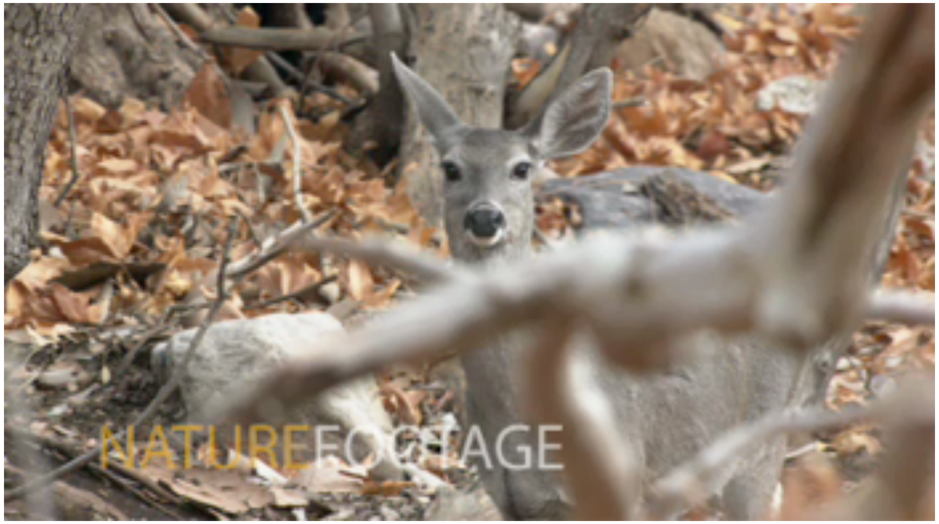
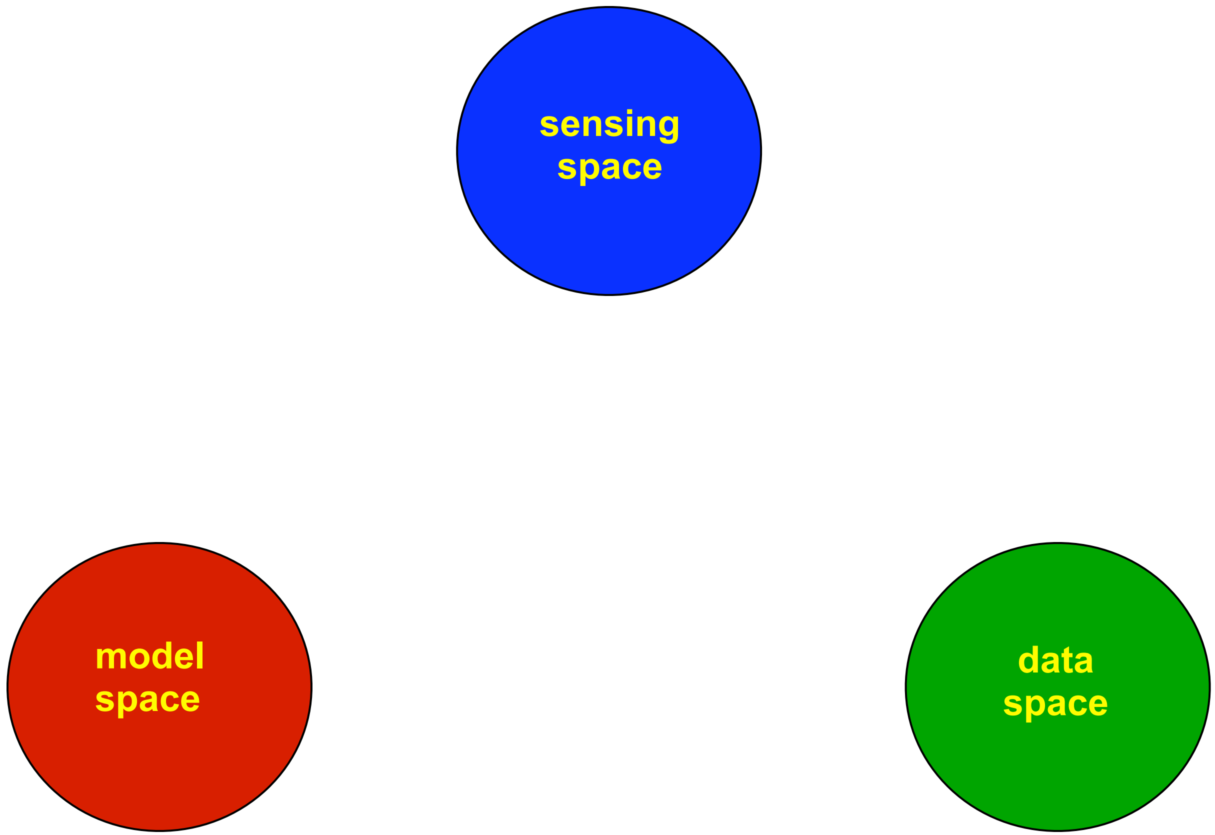
Interactive Information Gathering and Statistical Learning



Feedback from Data Analysis to Data Collection

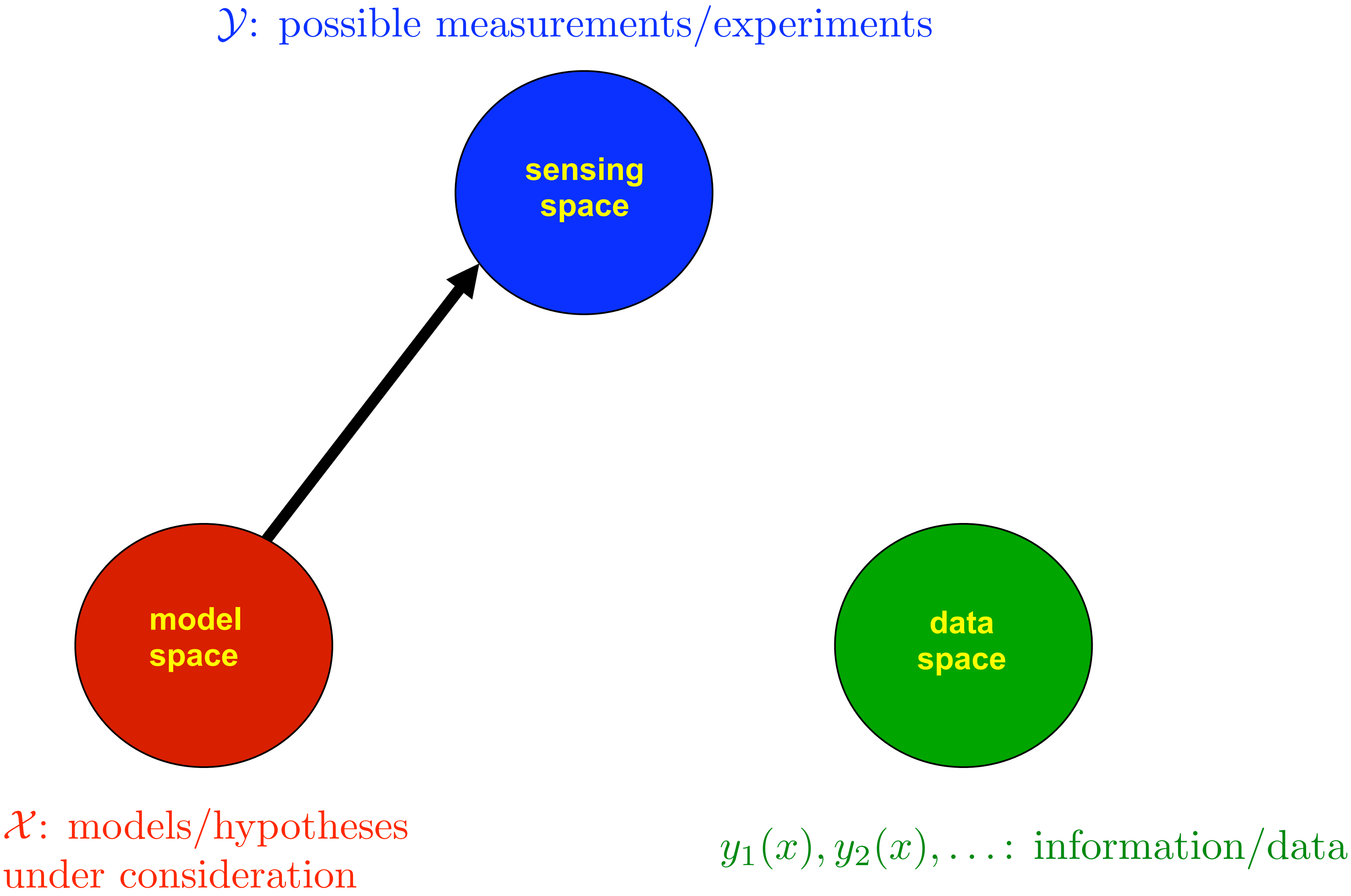
$\mathcal{Y}$: possible measurements/experiments



