

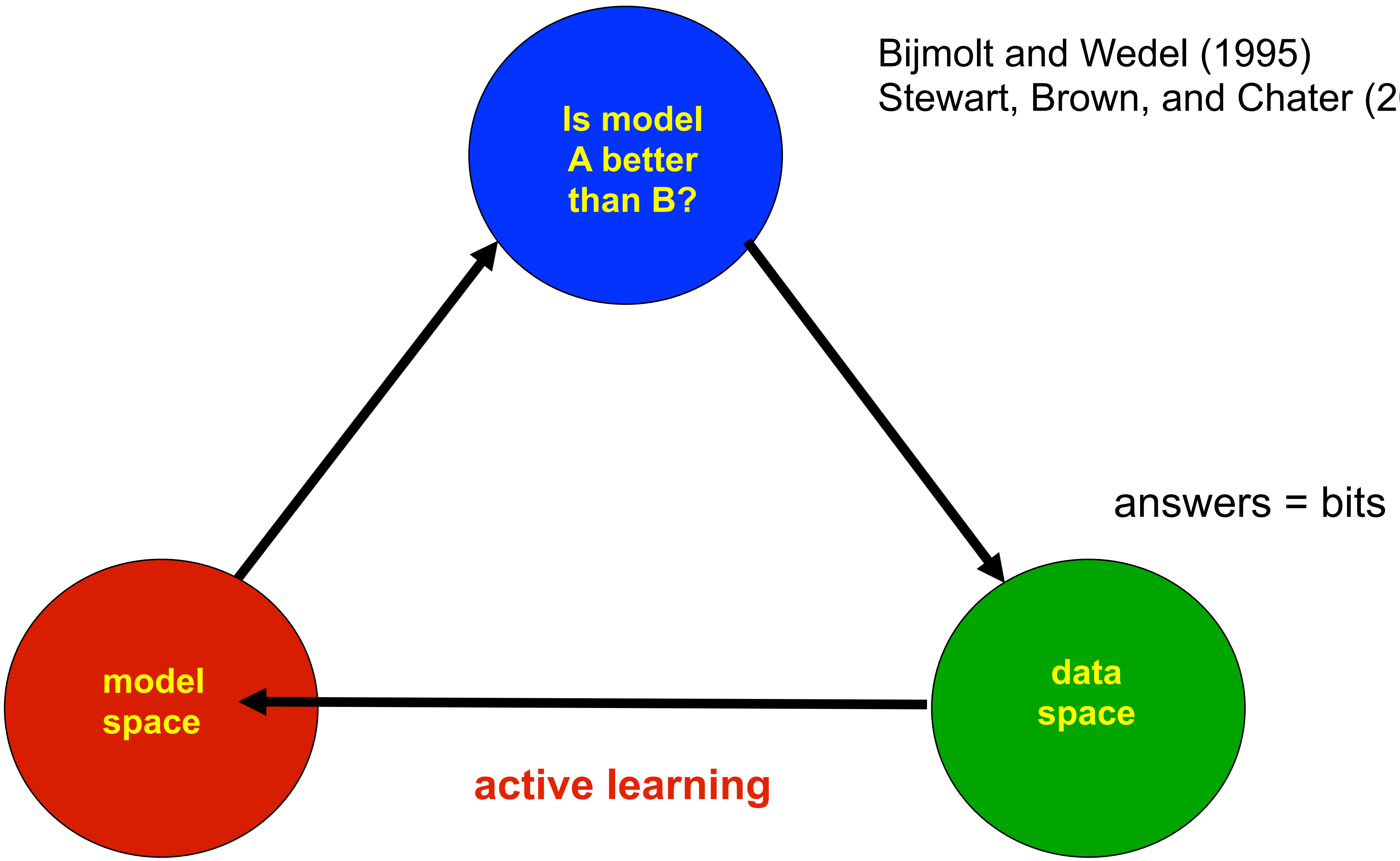
Rob Nowak www.ece.wisc.edu/~nowak

Learning from Comparative Judgements

L. L. Thurstone

Humans are much more reliable and consistent at making comparative judgements, than at giving numerical ratings or evaluations

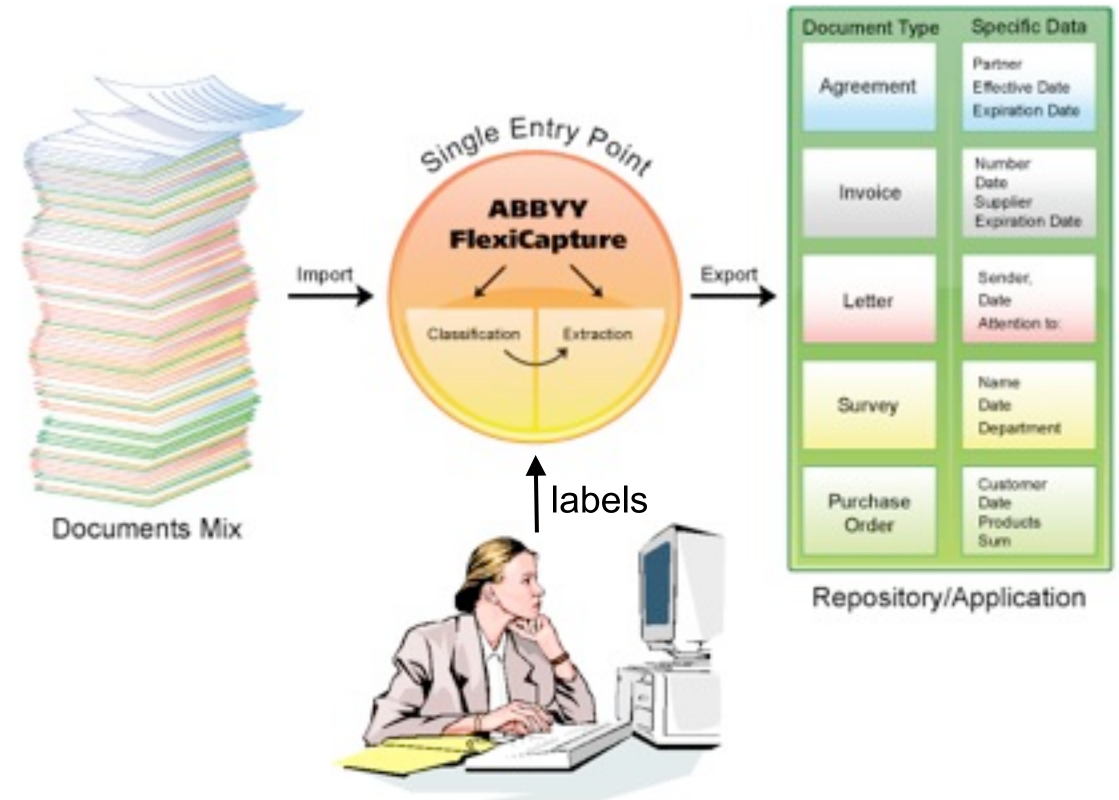
Bijmolt and Wedel (1995)
Stewart, Brown, and Chater (2005)



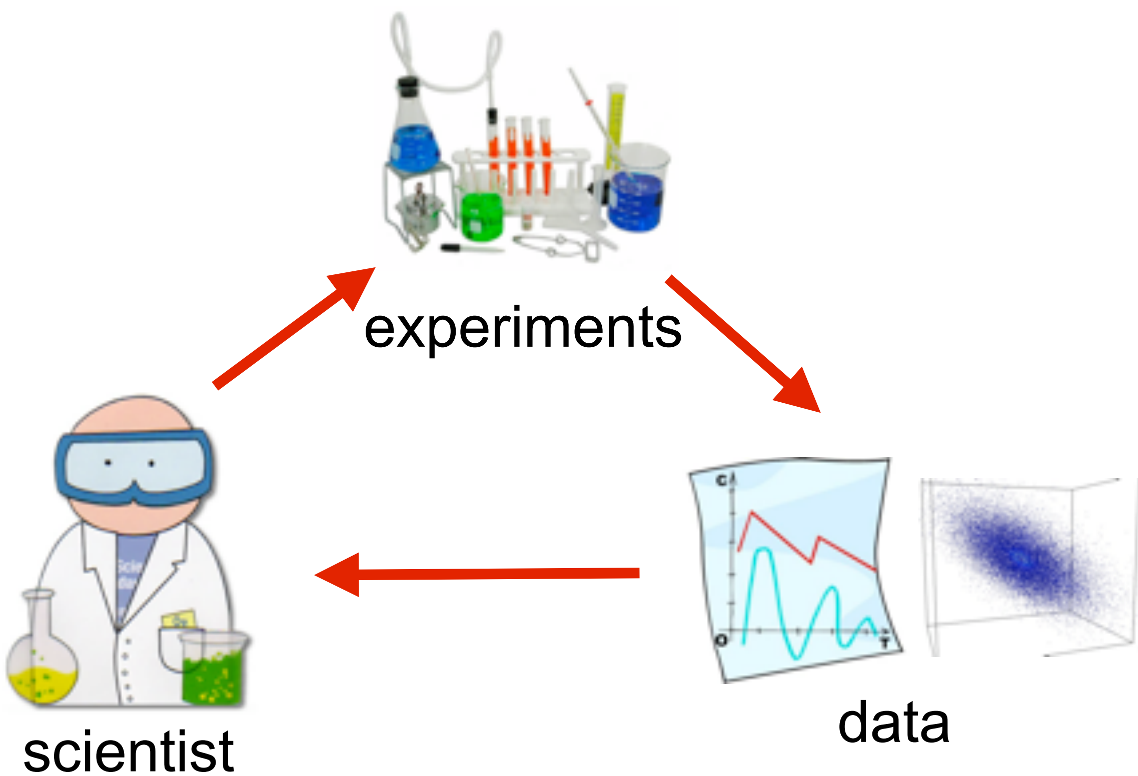
Machine Learning from Human Judgements

Recommendation Systems

Document Classification



Optimizing Experimentation



Challenge:

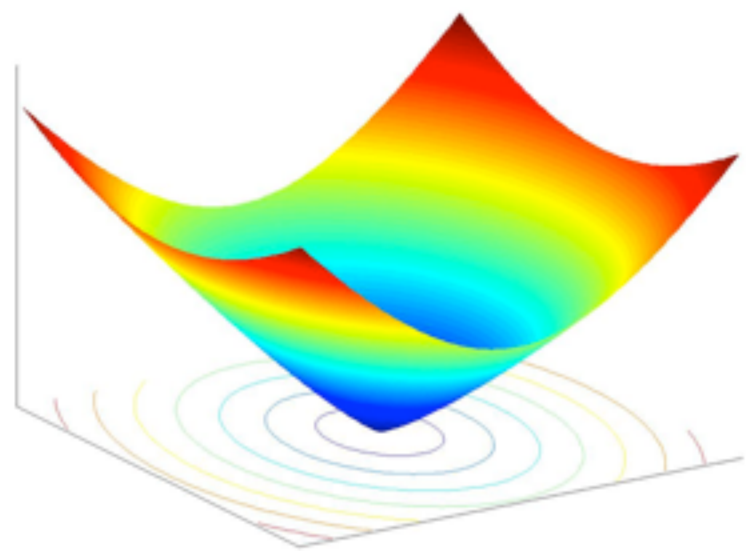
Computing is cheap, but human assistance/guidance is expensive

Goal:

Optimize such systems with as little human involvement as possible

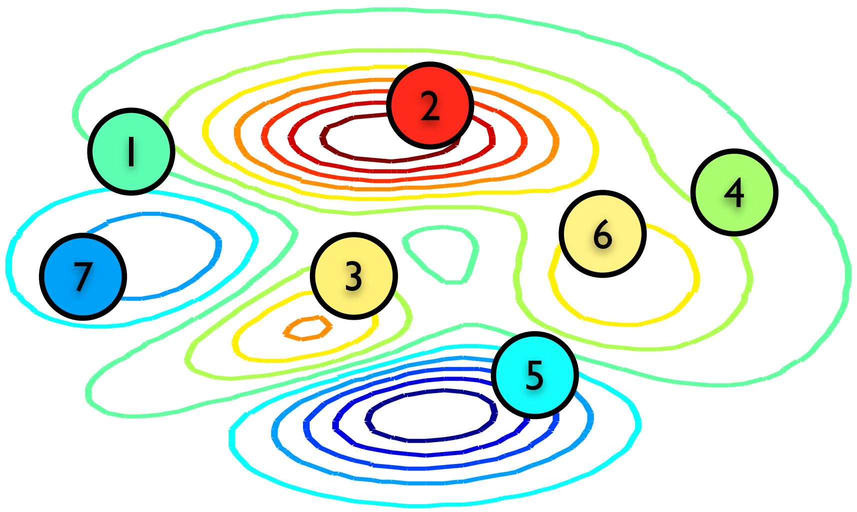
Learning from Paired Comparisons

1. Derivative Free Optimization using Human Subjects



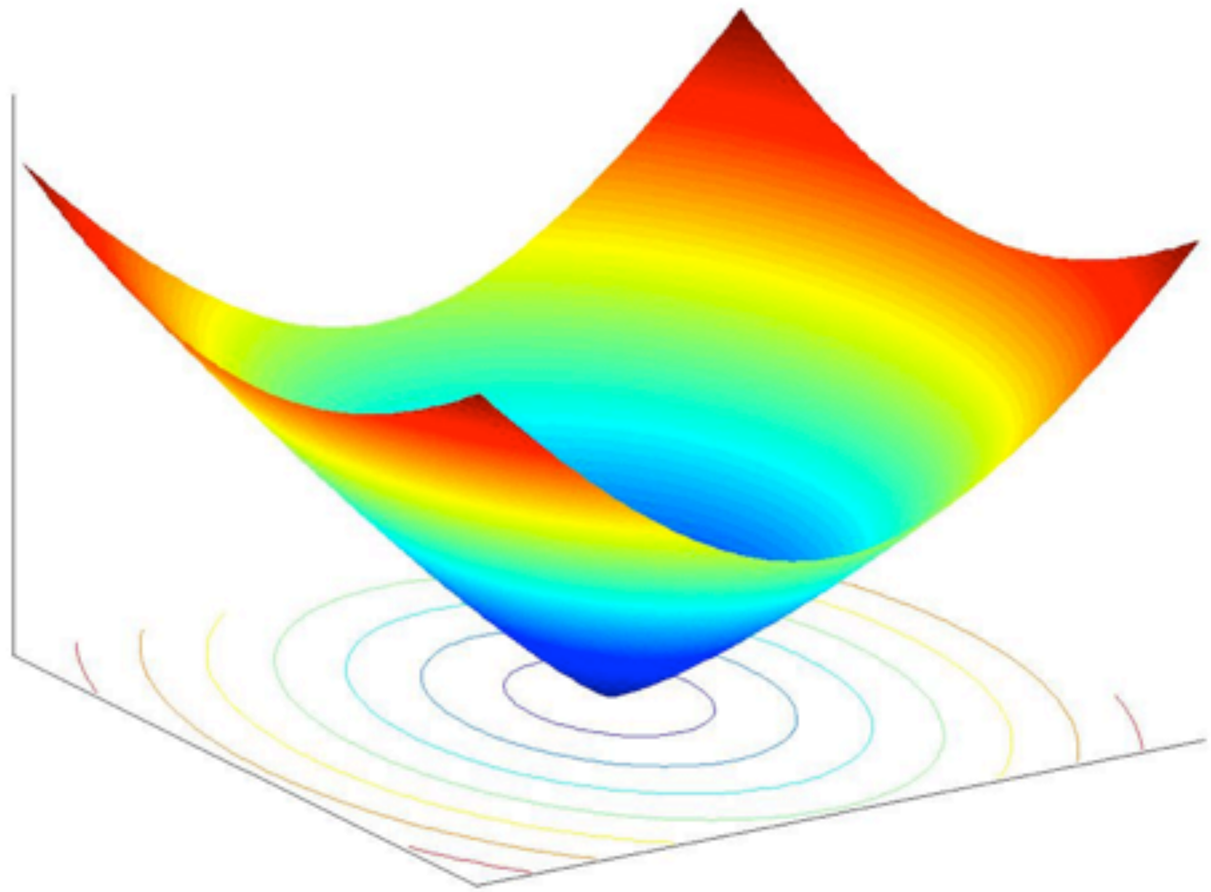
minimizing a convex function

2. Ranking from Pairwise Comparisons



ranking objects that embed into a low-dimensional space

Optimization Based on Human Judgements

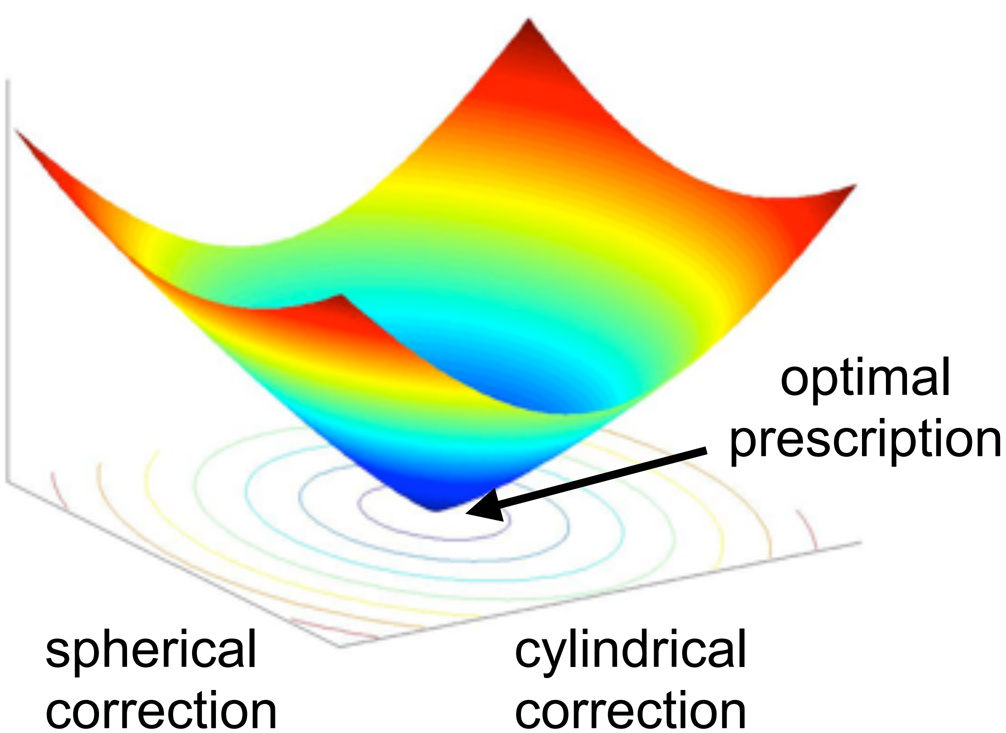


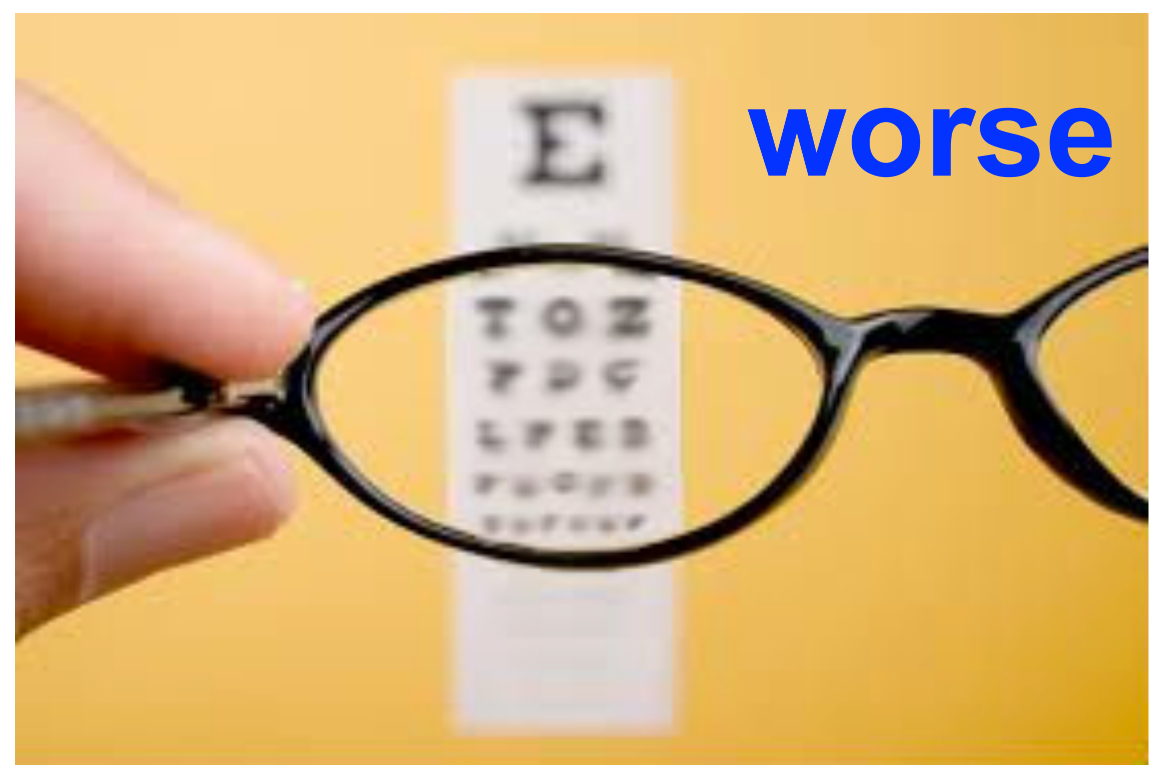
convex function to be minimized

Human oracles can provide function values or comparisons, but not function gradients

Methods that don't use gradients are called Derivative Free Optimization (DFO)

A Familiar Application





Personalized Search

Profile vector $w_A \in \mathbb{R}^d$

$$\text{Results} \leftarrow \text{SEARCH}(\text{query} = \text{“}\textit{sebastian bach}\text{”}, w_A)$$

$w_A = w_{\text{old}}$

Sebastian Bach

Johann Sebastian Bach (1685-1750)

- Composer

$w_A = w_{\text{new}}$

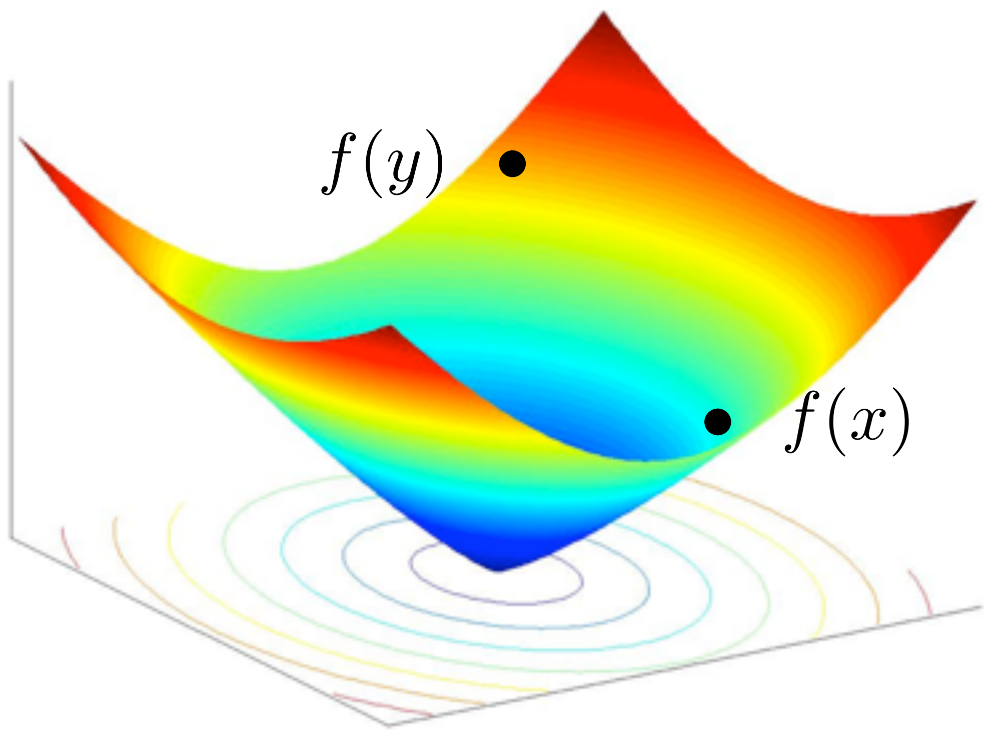
Sebastian Bach

Sebastian Bach (1968-current)

- Heavy Metal Singer
- Frontman of “Skid Row”

Optimization Based on Pairwise Comparisons

Assume that the (unknown) function f to be optimized is strongly convex with Lipschitz gradients



The function will be minimized by asking pairwise comparisons of the form:

$$\text{Is } f(x) \;>\; f(y) \quad ?$$

Assume that the answers are probably correct: for some $\delta > 0$

$$\mathbb{P}\left(\text{answer} = \operatorname{sign}(f(x) - f(y))\right) \;\geq\; \frac{1}{2} \;+\; \delta$$