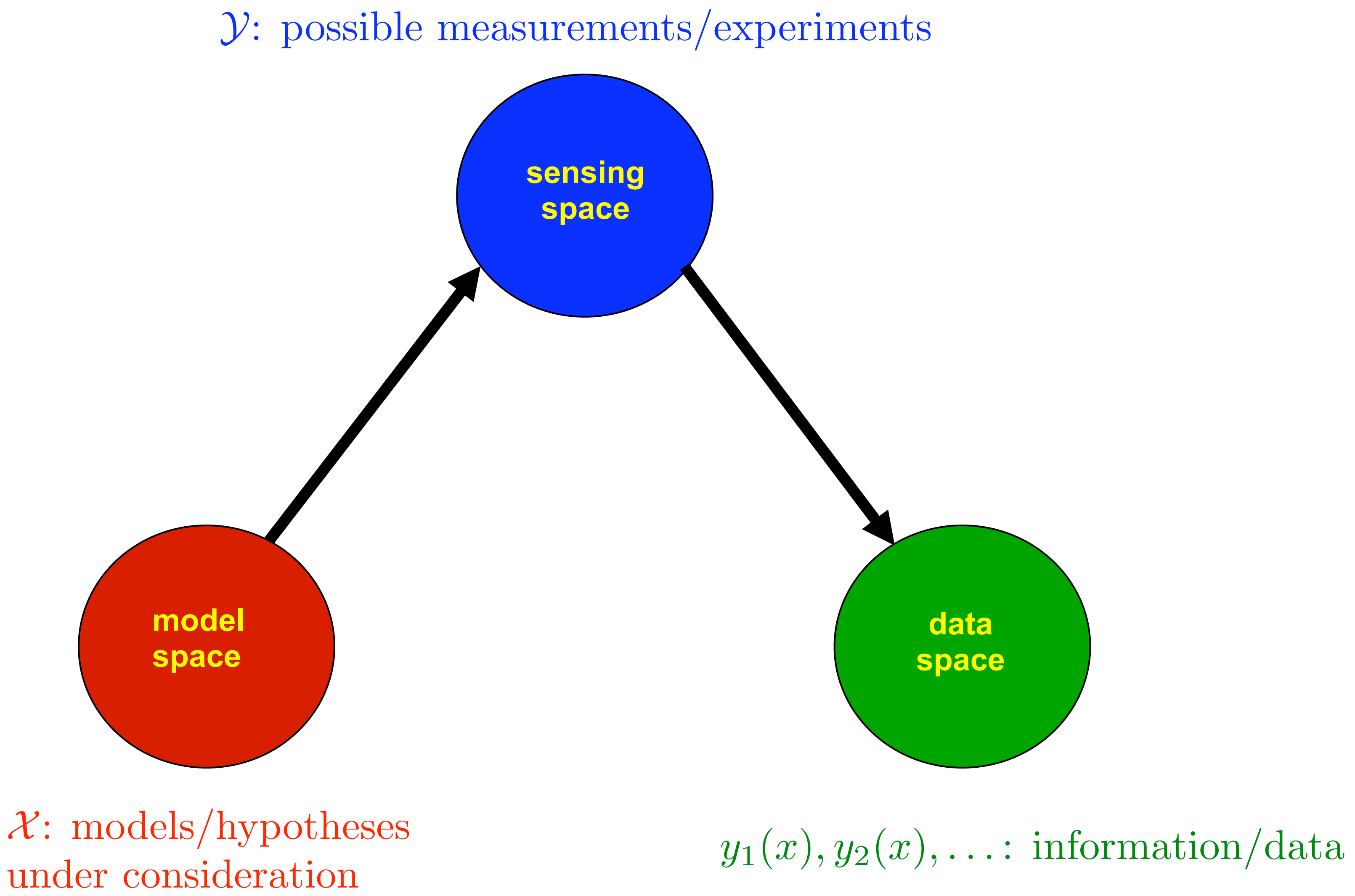
$\mathcal{X}$: models/hypotheses
under consideration

$y_1(x), y_2(x), \ldots$: information/data

Feedback from Data Analysis to Data Collection

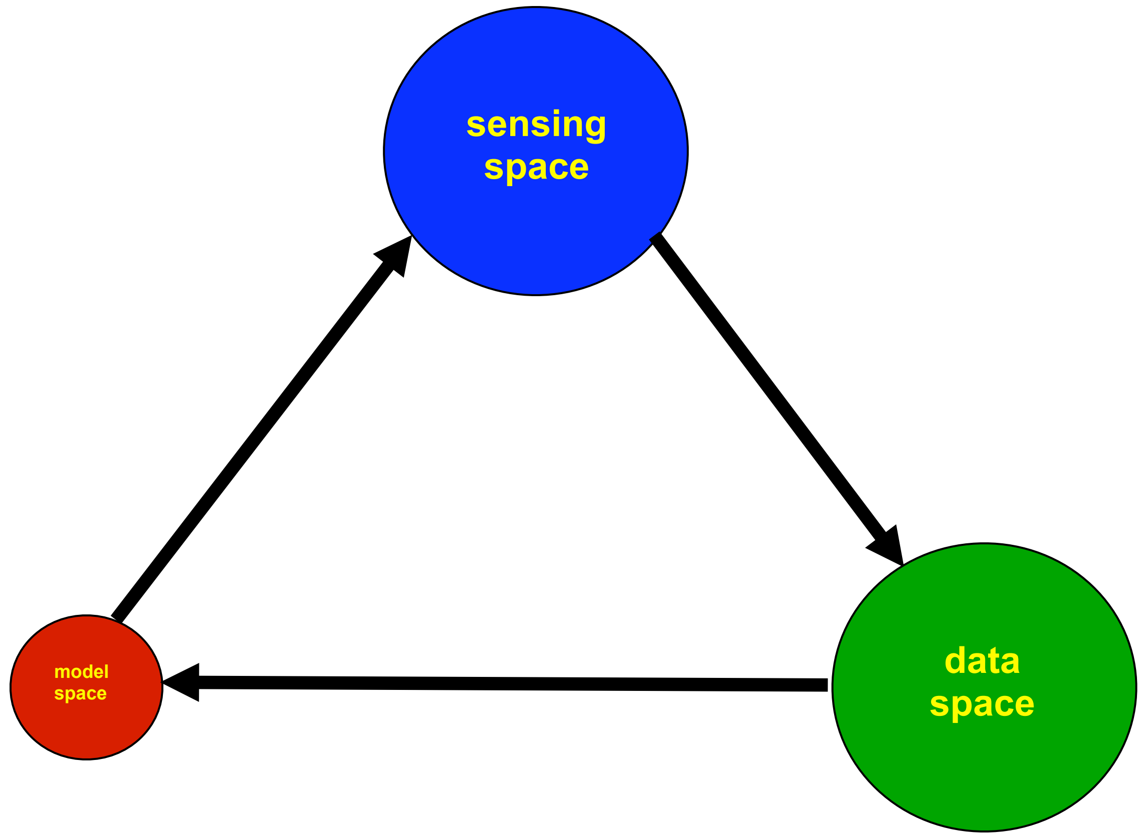


Feedback from Data Analysis to Data Collection



Feedback from Data Analysis to Data Collection

$\mathcal{Y}$: possible measurements/experiments



$\mathcal{X}$: models/hypotheses under consideration

$y_1(x), y_2(x), \ldots$: information/data

Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

Motivation: Inferring Biological Networks

virus

fruit fly

Paul Alhquist
(Molecular Virology)

Motivation: Inferring Biological Networks

virus

13,071 single-gene
knock-down cell strains

fruit fly

Paul Alhquist
(Molecular Virology)

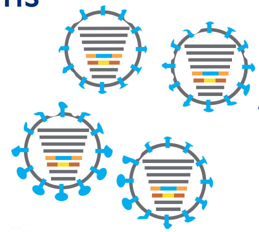
Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

virus

fruit fly

13,071 single-gene
knock-down cell strains



infect each strain with
fluorescing virus

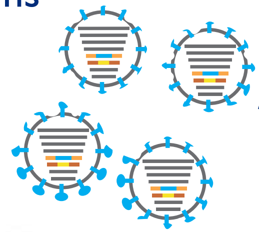
microwell
array

Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

virus

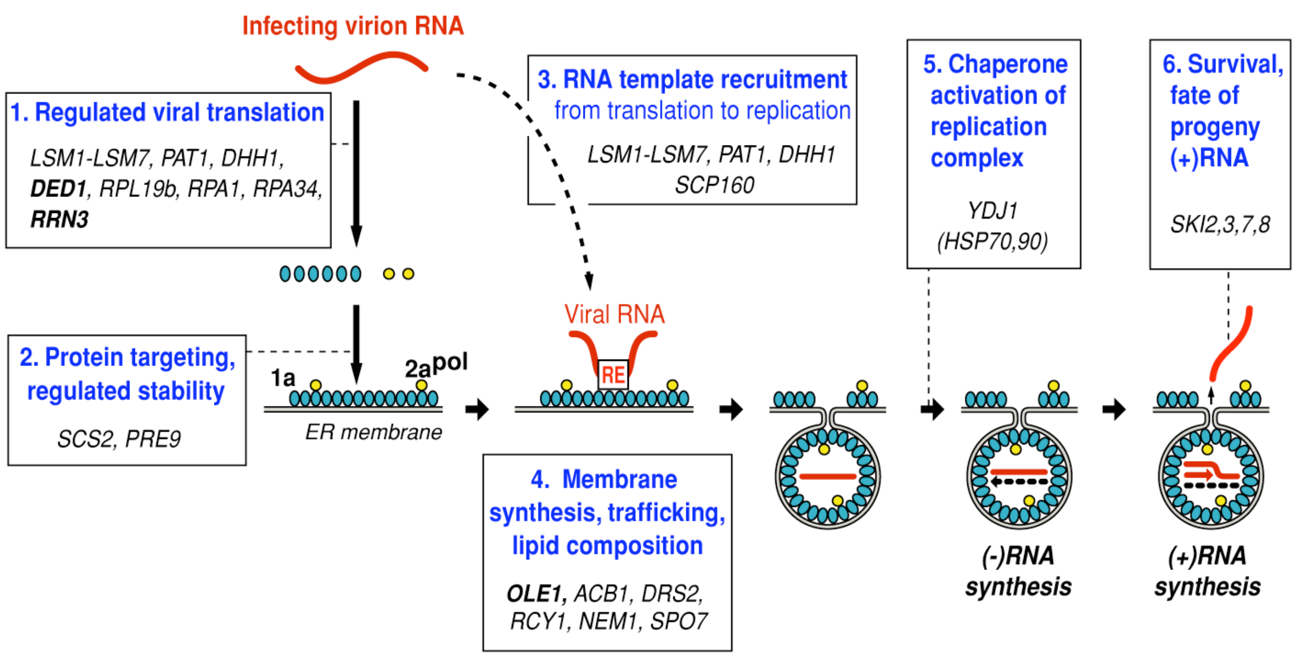
13,071 single-gene
knock-down cell strains



infect each strain with
fluorescing virus

fruit fly

microwell
array

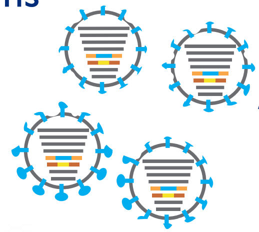


Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

virus

13,071 single-gene
knock-down cell strains



infect each strain with
fluorescing virus

fruit fly

microwell
array

First question: Who are the players in the network?

“Drosophila RNAi screen identifies host genes important for influenza virus replication,” Nature 2008. How do they confidently determine the ~100 out of 13K genes hijacked for virus replication from extremely noisy data?

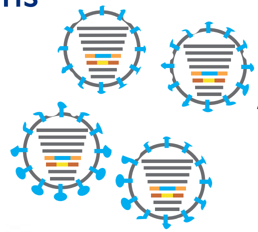
Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

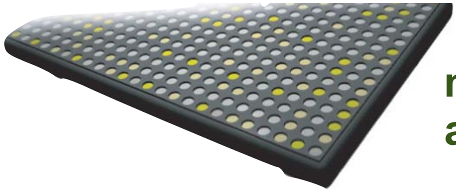
virus

fruit fly

13,071 single-gene knock-down cell strains



infect each strain with fluorescing virus



microwell array

First question: Who are the players in the network?

"Drosophila RNAi screen identifies host genes important for influenza virus replication," Nature 2008. How do they confidently determine the ~100 out of 13K genes hijacked for virus replication from extremely noisy data?

Sequential Experimental Design:

Stage 1: assay all 13K strains, twice; keep all with significant fluorescence in one or both assays for 2nd stage (13K $\rightarrow$ 1K)

Stage 2: assay remaining 1K strains, 6-12 times; retain only those with statistically significant fluorescence (1K $\rightarrow$ 100)

Motivation: Inferring Biological Networks

Paul Alhquist
(Molecular Virology)

virus

fruit fly

13,071 single-gene
knock-down cell strains

infect each strain with
fluorescing virus

microwell
array

First question: Who are the players in the network?

"Drosophila RNAi screen identifies host genes important for influenza virus replication," Nature 2008. How do they confidently determine the ~100 out of 13K genes hijacked for virus replication from extremely noisy data?

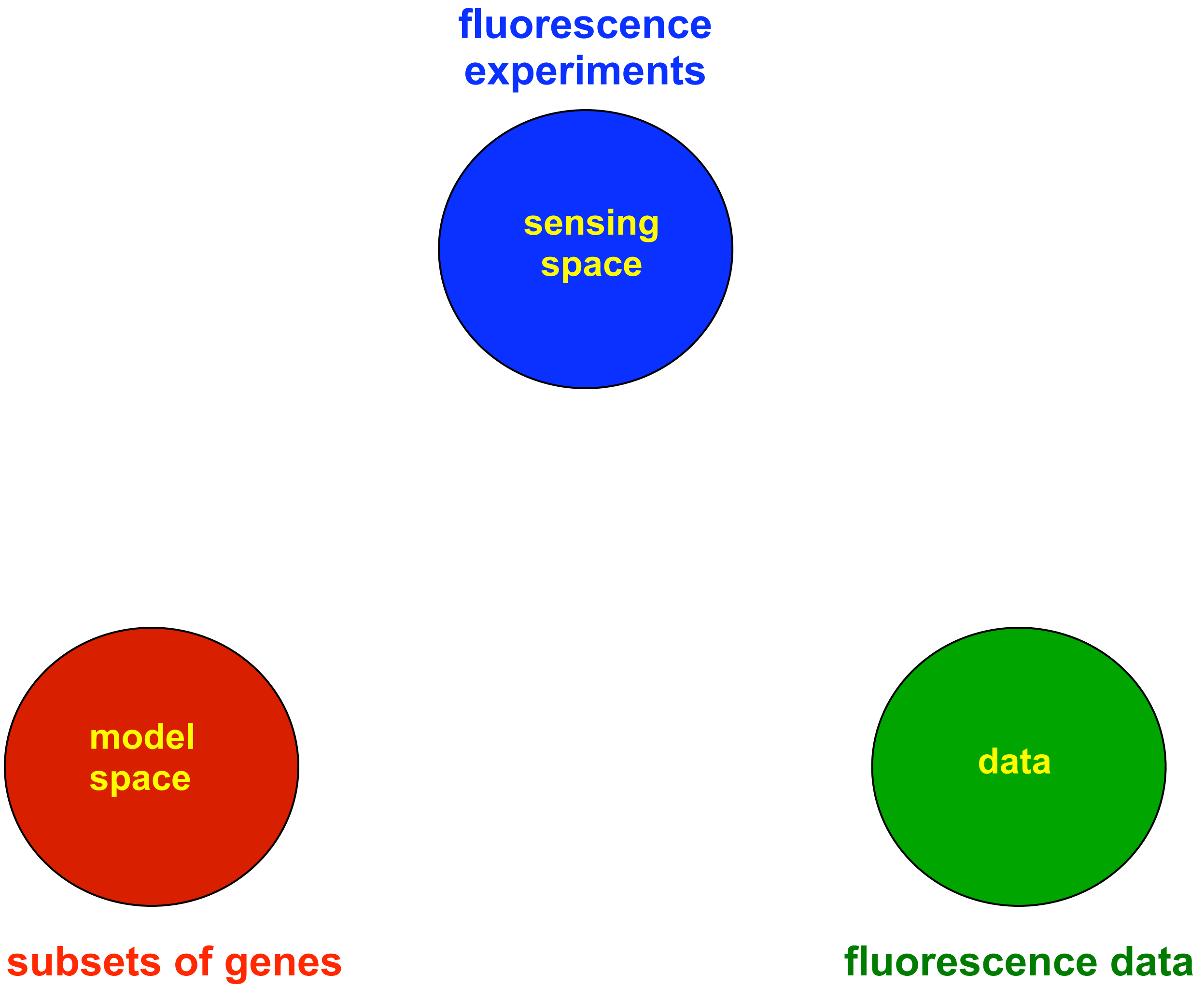
Sequential Experimental Design:

Stage 1: assay all 13K strains, twice; keep all with significant fluorescence in one or both assays for 2nd stage (13K $\rightarrow$ 1K)

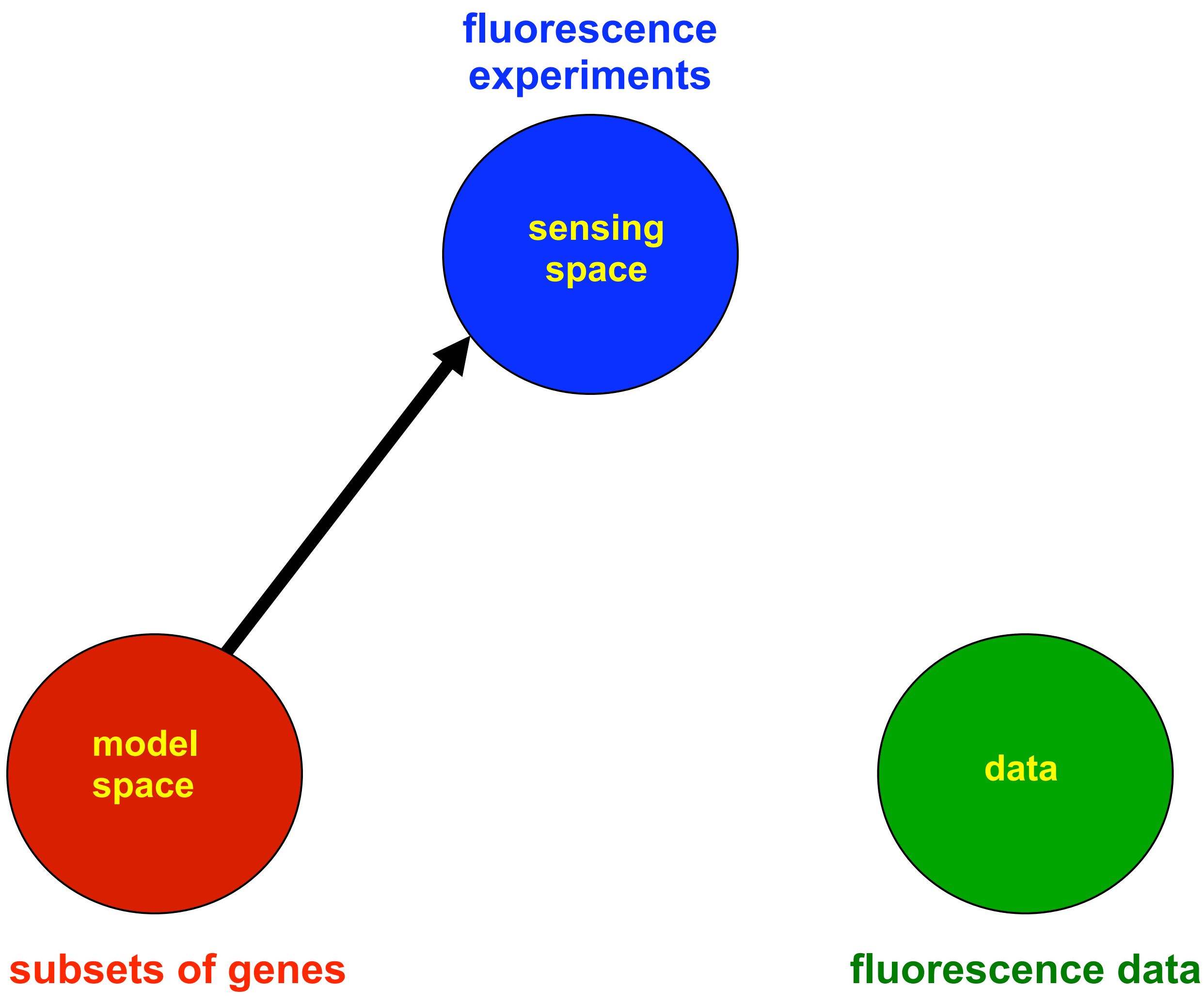
Stage 2: assay remaining 1K strains, 6-12 times; retain only those with statistically significant fluorescence (1K $\rightarrow$ 100)