Optimization based on Pairwise Comparisons

Optimization with Pairwise Comparisons

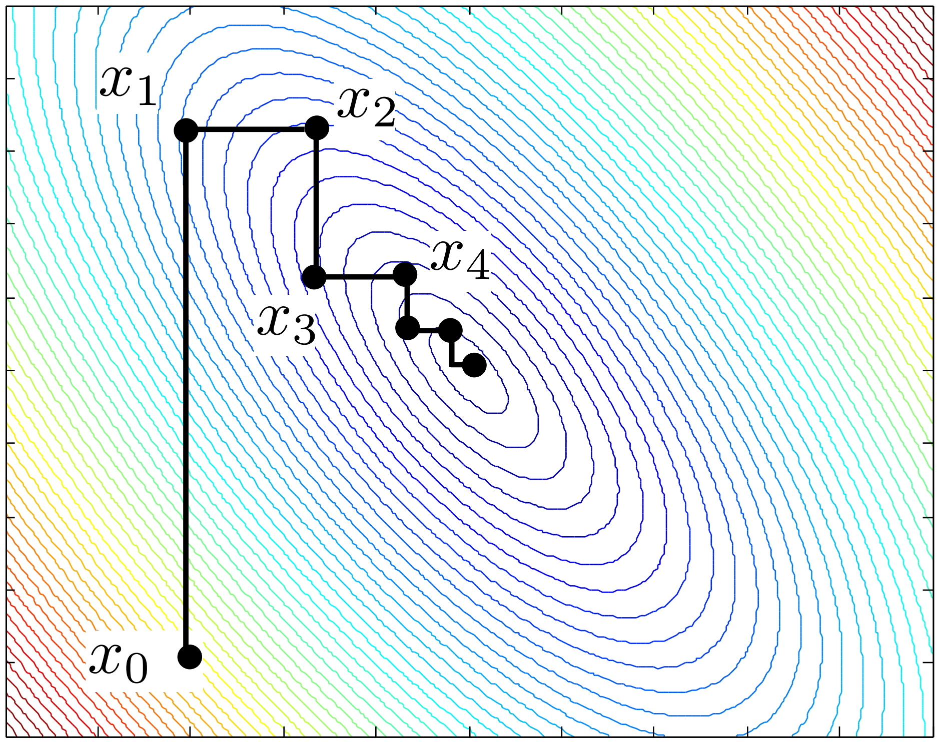
initialize: $x_0 =$ random point

for $n = 0, 1, 2, \ldots$

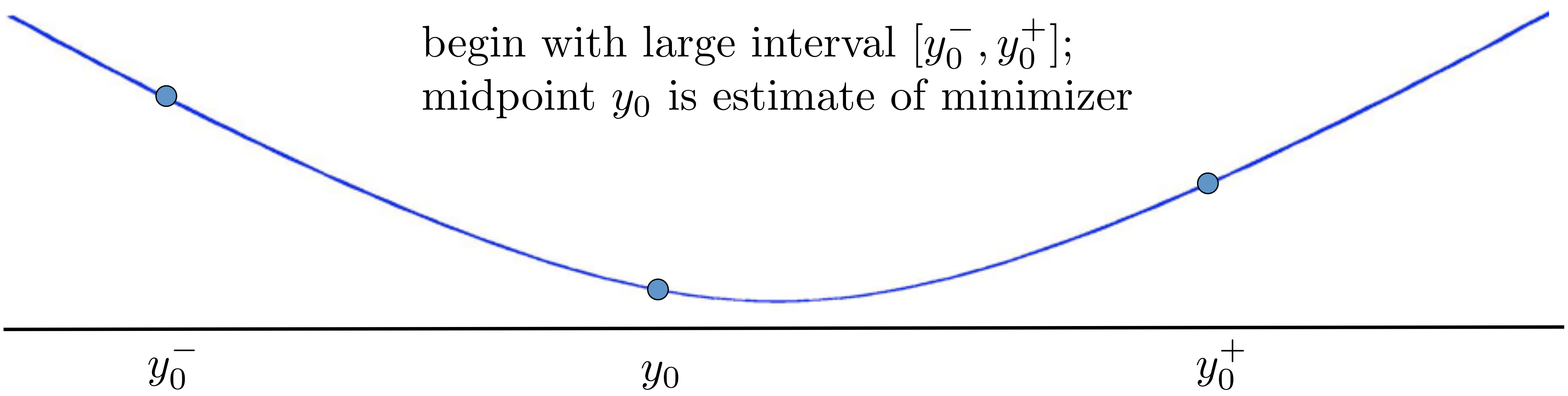
1) select one of d coordinates uniformly at random and consider line along coordinate that passes x_n

2) minimize along coordinate using pairwise comparisons and binary search

3) $x_{n+1} =$ approximate minimizer



line search iteratively reduces interval containing minimum



Optimization based on Pairwise Comparisons

Optimization with Pairwise Comparisons

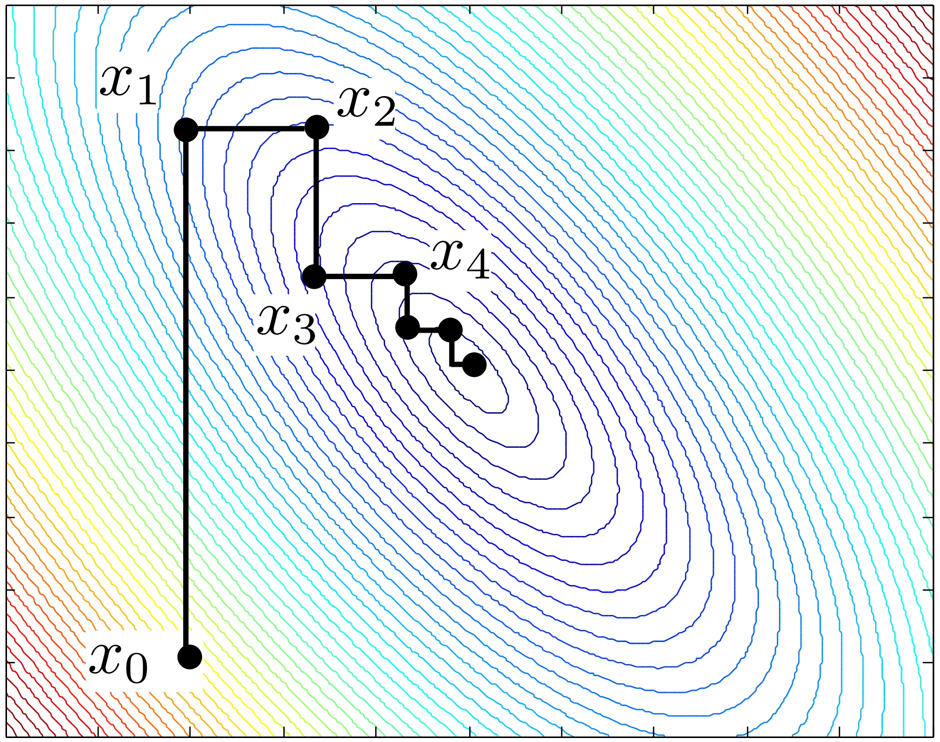
initialize: x_0 = random point

for $n = 0, 1, 2, \ldots$

1) select one of d coordinates uniformly at random and consider line along coordinate that passes x_n

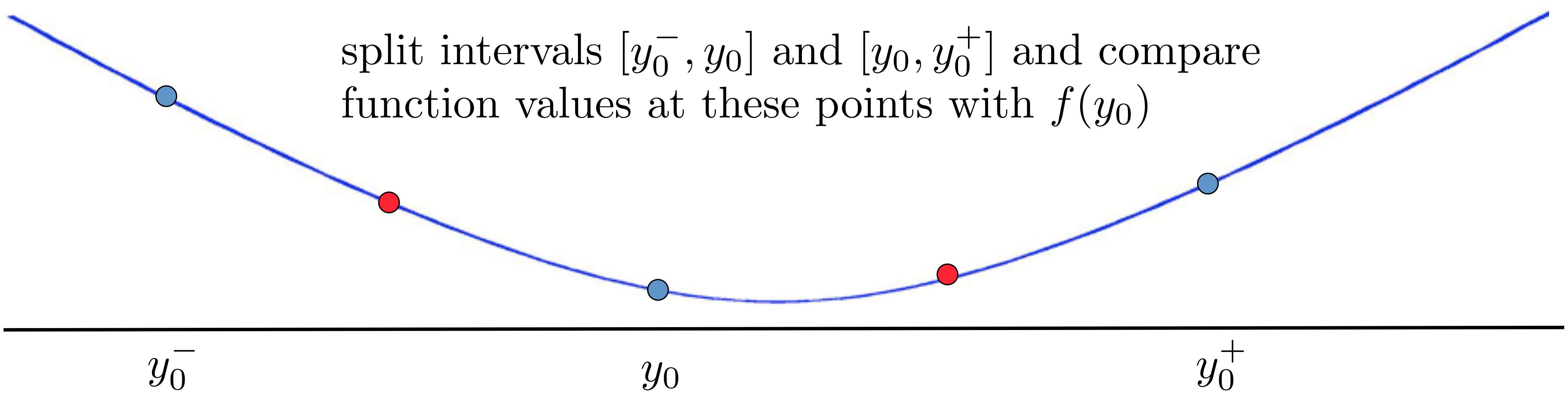
2) minimize along coordinate using pairwise comparisons and binary search

3) x_{n+1} = approximate minimizer



line search iteratively reduces interval containing minimum

split intervals $[y_0^-, y_0]$ and $[y_0, y_0^+]$ and compare function values at these points with $f(y_0)$



Optimization based on Pairwise Comparisons

Optimization with Pairwise Comparisons

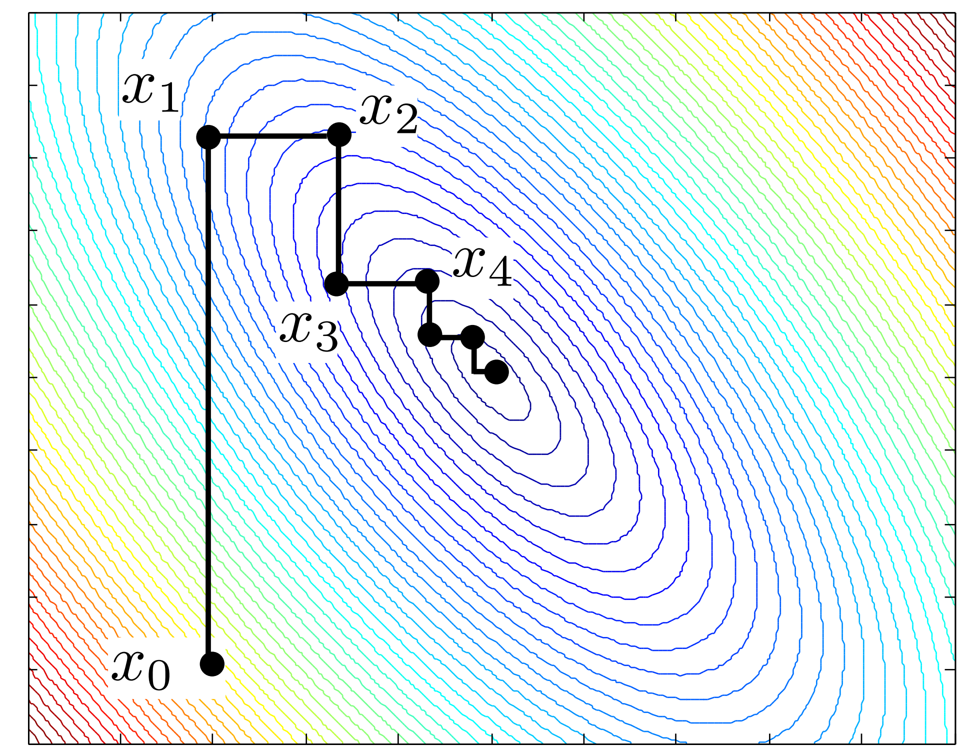
initialize: $x_0 =$ random point

for $n = 0, 1, 2, \ldots$

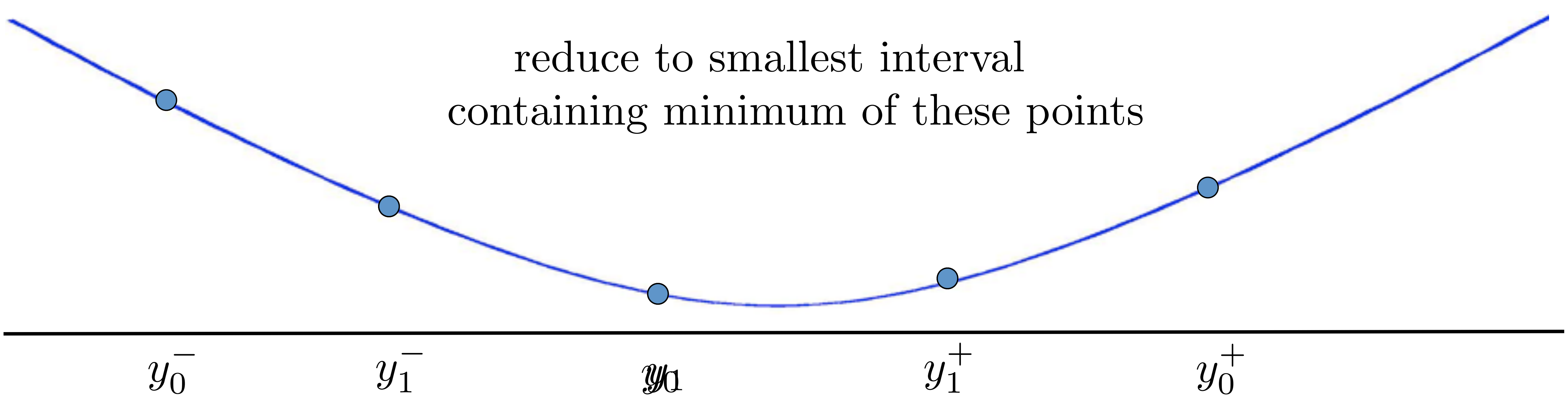
1) select one of d coordinates uniformly at random and consider line along coordinate that passes x_n