vastly more efficient that replicating all 13K experiments many times

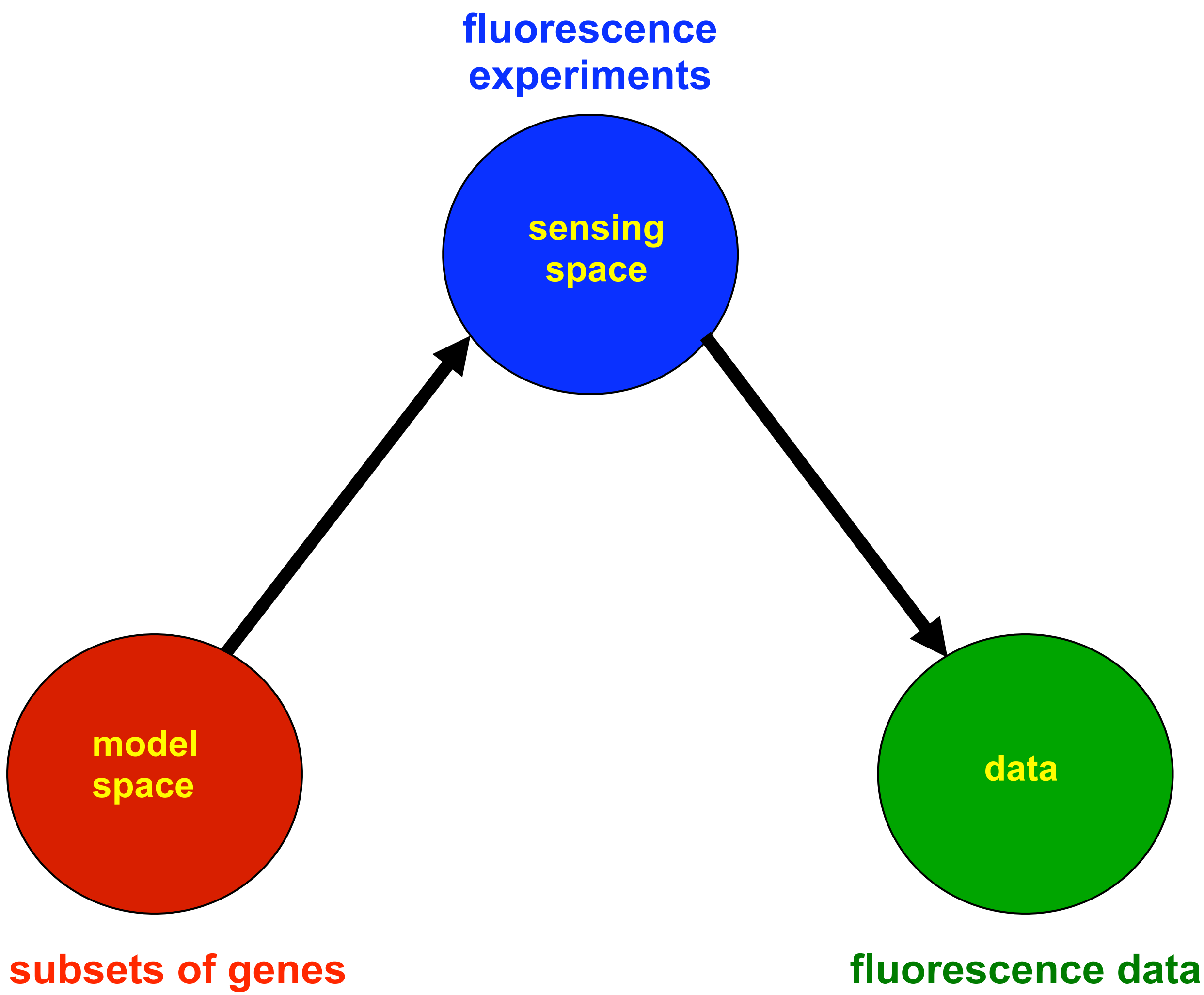
Feedback from Data Analysis to Data Collection



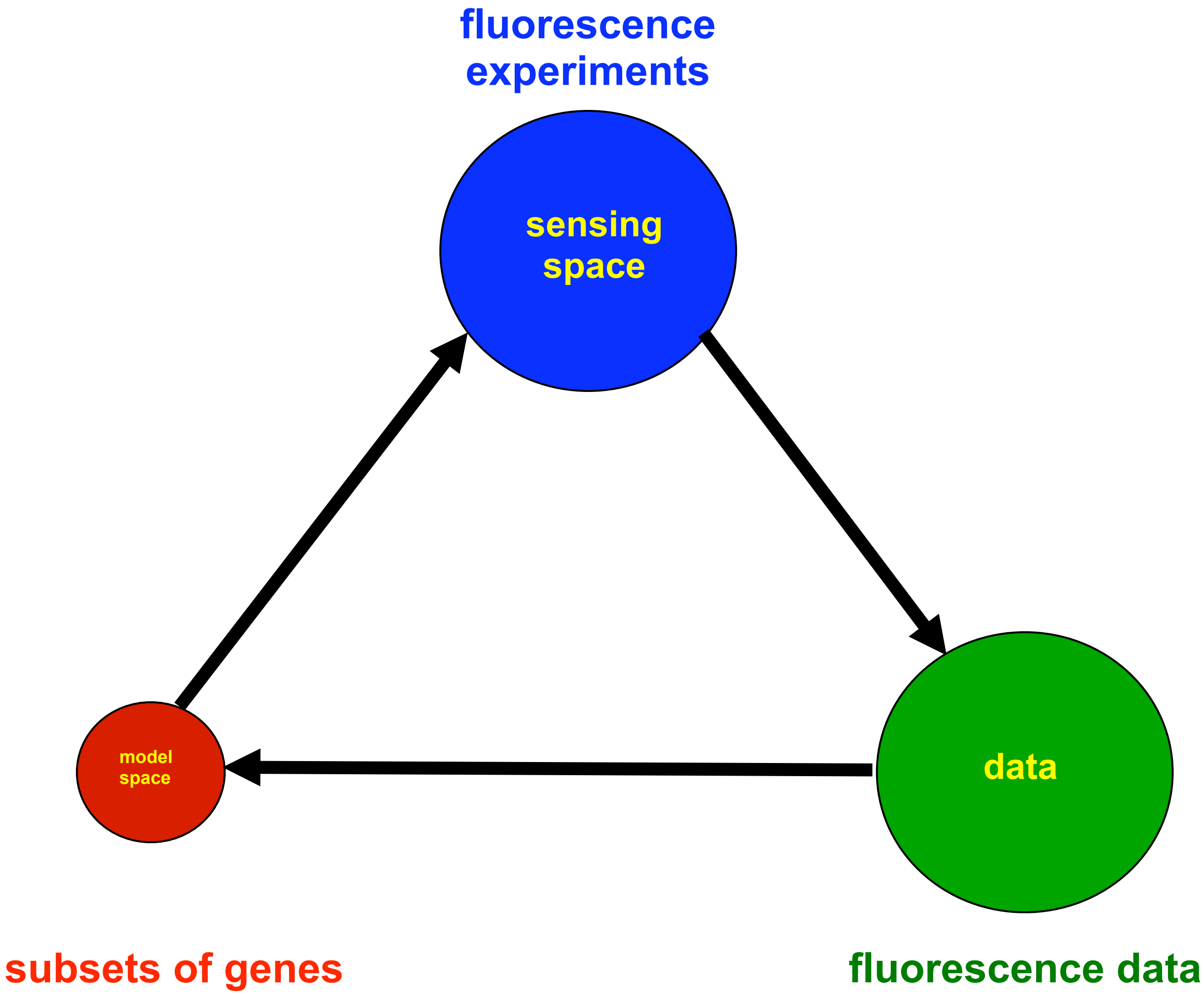
Feedback from Data Analysis to Data Collection



Feedback from Data Analysis to Data Collection



Feedback from Data Analysis to Data Collection



Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Goal: Infer the correct model $x \in \mathcal{X}$ from measurements $y(x)$, $y \in \mathcal{Y}$

Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Goal: Infer the correct model $x \in \mathcal{X}$ from measurements $y(x)$, $y \in \mathcal{Y}$

Information: samples of the form $y_1(x), \ldots, y_n(x)$

Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Goal: Infer the correct model $x \in \mathcal{X}$ from measurements $y(x)$, $y \in \mathcal{Y}$

Information: samples of the form $y_1(x), \ldots, y_n(x)$

Non-Adaptive Information: $y_1, y_2, \cdots \in \mathcal{Y}$ non-adaptively chosen (deterministically or randomly) independent of x

Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Goal: Infer the correct model $x \in \mathcal{X}$ from measurements $y(x)$, $y \in \mathcal{Y}$

Information: samples of the form $y_1(x), \ldots, y_n(x)$

Non-Adaptive Information: $y_1, y_2, \cdots \in \mathcal{Y}$ non-adaptively chosen (deterministically or randomly) independent of x

Adaptive Information: $y_1, y_2, \cdots \in \mathcal{Y}$ are selected sequentially and y_i can depend on previously gathered information, i.e., $y_1(x), \ldots, y_{i-1}(x)$

Adaptive Information

Model Space: $\mathcal{X}$ is a collection of models

Measurement Space: $\mathcal{Y}$ is a set of sensing or experimental actions

Goal: Infer the correct model $x \in \mathcal{X}$ from measurements $y(x)$, $y \in \mathcal{Y}$

Information: samples of the form $y_1(x), \ldots, y_n(x)$

Non-Adaptive Information: $y_1, y_2, \cdots \in \mathcal{Y}$ non-adaptively chosen (deterministically or randomly) independent of x

Adaptive Information: $y_1, y_2, \cdots \in \mathcal{Y}$ are selected sequentially and y_i can depend on previously gathered information, i.e., $y_1(x), \ldots, y_{i-1}(x)$

Does adaptivity help?

see "Information-Based Complexity" literature; e.g.,

E. Novak. *On the power of adaption*. J. Complexity 12 (1996), 199-237.

Adaptive vs. Non-Adaptive: Three Situations

The "bare minimum" number of measurements depends on intrinsic complexity of $\mathcal{X}$ (e.g, metric entropy).

In practice, the minimum number depends on jointly on $\mathcal{X}$ and $\mathcal{Y}$.

Adaptive vs. Non-Adaptive: Three Situations

The "bare minimum" number of measurements depends on intrinsic complexity of $\mathcal{X}$ (e.g, metric entropy).

In practice, the minimum number depends on jointly on $\mathcal{X}$ and $\mathcal{Y}$.

Equal and Good:
adaptive and non-adaptive equally informative and require about the bare minimum of measurements

Adaptive vs. Non-Adaptive: Three Situations

The "bare minimum" number of measurements depends on intrinsic complexity of $\mathcal{X}$ (e.g, metric entropy).

In practice, the minimum number depends on jointly on $\mathcal{X}$ and $\mathcal{Y}$.

Equal and Good:
adaptive and non-adaptive equally informative and require about the bare minimum of measurements

Equal and Bad:
adaptive and non-adaptive equally (non)-informative and require many more measurements then the bare minimum

Adaptive vs. Non-Adaptive: Three Situations

The "bare minimum" number of measurements depends on intrinsic complexity of $\mathcal{X}$ (e.g, metric entropy).

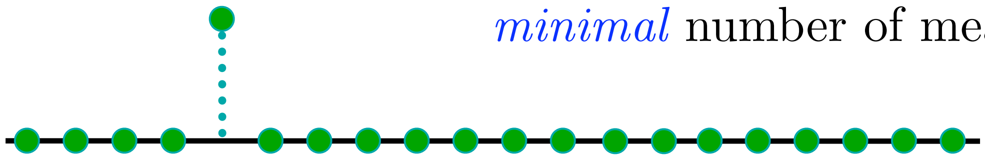
In practice, the minimum number depends on jointly on $\mathcal{X}$ and $\mathcal{Y}$.

Equal and Good: adaptive and non-adaptive equally informative and require about the bare minimum of measurements

Equal and Bad: adaptive and non-adaptive equally (non)-informative and require many more measurements then the bare minimum

Good and Bad: adaptive requires bare minimum number of measurements, non-adaptive requires many more

Does Adaptivity Help ?



identify a sparse signal $x \in \mathbb{R}^n$ from a *minimal* number of measurements

Point measurements: $y = \langle x\, ,\ \delta_k \rangle = x_k$

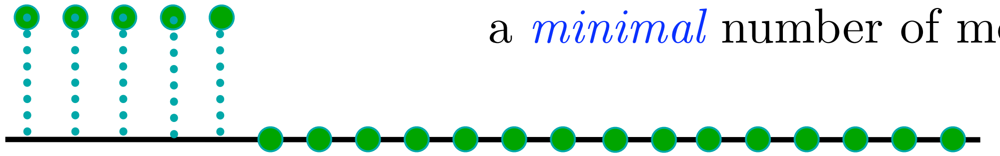
$O(n)$ measurements (random or adaptive) are needed to recover x

Compressed Sensing: $y \ = \ \langle x\, , \phi \rangle$ where $\phi \in \{-1, 1\}^n$

$O(\log n)$ measurements (random or adaptive) are needed to recover x

Adaptivity doesn't help

Does Adaptivity Help ?



identify a threshold signal $x \in \mathbb{R}^n$ from a *minimal* number of measurements

Point measurements: $y = \langle x \, , \; \delta_k \rangle = x_k$

$O(n)$ random measurements are needed to recover x

$O(\log n)$ adaptive measurements are needed to recover x (binary search)

Compressed Sensing: $y \; = \; \langle x \, , \, \phi \rangle$ where $\phi \in \{-1, 1\}^n$

$O(\log n)$ random measurements are needed to recover x

Adaptivity may help, depending on structure of signal and measurements

Noisy Compressed Sensing

$$y = Ax + w \qquad x \text{ is } k\text{-sparse}$$

experimental design: how to design A ?

Noisy Compressed Sensing

$$y \;=\; Ax \;+\; w \qquad x \text{ is } k\text{-sparse}$$

experimental design: how to design A ?

Constraints:

- sample budget: A is $m \times n$ with $k < m < n$
- precision budget: $\|A\|_F^2 \leq \text{Constant}$

Noisy Compressed Sensing

$$y = Ax + w \qquad x \text{ is } k\text{-sparse}$$

experimental design: how to design A ?

Constraints:

- sample budget: A is $m \times n$ with $k < m < n$
- precision budget: $\|A\|_F^2 \le$ Constant

Sequential Design: how to chose $A_1, \ldots, A_k$ to minimize MSE of recovery?

$$\begin{aligned} y_1 &= A_1 x + w_1 \\ y_2 &= A_2 x + w_2 \\ &\vdots \\ y_k &= A_k x + w_k \end{aligned}$$

Noisy Compressed Sensing

$$y \;=\; Ax \;+\; w \qquad x \text{ is } k\text{-sparse}$$

experimental design: how to design A ?

Constraints:

- sample budget: A is $m \times n$ with $k < m < n$
- precision budget: $\|A\|_F^2 \leq$ Constant

Sequential Design: how to chose $A_1, \ldots, A_k$ to minimize MSE of recovery?

$$\begin{aligned} y_1 &= A_1 x + w_1 \\ y_2 &= A_2 x + w_2 \\ &\vdots \\ y_k &= A_k x + w_k \end{aligned}$$

$$\begin{aligned} \text{Non-Adaptive:} \quad \text{MSE} &\leq C \log(n) \frac{k}{m} \\ \text{Adaptive/Sequential:} \quad \text{MSE} &\leq C' \frac{k}{m} \end{aligned}$$

Haupt, Baraniuk, Castro, RN '09

Noisy Compressed Sensing

$$y = Ax + w \qquad x \text{ is } k\text{-sparse}$$

experimental design: how to design A ?

Constraints:

- sample budget: A is $m \times n$ with $k < m < n$
- precision budget: $\|A\|_F^2 \leq$ Constant

Sequential Design: how to chose $A_1, \ldots, A_k$ to minimize MSE of recovery?

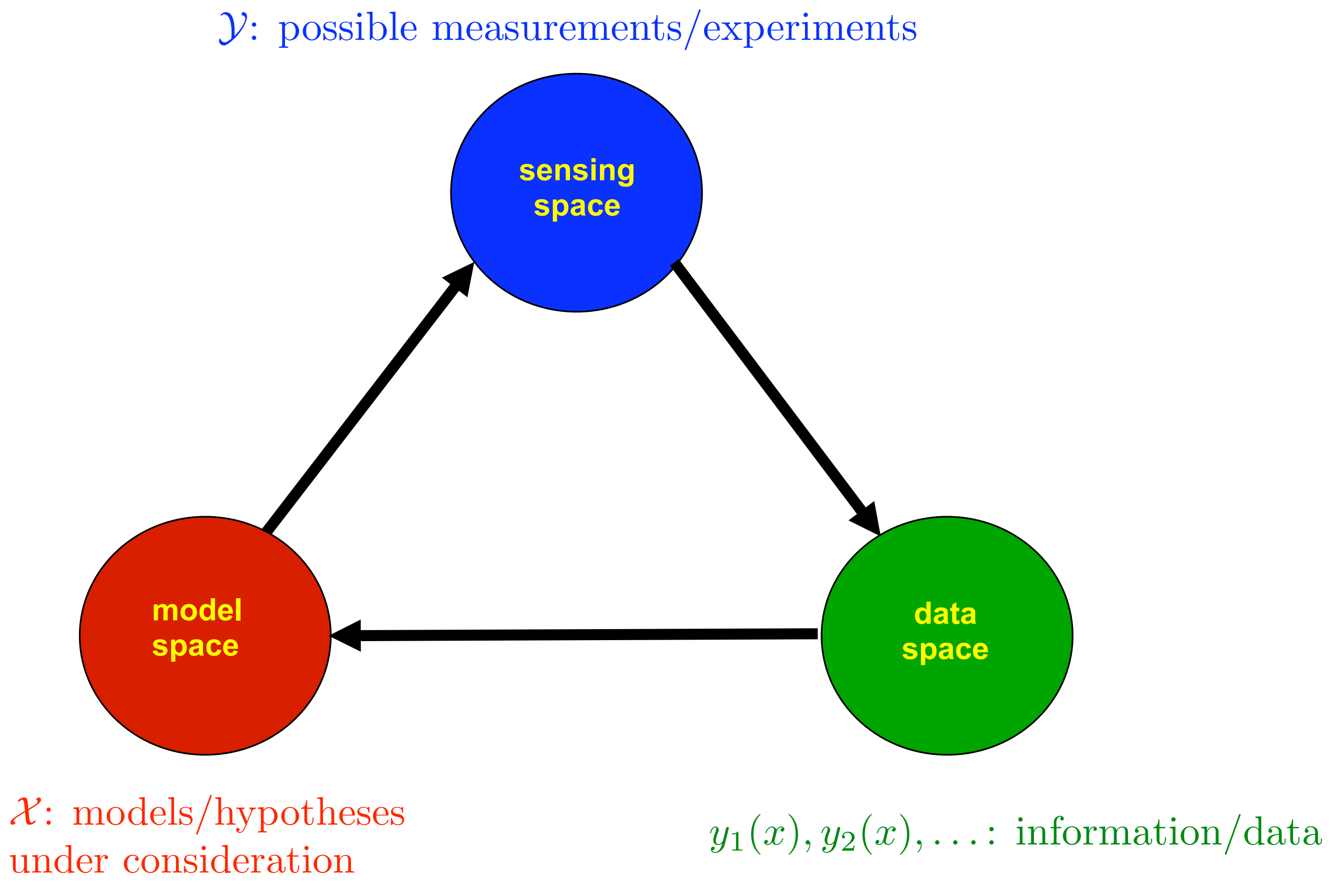
$$\begin{aligned} y_1 &= A_1 x + w_1 \\ y_2 &= A_2 x + w_2 \\ &\vdots \\ y_k &= A_k x + w_k \end{aligned}$$

Good and Better: adaptive and non-adaptive require bare minimum number of measurements, but adaptive measurements improve MSE

$$\begin{aligned} \text{Non-Adaptive:} \quad \text{MSE} &\leq C \log(n) \frac{k}{m} \\ \text{Adaptive/Sequential:} \quad \text{MSE} &\leq C' \frac{k}{m} \end{aligned}$$

Haupt, Baraniuk, Castro, RN ’09

The General Problem



1. Adaptive information can improve MSE/SNR performance (Matt Malloy's talk)

2. Adaptive information can reduce the number of measurements needed (especially when the nature of the measurements is restricted in some way)

Optimization: Incremental Information Gain Algorithm

Optimal sequential designs are intractable in most situations, so usually approximate methods are used.