2) minimize along coordinate using pairwise comparisons and binary search

3) $x_{n+1} =$ approximate minimizer



line search iteratively reduces interval containing minimum



Optimization based on Pairwise Comparisons

Optimization with Pairwise Comparisons

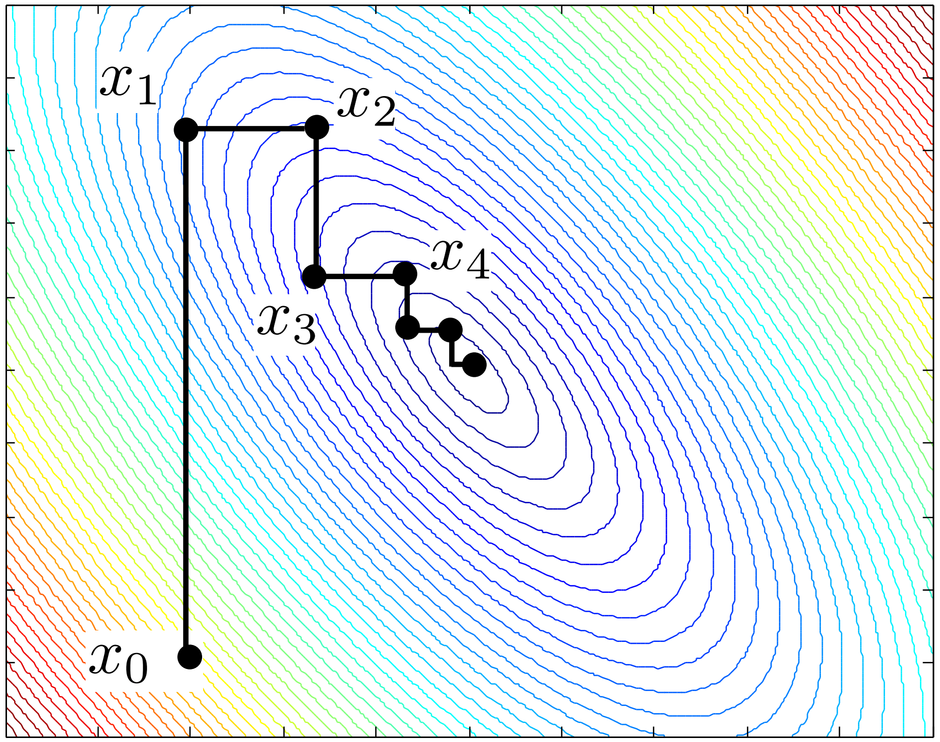
initialize: x_0 = random point

for $n = 0, 1, 2, \ldots$

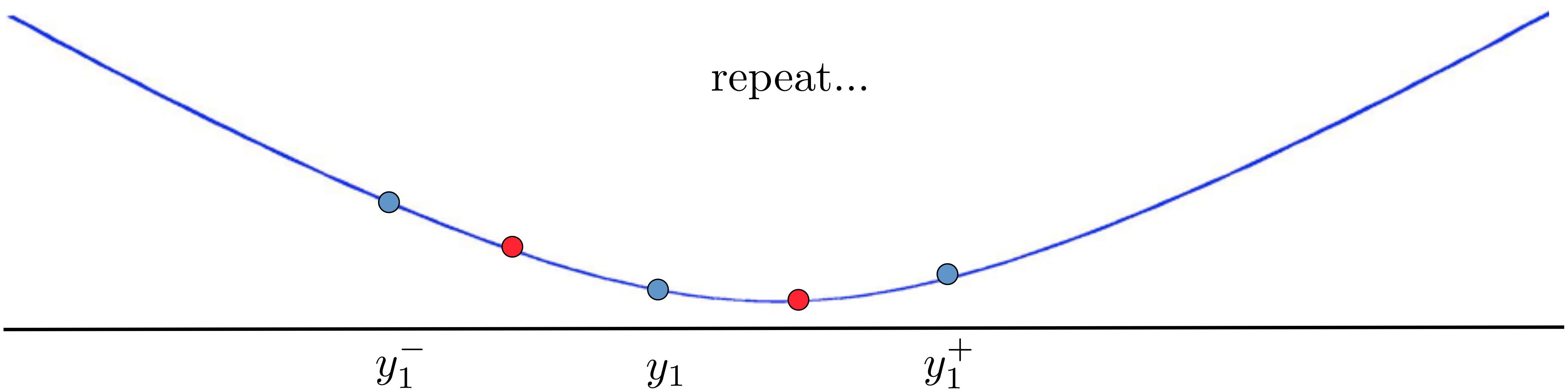
1) select one of d coordinates uniformly at random and consider line along coordinate that passes x_n

2) minimize along coordinate using pairwise comparisons and binary search

3) x_{n+1} = approximate minimizer



line search iteratively reduces interval containing minimum



Optimization based on Pairwise Comparisons

Optimization with Pairwise Comparisons

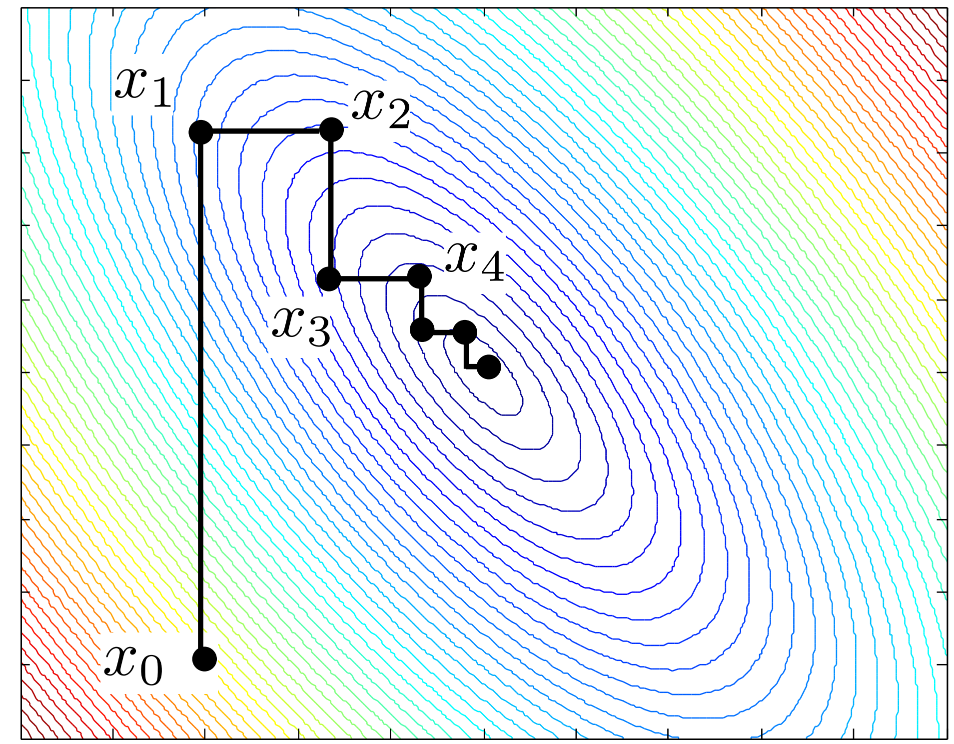
initialize: $x_0 =$ random point

for $n = 0, 1, 2, \ldots$

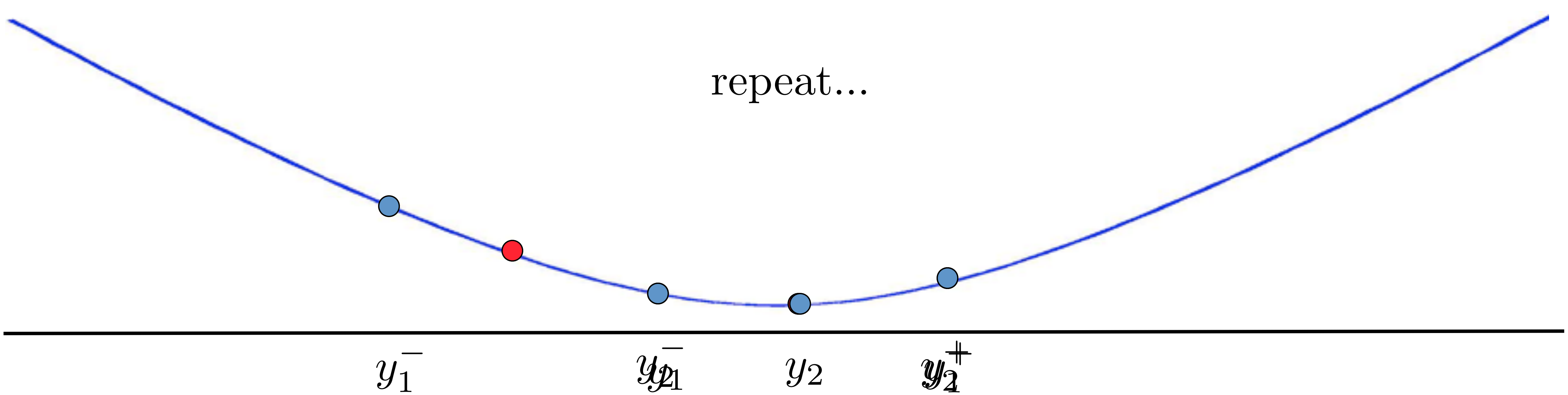
1) select one of d coordinates uniformly at random and consider line along coordinate that passes x_n

2) minimize along coordinate using pairwise comparisons and binary search

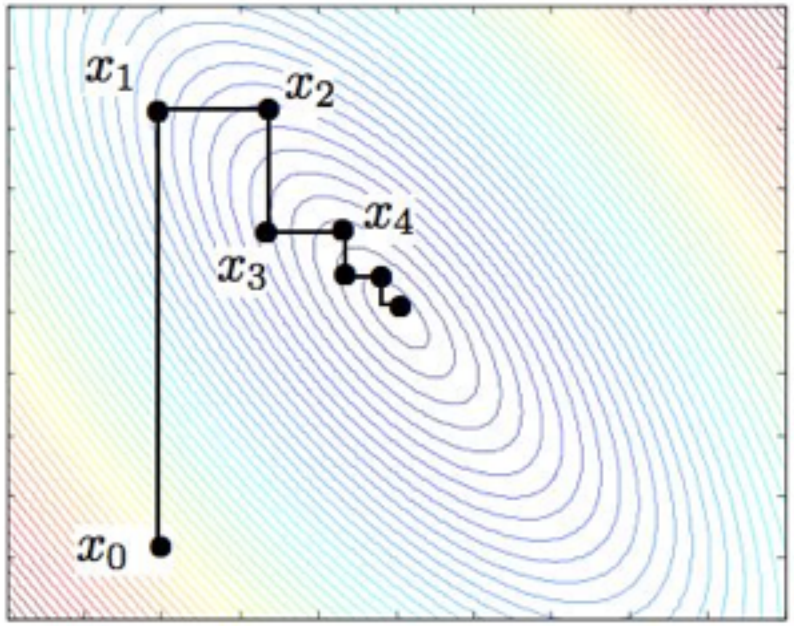
3) $x_{n+1} =$ approximate minimizer



line search iteratively reduces interval containing minimum



Convergence Analysis



If we want error $:= \mathbb{E}[f(x_k) - f(x^*)] \leq \epsilon$, we must solve $k \approx d \log \frac{1}{\epsilon}$ line searches (standard coordinate descent bound) and each must be at least $\sqrt{\frac{\epsilon}{d}}$ accurate

Noiseless Case:

each line search requires $\frac{1}{2}\log(\frac{d}{\epsilon})$ comparisons

$\Rightarrow$ total of $n \approx d \log \frac{1}{\epsilon} \log \frac{d}{\epsilon}$ comparisons