Optimization: Incremental Information Gain Algorithm

Optimal sequential designs are intractable in most situations, so usually approximate methods are used.

Given a distribution $p(x)$, the information gained by observing $z = y(x)$ is quantified by the reduction in Shannon entropy

$$U(y, z) := \int p(x|y, z) \log p(x|y, z)\, dx - \int p(x) \log p(x)\, dx .$$

Optimization: Incremental Information Gain Algorithm

Optimal sequential designs are intractable in most situations, so usually approximate methods are used.

Given a distribution $p(x)$, the information gained by observing $z = y(x)$ is quantified by the reduction in Shannon entropy

$$U(y,z) \;:=\; \int p(x|y,z)\log p(x|y,z)\,dx \;-\; \int p(x)\log p(x)\,dx \;.$$

A priori, z is a random variable with distribution $p(z|y) = \int p(z|x,y)p(x)\,dx$. The *information-gain* is defined as the expected value

$$U(y) := \int U(y,z)p(z|y)\,dz.$$

"Information-Gain"
(Shannon '48, Lindley '56)

Optimization: Incremental Information Gain Algorithm

Optimal sequential designs are intractable in most situations, so usually approximate methods are used.

Given a distribution $p(x)$, the information gained by observing $z = y(x)$ is quantified by the reduction in Shannon entropy

$$U(y,z) \;:=\; \int p(x|y,z)\log p(x|y,z)\,dx \;-\; \int p(x)\log p(x)\,dx\;.$$

A priori, z is a random variable with distribution $p(z|y) = \int p(z|x,y)p(x)\,dx$. The *information-gain* is defined as the expected value

$$U(y) := \int U(y,z)p(z|y)\,dz.$$

"Information-Gain"
(Shannon '48, Lindley '56)

Incremental Information-Gain Algorithm

initialize: $p_0 =$ uniform over $\mathcal{X}$
for $n = 0, 1, 2, \ldots$
1) Compute information gain U_n based on p_n
2) Select $y_n = \arg\max_{y\in\mathcal{Y}} U_n(y)$
3) Obtain $z_n = y_n(x^*)$
4) Update posterior distribution $p_n \to p_{n+1}$
$\widehat{x}_n = \arg\max_h p_n(x)$

Optimization: Incremental Information Gain Algorithm

Optimal sequential designs are intractable in most situations, so usually approximate methods are used.

Given a distribution $p(x)$, the information gained by observing $z = y(x)$ is quantified by the reduction in Shannon entropy

$$U(y,z) \;:=\; \int p(x|y,z)\log p(x|y,z)\,dx \;-\; \int p(x)\log p(x)\,dx\;.$$

A priori, z is a random variable with distribution $p(z|y) = \int p(z|x,y)p(x)\,dx$. The *information-gain* is defined as the expected value

$$U(y) := \int U(y,z)p(z|y)\,dz.$$

"Information-Gain"
(Shannon '48, Lindley '56)

Incremental Information-Gain Algorithm

initialize: $p_0 =$ uniform over $\mathcal{X}$
for $n = 0, 1, 2, \ldots$

1) Compute information gain U_n based on p_n
2) Select $y_n = \arg\max_{y\in\mathcal{Y}} U_n(y)$
3) Obtain $z_n = y_n(x^*)$
4) Update posterior distribution $p_n \to p_{n+1}$

$$\widehat{x}_n = \arg\max_h p_n(x)$$

long history, special cases known to yield near-optimal designs (see classic papers by Lindley, Degroot)

a very nice recent paper that unifies many ideas:

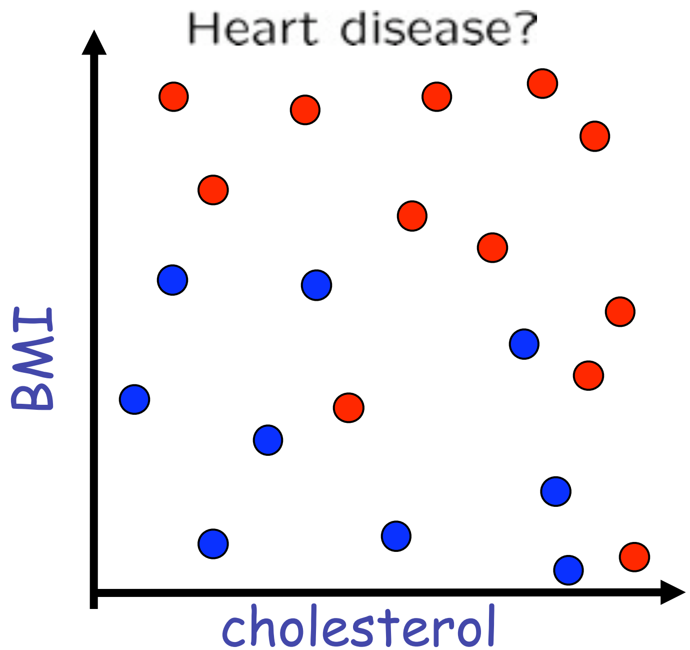
Golovin and Krause. *Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization*, 2010

Active Learning

Learning Problem: Consider a binary prediction problem involving a collection of “classifiers.” Each classifier maps points in the ”feature-space” (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.

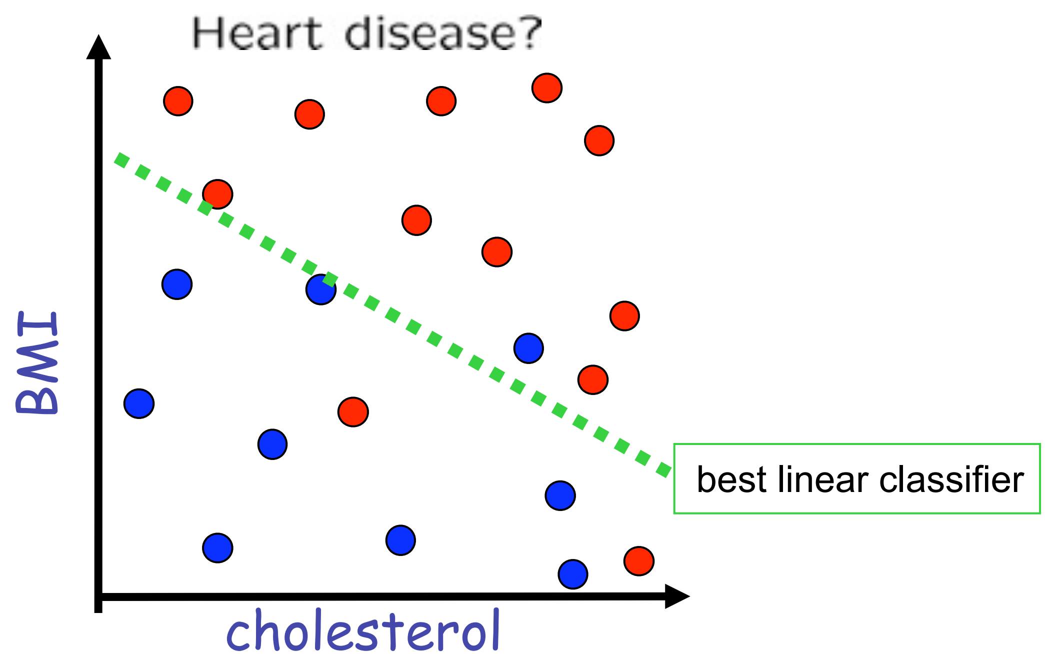
Active Learning

Learning Problem: Consider a binary prediction problem involving a collection of "classifiers." Each classifier maps points in the "feature-space" (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.



Active Learning

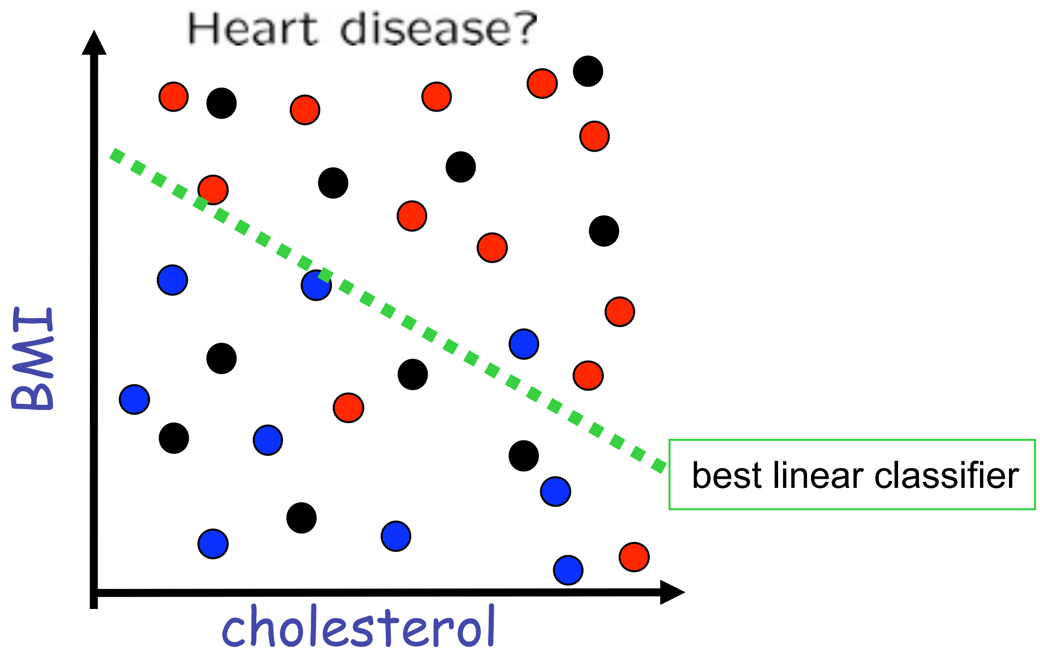
Learning Problem: Consider a binary prediction problem involving a collection of "classifiers." Each classifier maps points in the "feature-space" (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.