$\Rightarrow \epsilon \approx \exp\left(-\sqrt{\frac{n}{d}}\right)$

Noisy Case: probably correct answers to comparisons:

$$\mathbb{P}\left(\text{answer} = \text{sign}(f(x) - f(y))\right) \geq \frac{1}{2} + \delta$$

take majority vote of repeated comparisons to mitigate noise

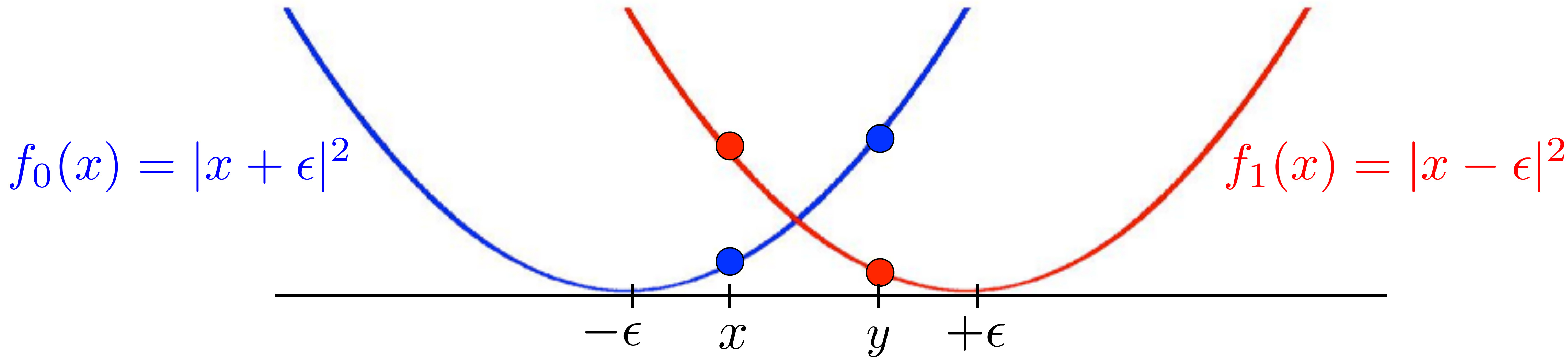
Bounded Noise ($\delta \geq \delta_0 > 0$):

line searches require $C \log \frac{d}{\epsilon}$ comparisons,
where $C > 1/2$ depends on $\delta_0 \Rightarrow \epsilon \approx \exp\left(-\sqrt{\frac{n}{d\,C}}\right)$

Unbounded Noise ($\delta \propto |f(x) - f(y)|$):

line searches require $\left(\frac{d}{\epsilon}\right)^2$ comparisons $\Rightarrow \epsilon \approx \sqrt{\frac{d^3}{n}}$

Lower Bounds



For unbounded noise, $\delta \propto |f(x) - f(y)|$, Kullback-Leibler Divergence between response to $f_0(x) > f_0(y)$? vs. $f_1(x) > f_1(y)$? is $O(\epsilon^4)$, and KL Divergence between n responses is $O(n\epsilon^4)$

with $\epsilon \sim n^{-1/4}$

- KL Divergence = constant
- squared distance between minima $\sim n^{-1/2}$

$$\Rightarrow \ \mathbb{P}\left(f(x_n) - f(x^*) \geq n^{-1/2}\right) \ \geq \ \text{constant}$$

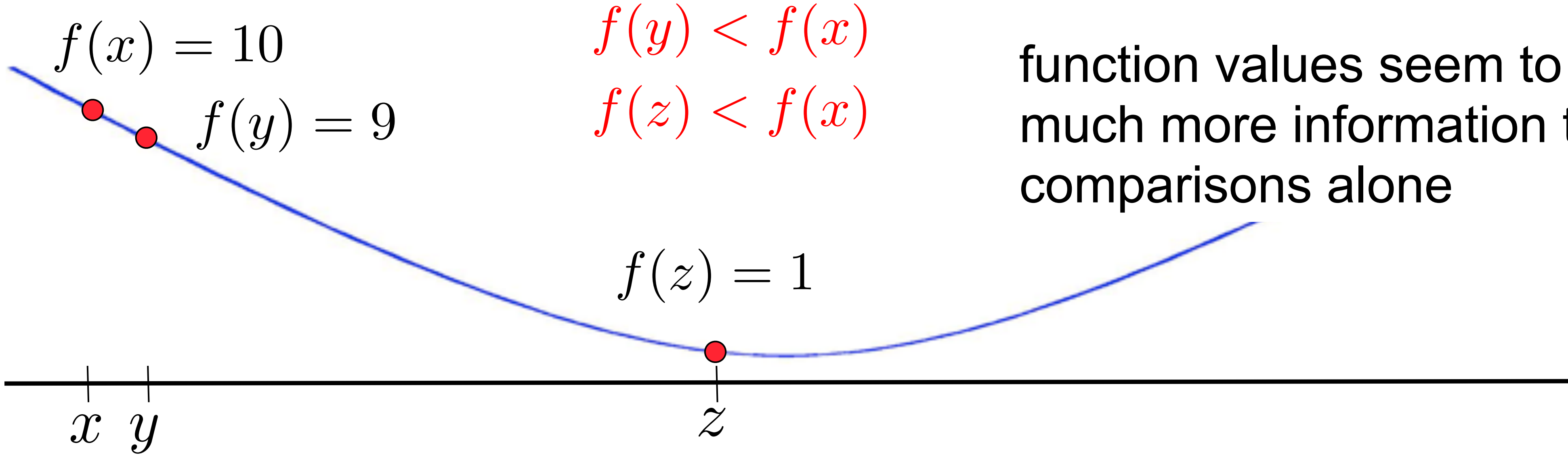
matches $O(n^{-1/2})$ upper bound of algorithm $\qquad \sqrt{\frac{d}{n}}$ in $\mathbb{R}^d$

Jamieson, Recht, RN (2012)

A Surprise

Could we do better with function evaluations (e.g., ratings instead of comparisons)?

suppose we can obtain noisy function evaluations of the form: $f(x) + \text{noise}$



$f(y) < f(x)$

$f(z) < f(x)$

function values seem to provide much more information than comparisons alone

lower bound on optimization error with noisy function evaluations $\sqrt{\frac{d^2}{n}}$

evaluations give at best a small improvement over comparisons

O. Shamir (2012)

upper bound on optimization error with noisy pairwise comparisons $\sqrt{\frac{d^3}{n}}$

see Agrawal, Dekel, Xiao (2010)
for similar upper bounds for function evals

if we could measure noisy gradients (and function is strongly convex), then $O(\frac{d}{n})$ convergence rate is possible

Nemirovski et al 2009

Preference Learning

Bartender: “What beer would you like?”

Philippe: “Hmm... I prefer French wine”

Bartender: “Try these two samples. Do you prefer A or B?”

Philippe: “B”

Bartender: “Ok try these two: C or D?” ....

Ranking Based on Pairwise Comparisons

Consider 10 beers ranked from best to worst: D < G < I < C < J < E < A < H < B < F

	A	B	C	D	E	F	G	H	I	J
A	0	1	-1	-1	-1	1	-1	1	-1	-1
B	-1	0	-1	-1	-1	1	-1	-1	-1	-1
C	1	1	0	-1	1	1	-1	1	-1	1
D	1	1	1	0	1	1	1	1	1	1
E	1	1	-1	-1	0	1	-1	1	-1	-1
F	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
G	1	1	1	-1	1	1	0	1	1	1
H	-1	1	-1	-1	-1	1	-1	0	-1	-1
I	1	1	1	-1	1	1	-1	1	0	1
J	1	1	-1	-1	1	1	-1	1	-1	0

Which pairwise comparisons should we ask? How many are needed?

Assumption: responses to pairwise comparisons are consistent with ranking

Ranking Based on Pairwise Comparisons

Consider 10 beers ranked from best to worst:

D < G < I < C < J < E < A < H < B < F

	A	B	C	D	E	F	G	H	I	J
A	0	1	-1	-1	-1	1	-1	1	-1	-1
B	-1	0	-1	-1	-1	1	-1	-1	-1	-1
C	1	1	0	-1	1	1	-1	1	-1	1
D	1	1	1	0	1	1	1	1	1	1
E	1	1	-1	-1	0	1	-1	1	-1	-1
F	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
G	1	1	1	-1	1	1	0	1	1	1
H	-1	1	-1	-1	-1	1	-1	0	-1	-1
I	1	1	1	-1	1	1	-1	1	0	1
J	1	1	-1	-1	1	1	-1	1	-1	0

select m pairwise comparisons **at random**

perfect recovery: almost all pairs must be compared, i.e., about $n(n-1)/2$ comparisons

approximate recovery: fraction of pairs misordered $\leq \dfrac{c\, n \log n}{m}$

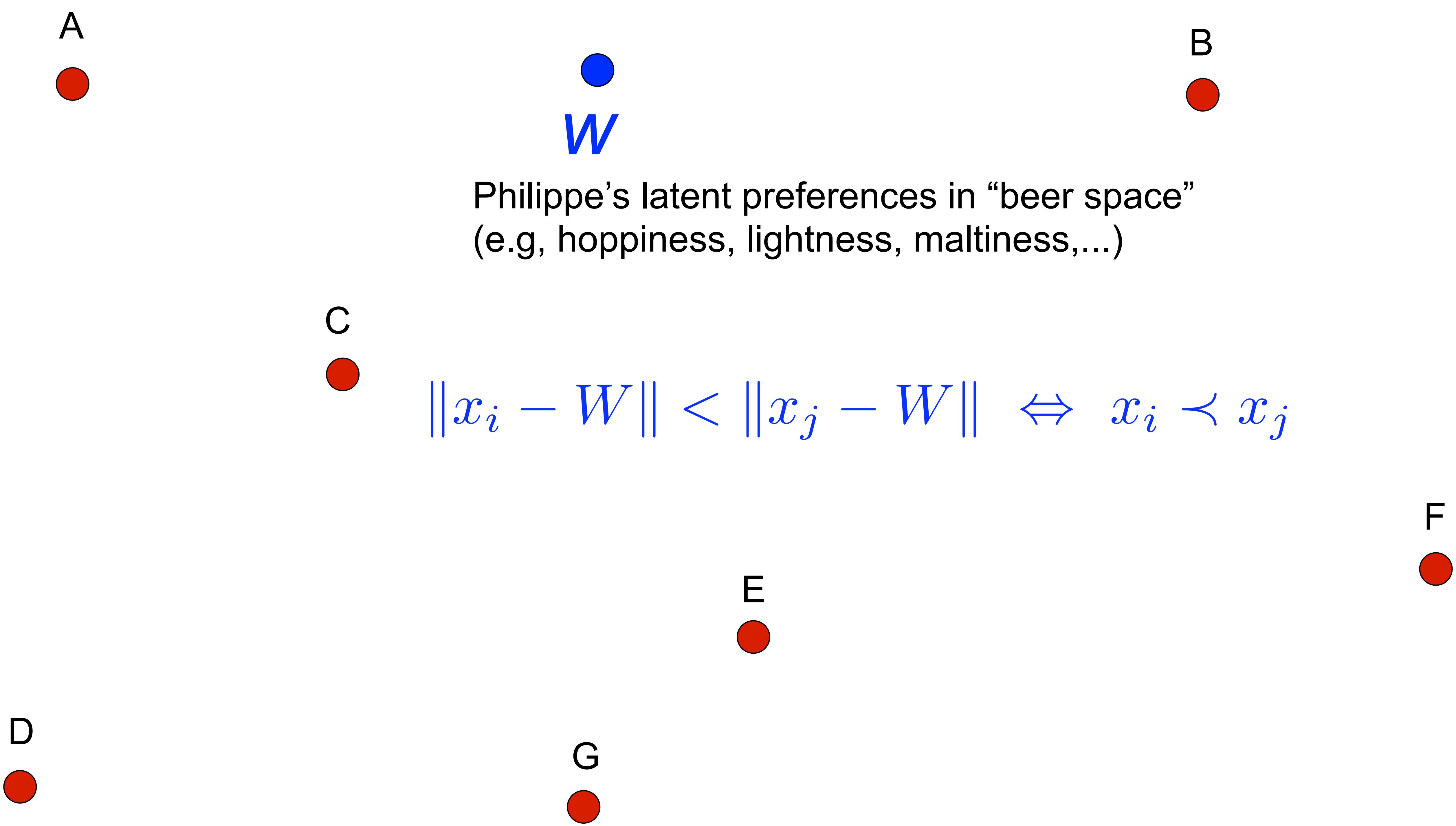
adaptive selection: binary insertion sort also requires $n \log n$ comparisons

That's a lot of beer!

Problem: $n!$ possible rankings requires $n \log n$ bits of information

Low-Dimensional Assumption: Beer Space

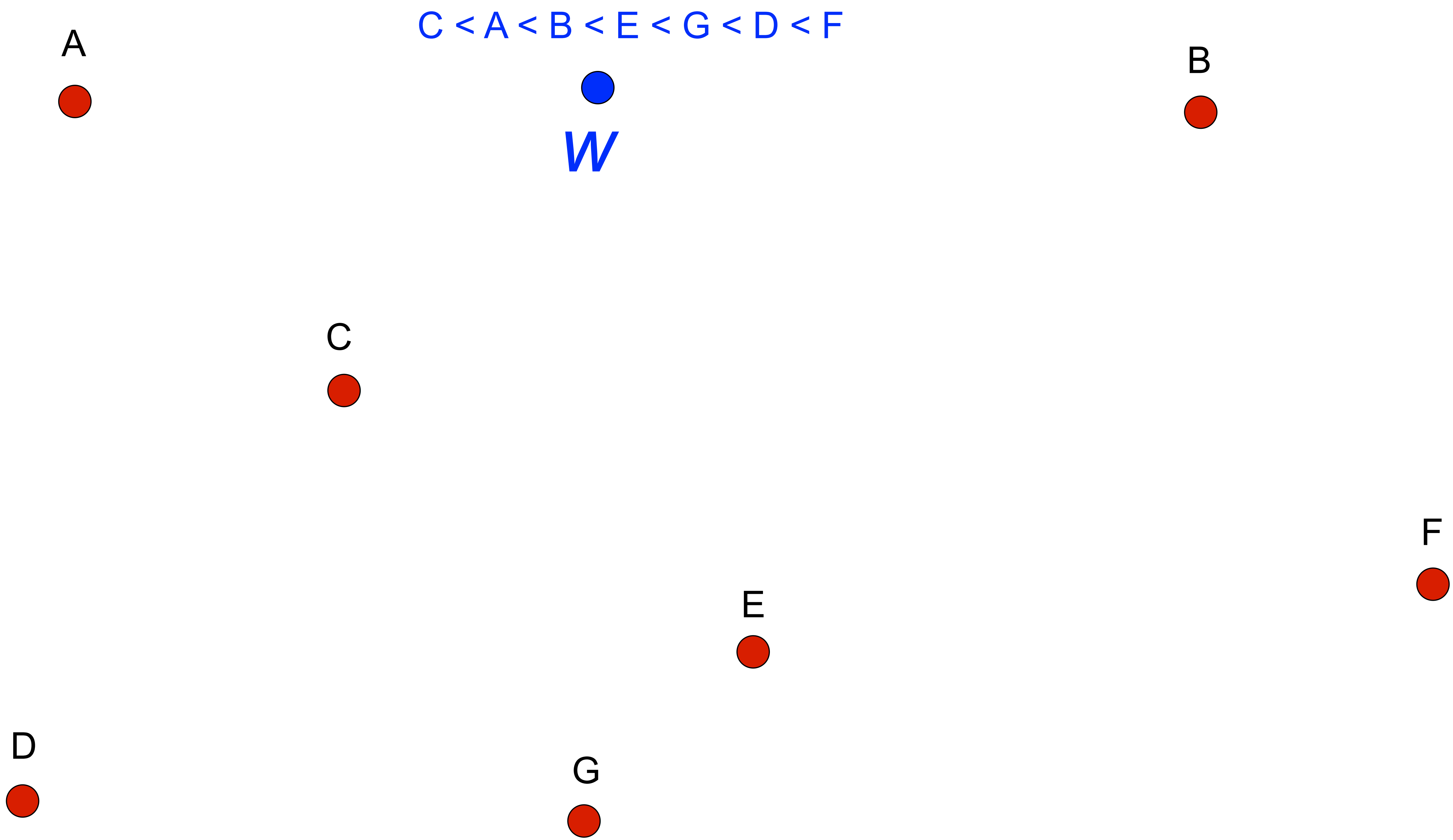
Suppose beers can be embedded (according to characteristics) into a low-dimensional Euclidean space.



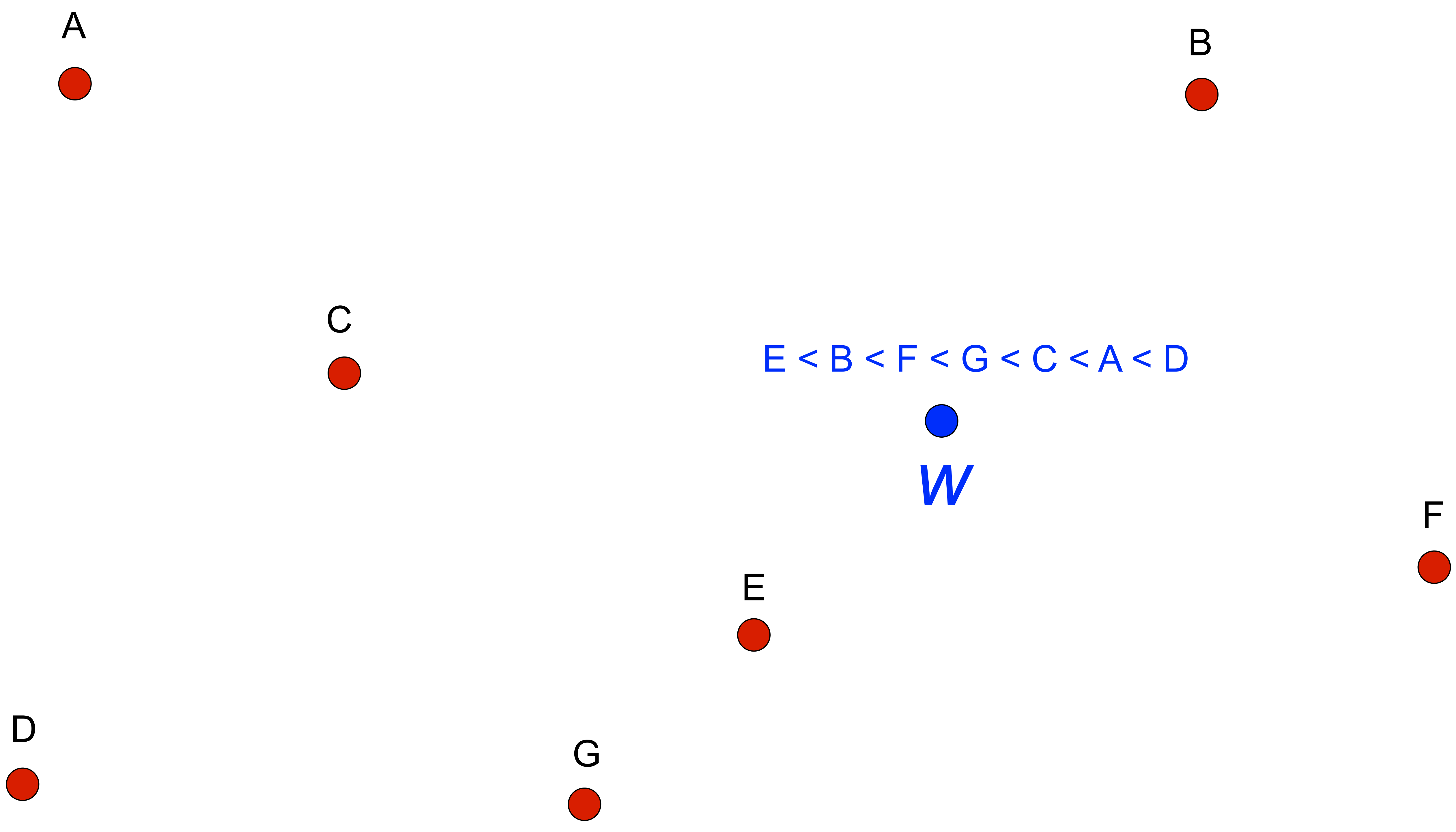
Philippe's latent preferences in "beer space"
(e.g, hoppiness, lightness, maltiness,...)

$$\|x_i - W\| < \|x_j - W\| \;\Leftrightarrow\; x_i \prec x_j$$

Ranking According to Distance



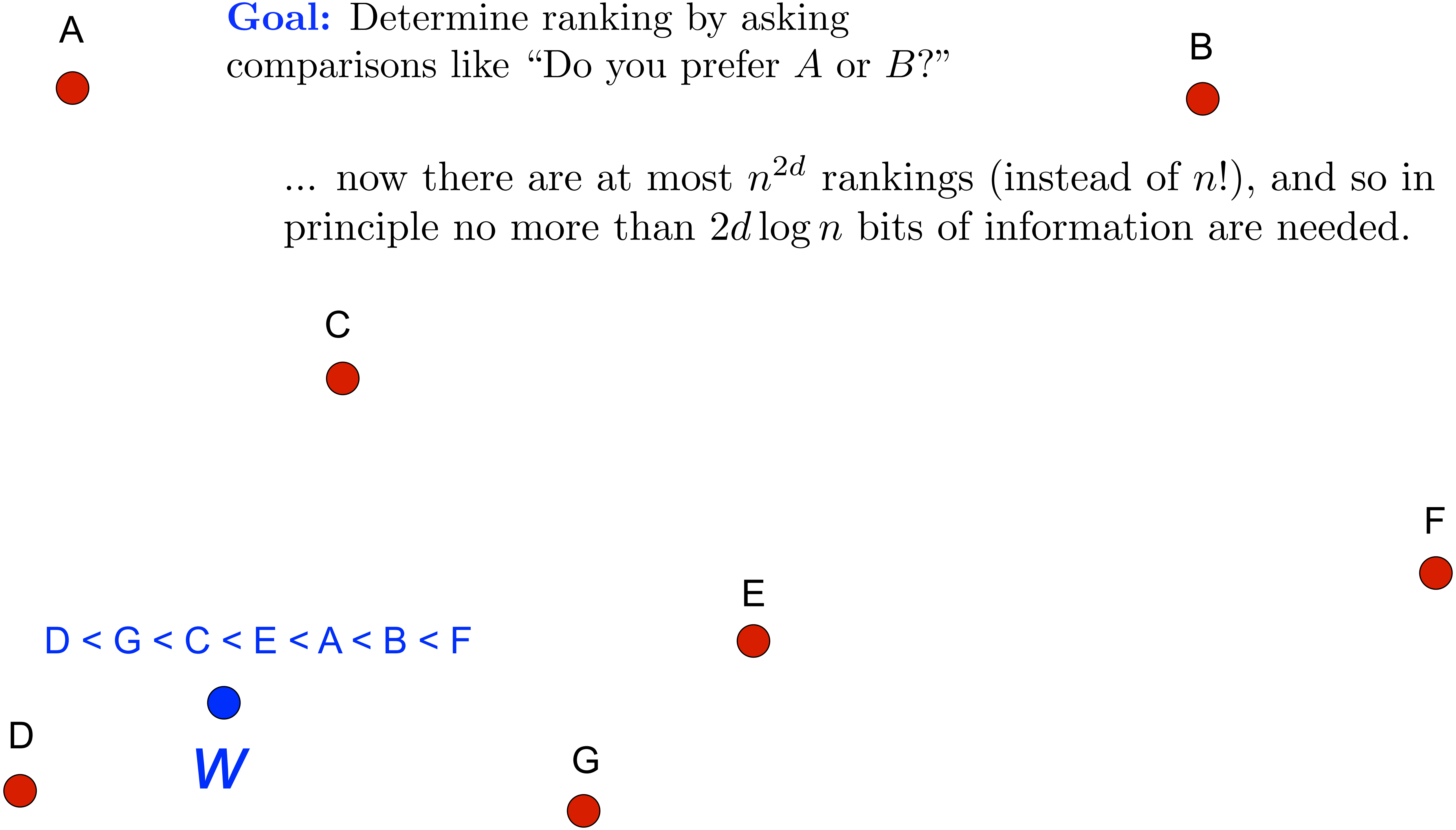
Ranking According to Distance



Ranking According to Distance

Goal: Determine ranking by asking comparisons like "Do you prefer A or B?"

... now there are at most n^{2d} rankings (instead of $n!$), and so in principle no more than $2d \log n$ bits of information are needed.



Optimization

Consider n objects $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$. Many comparisons are redundant because the objects embed in $\mathbb{R}^d$, and therefore it may be possible to correctly rank based on a small subset.

binary information we can gather: $q_{i,j} \equiv$ **do you prefer** x_i **or** x_j

Optimal selection of a sequence of $q_{i,j}$ requires a computationally difficult search, involving a combinatorial optimization.