Standard approaches assume training data are obtained prior to learning.

Active Learning

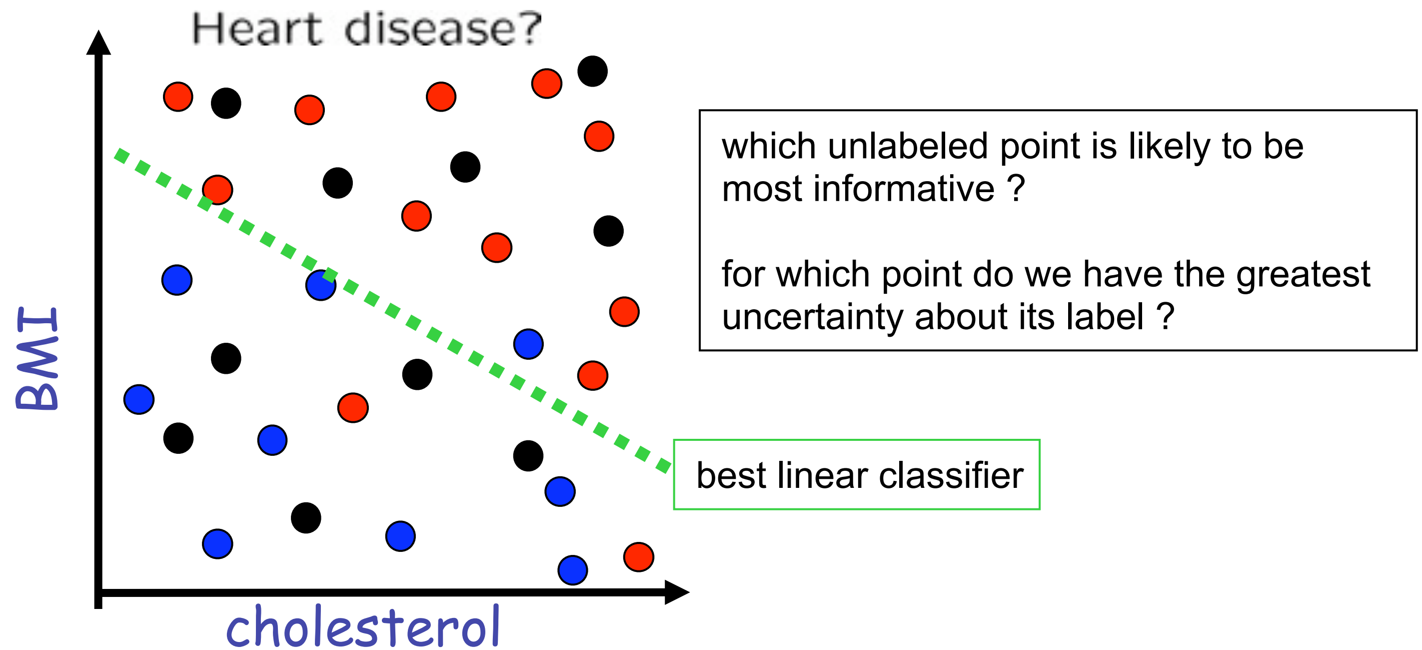
Learning Problem: Consider a binary prediction problem involving a collection of "classifiers." Each classifier maps points in the "feature-space" (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.



Standard approaches assume training data are obtained prior to learning.

Active Learning

Learning Problem: Consider a binary prediction problem involving a collection of "classifiers." Each classifier maps points in the "feature-space" (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.

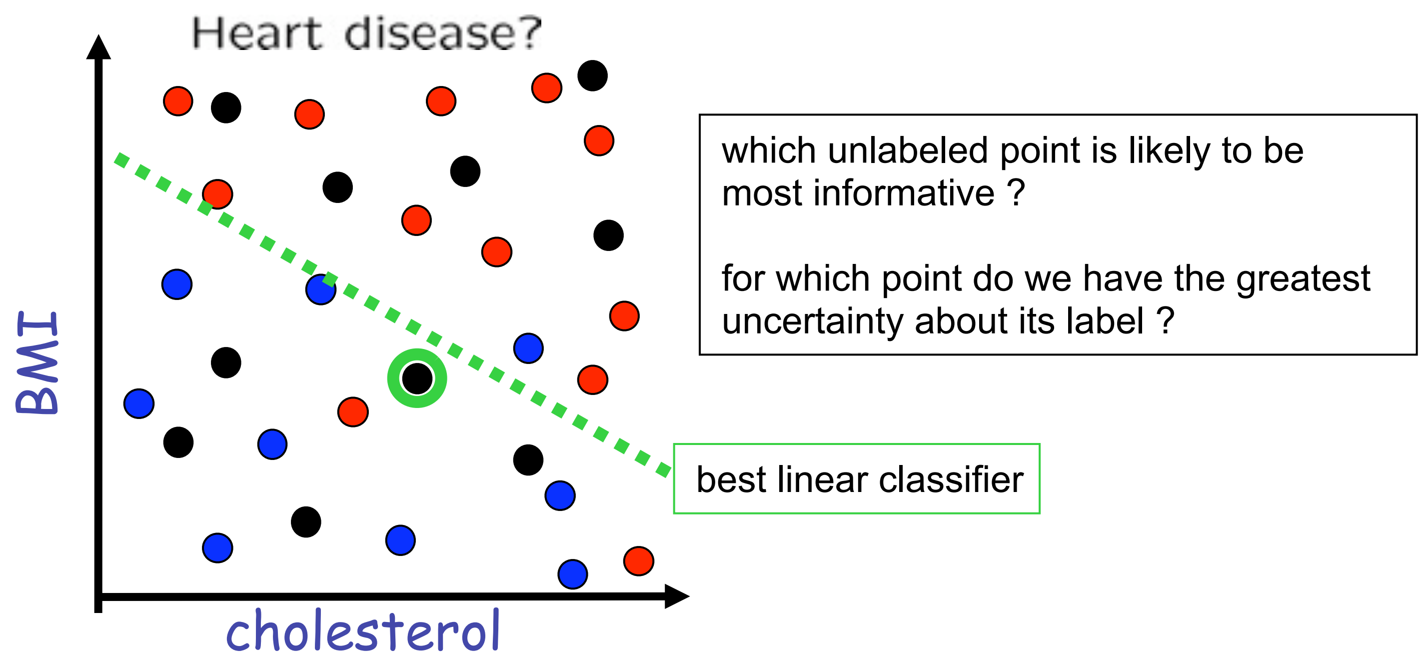


Standard approaches assume training data are obtained prior to learning.

However, some examples are more informative than others, so sequential selection of data can dramatically accelerate learning.

Active Learning

Learning Problem: Consider a binary prediction problem involving a collection of "classifiers." Each classifier maps points in the "feature-space" (e.g., $\mathbb{R}^d$) to binary labels. The features and labels are governed by an *unknown* distribution P. The goal is to select the classifier that minimizes the probability of misclassification using as few training examples as possible.

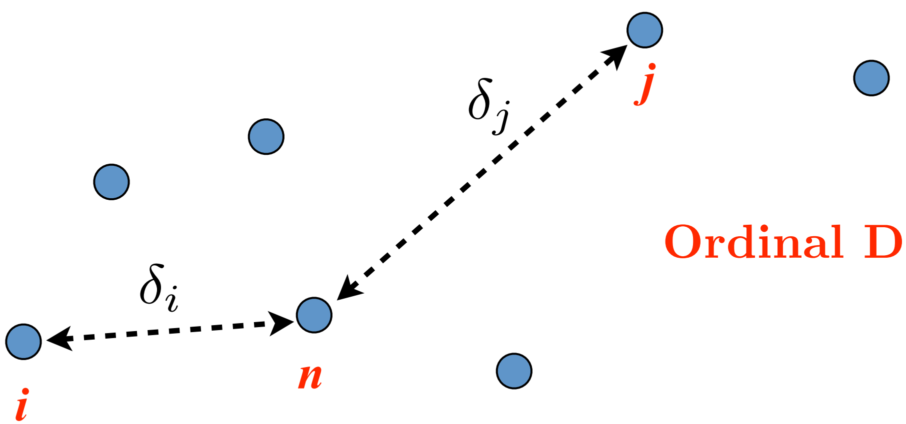


Standard approaches assume training data are obtained prior to learning.

However, some examples are more informative than others, so sequential selection of data can dramatically accelerate learning.

Ranking Based on Pairwise Comparisons

Ranking Problem: Consider a set of n objects $x_1, \ldots, x_n \in \mathbb{R}^d$. The locations of $x, \ldots, x_{n-1}$ are known, but location of x_n is unknown. To gather information about x_n, we can only ask questions of the form "is object x_n closer to x_i than x_j?" The goal is to rank $x_1, \ldots, x_{n-1}$ relative to distances to x_n by asking as few questions as possible.