Lazy Binary Search

input: $x_1, \ldots, x_n \in \mathbb{R}^d$
initialize: $x_1, \ldots, x_n$ in uniformly random order

for k=2,...,n
  for i=1,...,k-1
    if $q_{i,k}$ is ***ambiguous*** given $\{q_{i,j}\}_{i,j<k}$,
      then ask for pairwise comparison,
    else impute $q_{i,j}$ from $\{q_{i,j}\}_{i,j<k}$

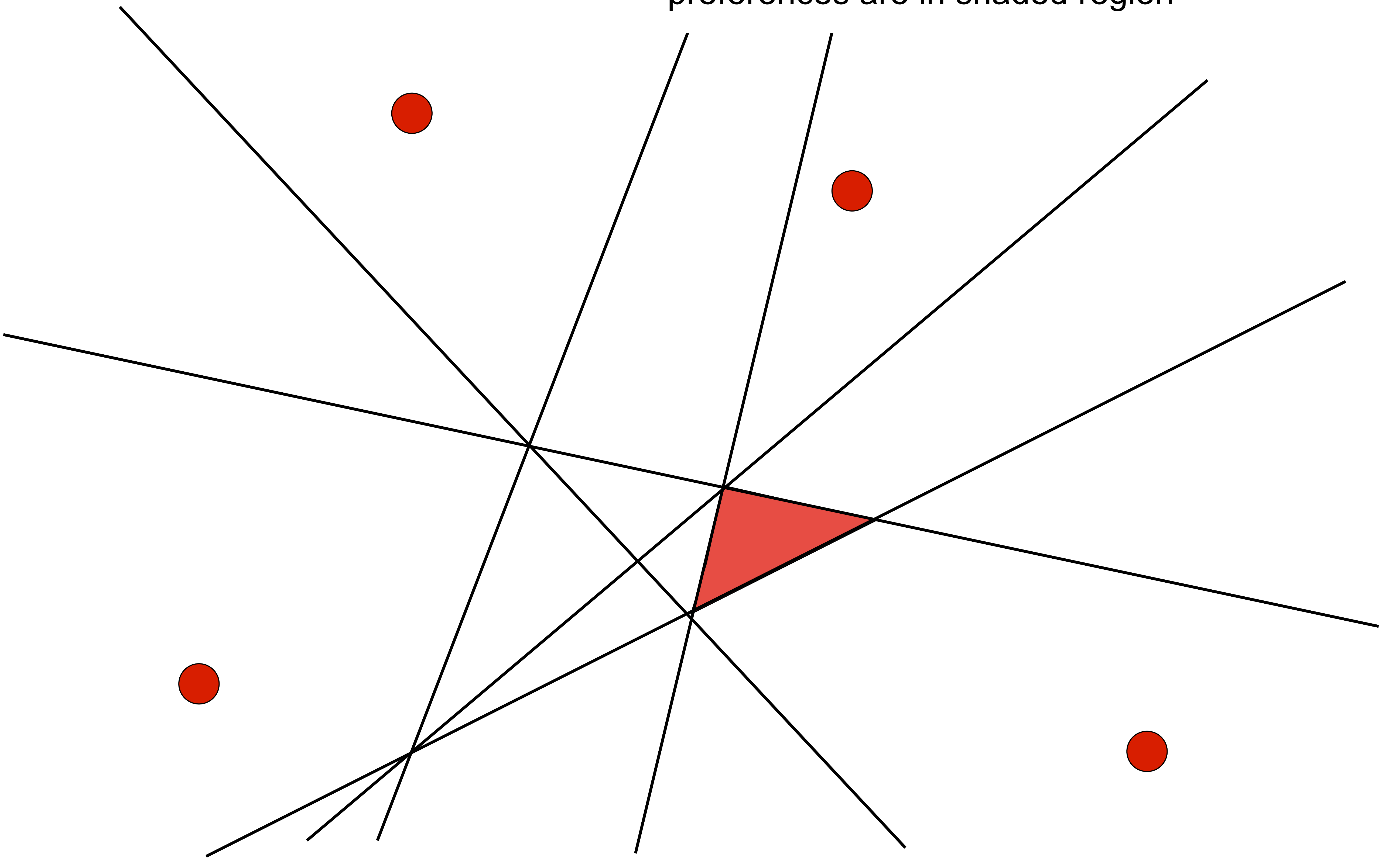
simple linear program

output: ranking of $x_1, \ldots, x_n$ consistent with *all* pairwise comparisons

Ranking and Geometry

suppose we have ranked 4 beers

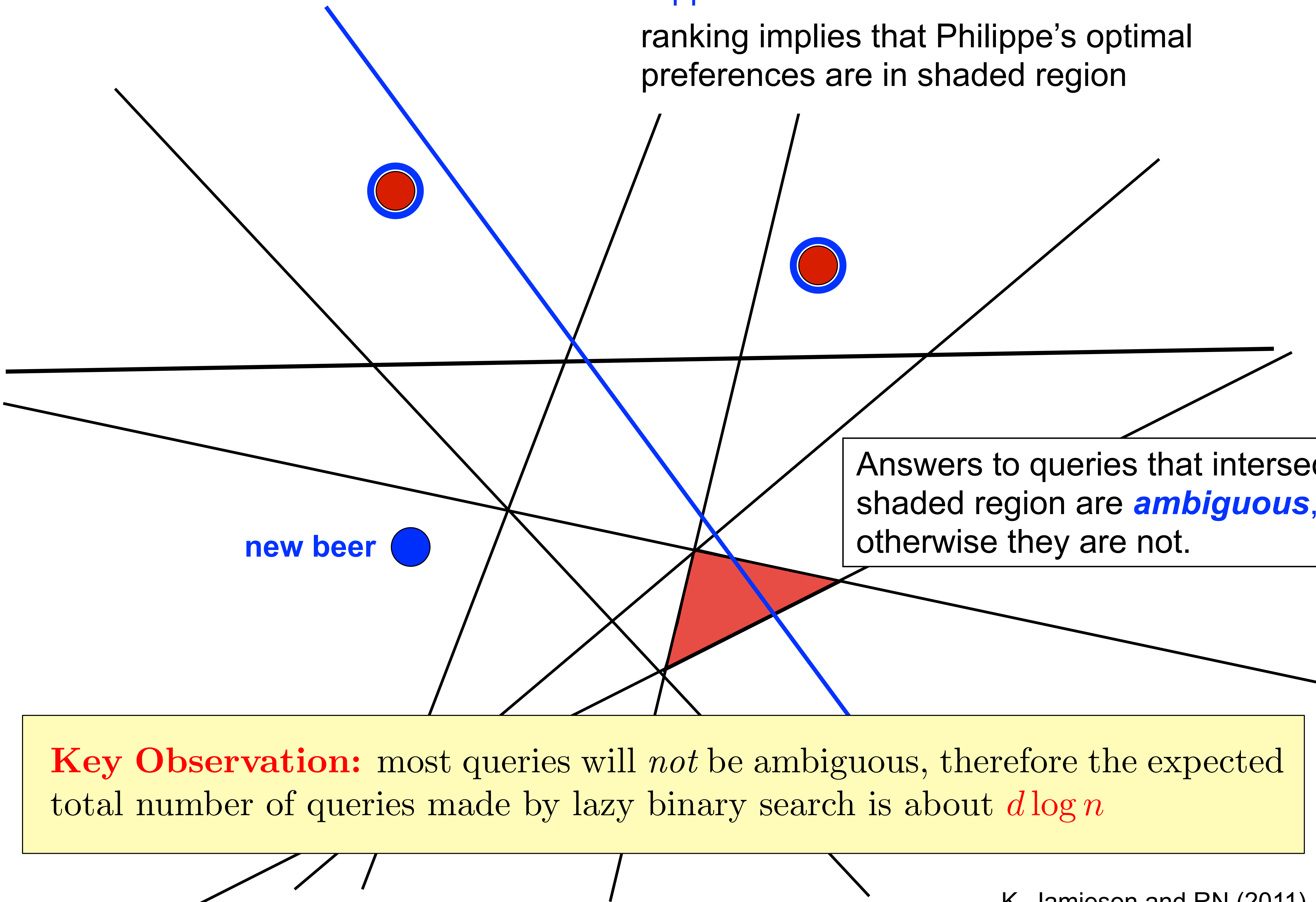
ranking implies that Philippe's optimal preferences are in shaded region



Ranking and Geometry

suppose we have ranked 4 beers

ranking implies that Philippe's optimal preferences are in shaded region



Answers to queries that intersect shaded region are ***ambiguous***, otherwise they are not.

Key Observation: most queries will *not* be ambiguous, therefore the expected total number of queries made by lazy binary search is about $d \log n$

K. Jamieson and RN (2011)

Ranking and Geometry

at k-th step of algorithm

$$\# \text{ of } d\text{-cells} \approx \frac{k^{2d}}{d!} \qquad \text{(Coombs 1960)}$$

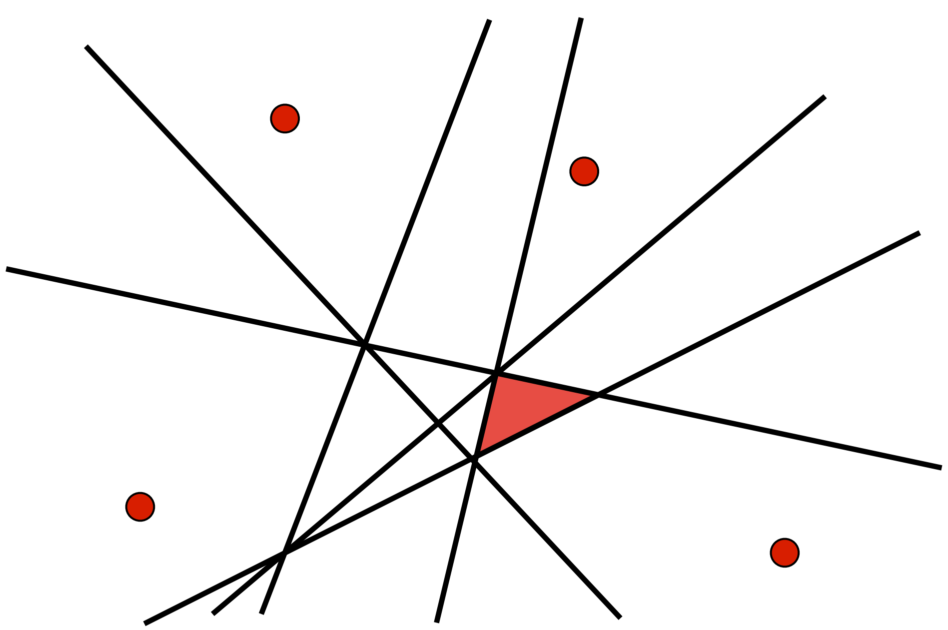
$$\# \text{ intersected} \approx \frac{k^{2(d-1)}}{(d-1)!} \qquad \text{(Buck 1943)}$$

$$\implies \mathbb{P}(\text{ambiguous}) \approx \frac{d}{k^2} \qquad \text{(Cover 1965)}$$

$$\implies \mathbb{E}[\#\text{ambiguous}] \approx \frac{d}{k}$$

$$\implies \mathbb{E}[\# \text{ requested}] \approx \sum_{k=2}^{n} \frac{d}{k} \qquad \text{(Jamieson \& RN 2011)}$$

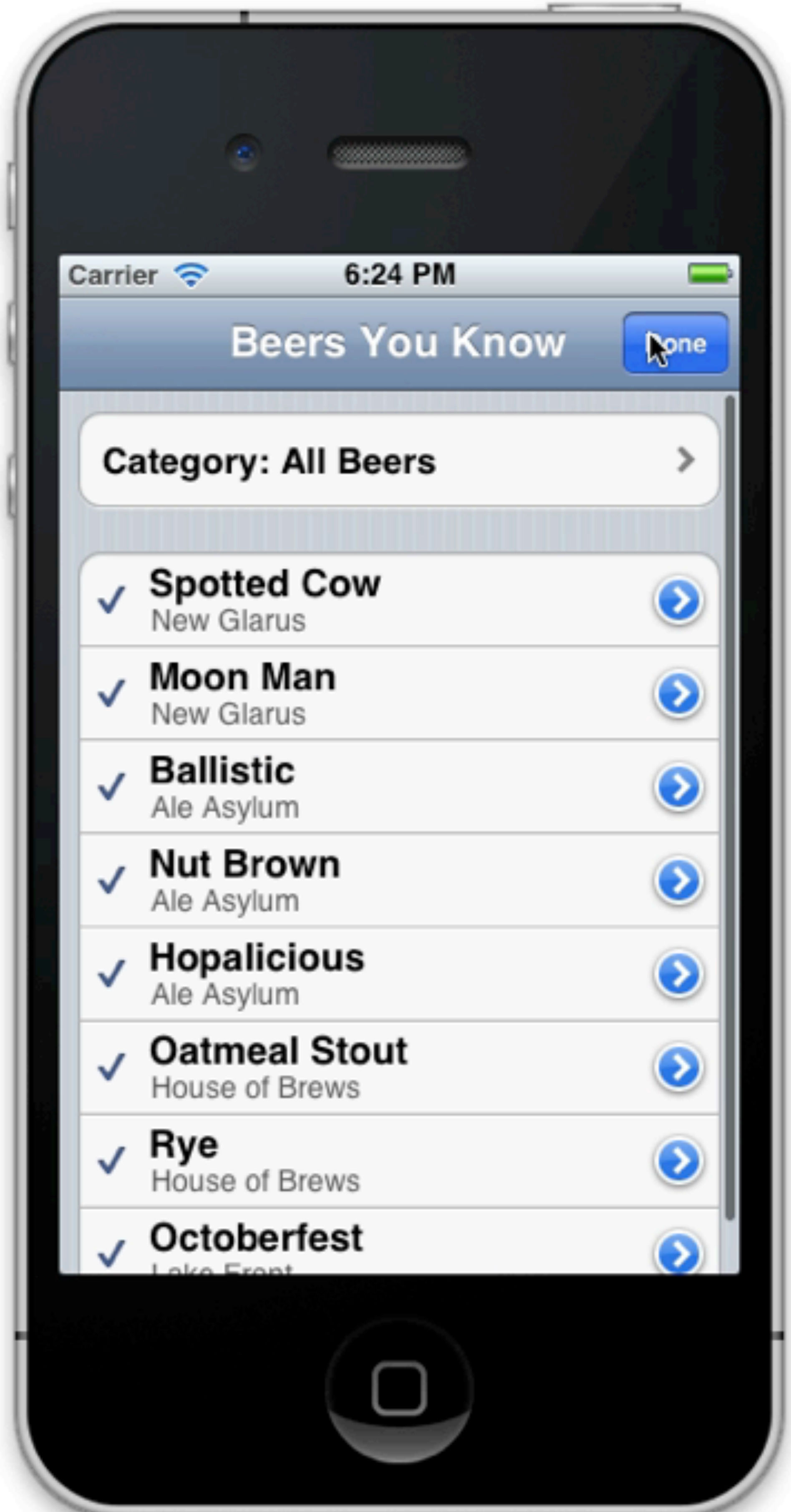
$$\approx d \log n$$



Tolerance to erroneous responses using $d \log^2 n$ queries

robust to noise and non-transitivity

BeerMapper



BeerMapper app learns a persons ranking of beers by selecting pairwise comparisons using lazy binary search and a low-dimensional embedding based on key beer features

BeerMapper - Under the Hood

Algorithm requires feature representations of the beers $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$

ratebeer

<u>Two Hearted Ale - Input ~2500 natural language reviews</u>

http://www.ratebeer.com/beer/two-hearted-ale/1502/2/1/

3.8 AROMA 8/10 APPEARANCE 4/5 TASTE 8/10 PALATE 3/5 OVERALL 15/20

fonefan (25678) - VestJylland, DENMARK - JAN 18, 2009

Bottle 355ml.
Clear light to medium yellow orange color with a average, frothy, good lacing, fully lasting, off-white head. Aroma is moderate to heavy malty, moderate to heavy hoppy, perfume, grapefruit, orange shell, soap. Flavor is moderate to heavy sweet and bitter with a average to long duration. Body is medium, texture is oily, carbonation is soft. [250908]

---

4 AROMA 8/10 APPEARANCE 4/5 TASTE 7/10 PALATE 4/5 OVERALL 17/20

Ungstrup (24358) - Oamaru, NEW ZEALAND - MAR 31, 2005

An orange beer with a huge off-white head. The aroma is sweet and very freshly hoppy with notes of hop oils - very powerful aroma. The flavor is sweet and quite hoppy, that gives flavors of oranges, flowers as well as hints of grapefruit. Very refreshing yet with a powerful body.

Reviews for each beer	Bag of Words weighted by TF*IDF	Get 100 nearest neighbors using cosine distance	Non-metric multidimensional scaling	Embedding in 3 dimensions

BeerMapper - Under the Hood

Algorithm requires feature representations of the beers $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$

Two Hearted Ale - Weighted Bag of Words:

Reviews for each beer

Bag of Words weighted by TF*IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in 3 dimensions

BeerMapper - Under the Hood

Algorithm requires feature representations of the beers $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$

Weighted count vector for the ith beer:

$$z_i \in \mathbb{R}^{400,000}$$

Cosine distance:

$$d(z_i, z_j) = 1 - \frac{z_i^T z_j}{||z_i||\,||z_j||}$$

<u>Two Hearted Ale - Nearest Neighbors:</u>

Bear Republic Racer 5
Avery IPA
Stone India Pale Ale (IPA)
Founders Centennial IPA
Smuttynose IPA
Anderson Valley Hop Ottin IPA
AleSmith IPA
BridgePort IPA
Boulder Beer Mojo IPA
Goose Island India Pale Ale
Great Divide Titan IPA
New Holland Mad Hatter Ale
Lagunitas India Pale Ale
Heavy Seas Loose Cannon Hop3
Sweetwater IPA ...

Reviews for each beer

Bag of Words weighted by TF-IDF

Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in 3 dimensions

BeerMapper - Under the Hood

Algorithm requires feature representations of the beers $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$

Weighted count vector for the ith beer:

$$z_i \in \mathbb{R}^{400{,}000}$$

Cosine distance:

$$d(z_i, z_j) = 1 - \frac{z_i^T z_j}{||z_i||\,||z_j||}$$

<u>Two Hearted Ale - Nearest Neighbors:</u>

Bear Republic Racer 5
Avery IPA
Stone India Pale Ale (IPA)
Founders Centennial IPA
Smuttynose IPA
Anderson Valley Hop Ottin IPA
AleSmith IPA
BridgePort IPA
Boulder Beer Mojo IPA
Goose Island India Pale Ale
Great Divide Titan IPA
New Holland Mad Hatter Ale
Lagunitas India Pale Ale
Heavy Seas Loose Cannon Hop3
Sweetwater IPA ...

Reviews for each beer

Bag of Words weighted by TF*IDF

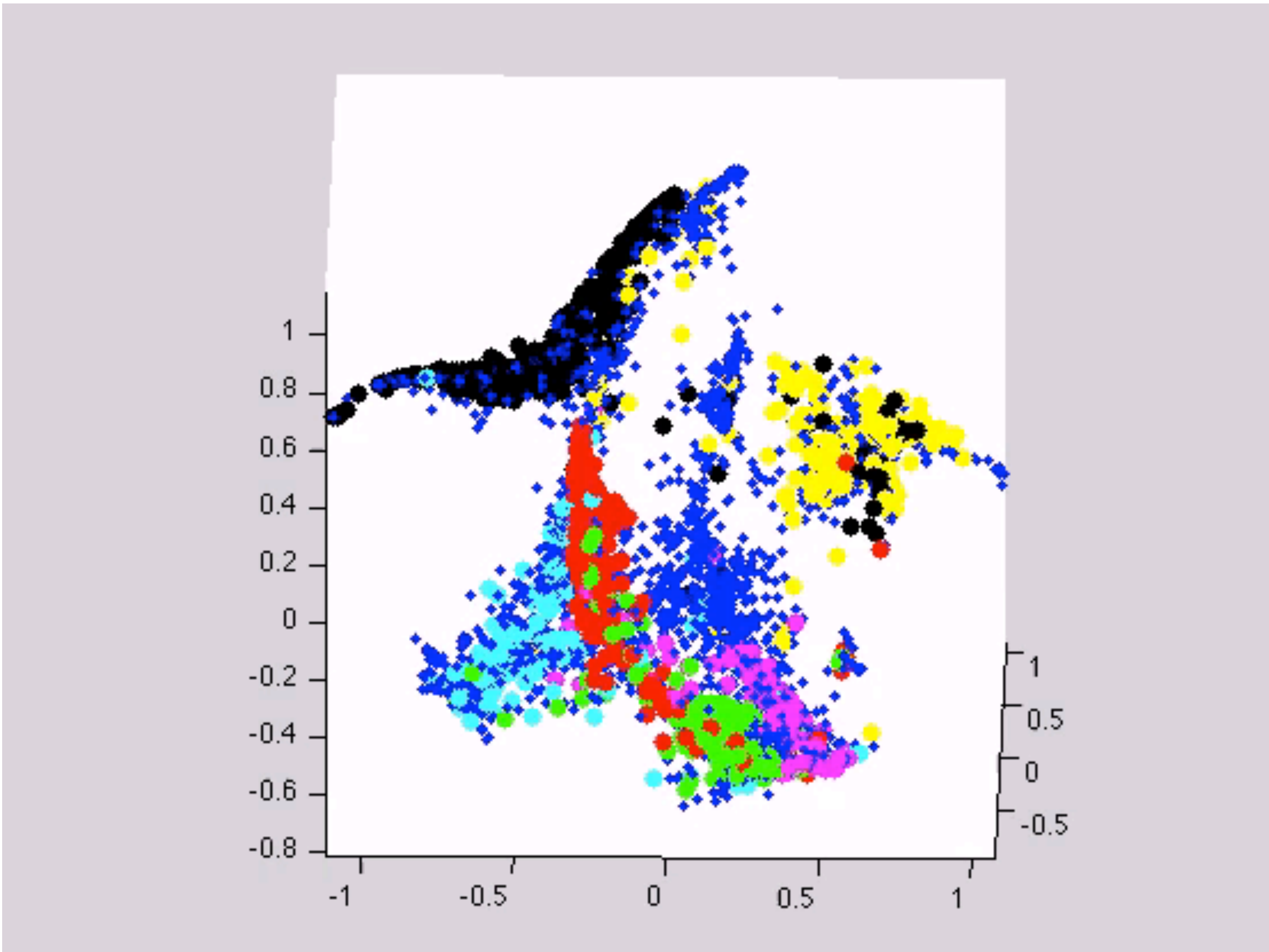
Get 100 nearest neighbors using cosine distance

Non-metric multidimensional scaling

Embedding in 3 dimensions

BeerMapper - Under the Hood

Algorithm requires feature representations of the beers $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$



Sanity check: styles should cluster together and similar styles should be close.

Red = IPA
Green = Pale Ale
Magenta = Amber Ale
Cyan = Lager + Pilsener
Yellow = Belgians (light + dark)
Black = Stout + Porter
Blue = Everything else

Reviews for each beer

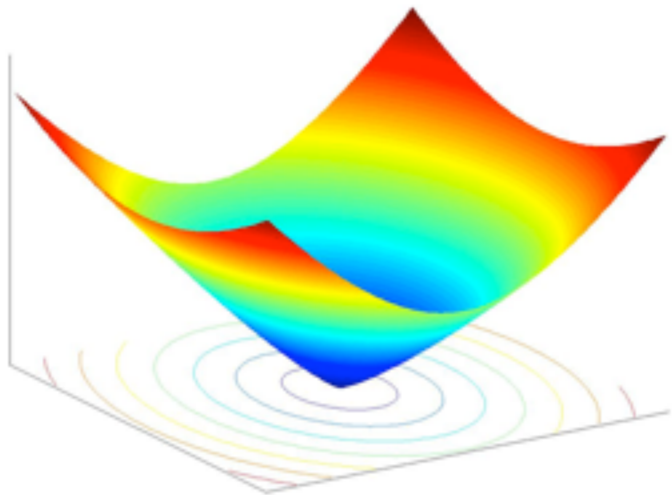
Bag of Words weighted by TF*IDF

Get 15 nearest neighbors using cosine distance

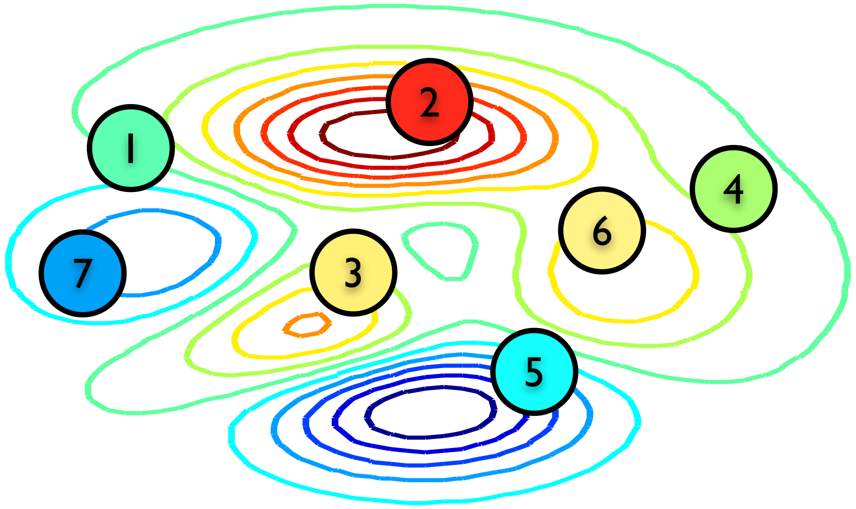
Non-metric multidimensional scaling

Embedding in 3 dimensions

Machine Learning from Comparative Judgements



Derivative Free Optimization
using Human Subjects



Ranking from
Pairwise Comparisons

Challenge:

Computing is cheap, but human assistance/guidance is expensive

Goal:

Optimize such systems with as little human involvement as possible

Humans are much more reliable and consistent at making comparative judgements, than in giving numerical ratings or evaluations

"Binary search" procedures can play a role in *active learning*

References

T. Bijmolt and M. Wedel, “The effects of alternative methods of collecting similarity data for multidimensional scaling,” IJRM 1995

N. Steward, G. Brown and N. Chater, “Absolute identification by relative judgement,” Psych. Review 2005

K. Jamieson, B. Recht, and R. Nowak, “Query complexity of derivative free optimization,” NIPS 2012

O. Shamir, “On the complexity of bandit and derivative free stochastic convex optimization,” arxiv 2012

A. Agrawal, O. Dekel and L. Xiao, “Optimal algorithms for online convex optimization with multi-point bandit feedback,” COLT 2010

A. Nemirovski, A. Juditsky, G. Lan and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” SIAM J. Opt 2009

Y. Yue and T. Joachims, “Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem, 2009

S. Tong and D. Koller, “Support vector machine active learning with applications,” JMLR 2001

M. Horstein, “Sequential decoding using noiseless feedback,” IEEE Trans. IT 1963

M. Burnashev and K. Zigangirov, “An interval estimation problem for controlled observations,” Prob. Info. Transmission 1974

R. Karp and R. Kleinberg, “Noisy binary search and its applications,” SODA 2007

R. Nowak, “The geometry of generalized binary search,” IEEE Trans. IT 2011

R. Castro and R. Nowak, “Minimax bounds for active learning,” IEEE Trans. IT 2008

S. Hanneke, “Rates of convergence in active learning,” Ann. Stat. 2011

M. Raginsky and S. Rahklin, “Lower bounds for passive and active learning,” NIPS 2011

K. Jamieson and R. Nowak, “Active ranking using pairwise comparisons,” NIPS 2011