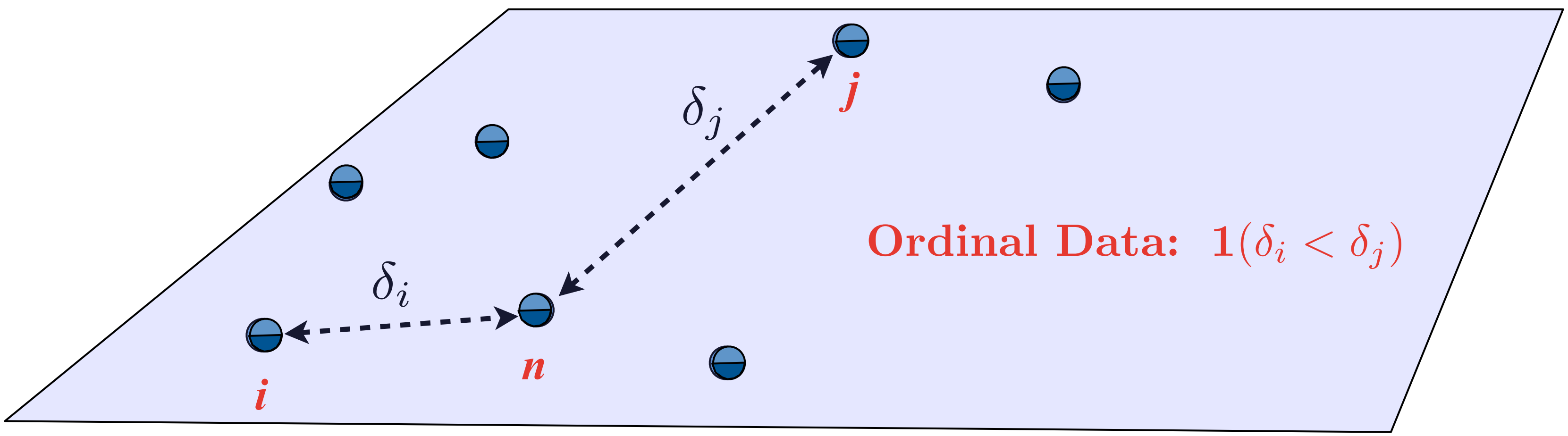


Ordinal Data: $\mathbf{1}(\delta_i < \delta_j)$

Standard sorting methods require $n \log n$ comparisons, but this can be prohibitive when n is large, especially since it is often humans who are judging the comparisons (e.g., database search).

Ranking Based on Pairwise Comparisons

Ranking Problem: Consider a set of n objects $x_1, \ldots, x_n \in \mathbb{R}^d$. The locations of $x, \ldots, x_{n-1}$ are known, but location of x_n is unknown. To gather information about x_n, we can only ask questions of the form "is object x_n closer to x_i than x_j?" The goal is to rank $x_1, \ldots, x_{n-1}$ relative to distances to x_n by asking as few questions as possible.

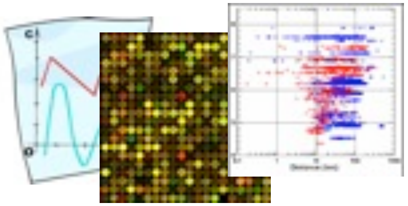


Standard sorting methods require $n \log n$ comparisons, but this can be prohibitive when n is large, especially since it is often humans who are judging the comparisons (e.g., database search).

However, 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.

Machine Learning (Passive)

Raw unlabeled data



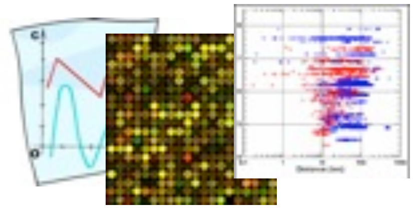
$X_1, X_2, X_3, \ldots$

passive learner

expert/oracle
analyzes/experiments
to determine labels

Machine Learning (Passive)

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

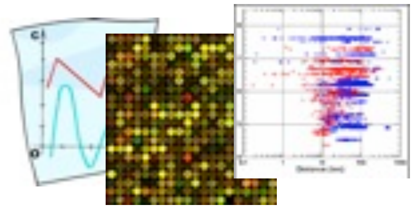
passive learner

expert/oracle

analyzes/experiments
to determine labels

Machine Learning (Passive)

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

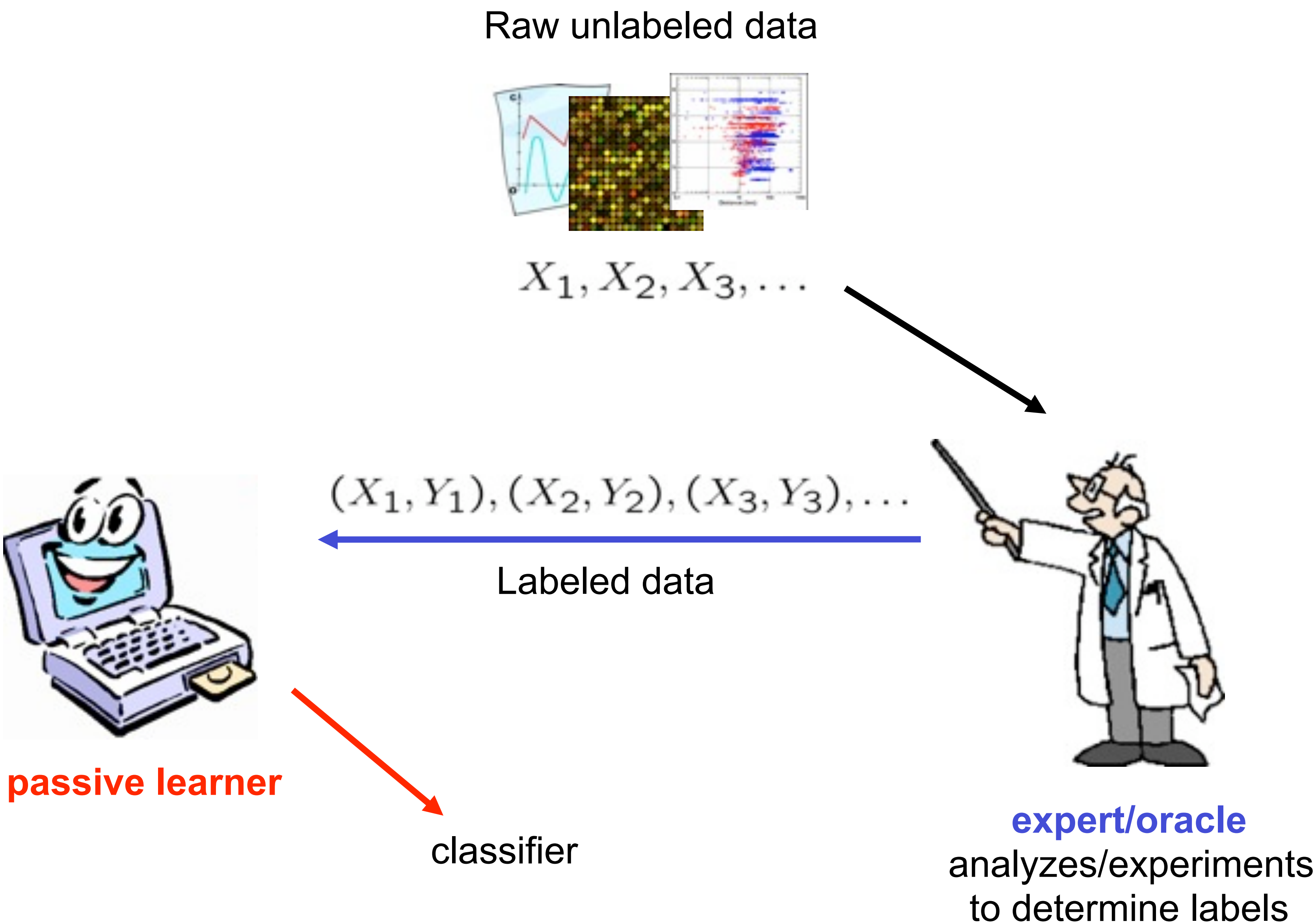
$(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots$

Labeled data

passive learner

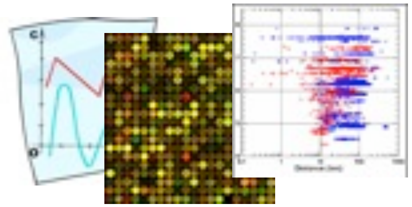
expert/oracle
analyzes/experiments
to determine labels

Machine Learning (Passive)



Active Learning

Raw unlabeled data



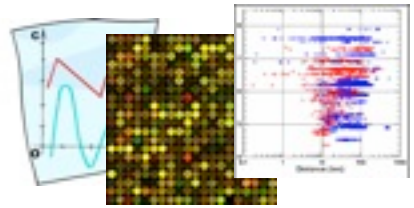
$X_1, X_2, X_3, \ldots$

active learner

expert/oracle
analyzes/experiments
to determine labels

Active Learning

Raw unlabeled data



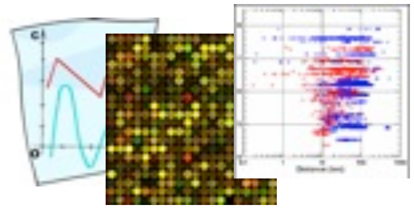
$X_1, X_2, X_3, \ldots$

active learner

expert/oracle
analyzes/experiments
to determine labels

Active Learning

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

Learner requests labels
for **selected** data

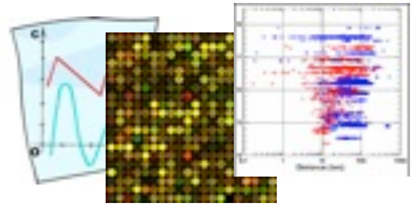
$(X_1, ?)$

active learner

expert/oracle
analyzes/experiments
to determine labels

Active Learning

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

Learner requests labels
for **selected** data

$(X_1, ?)$

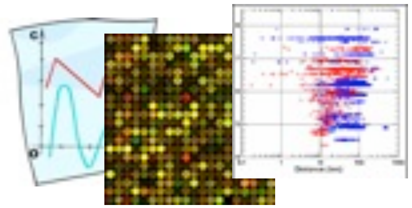
(X_1, Y_1)

active learner

expert/oracle
analyzes/experiments
to determine labels

Active Learning

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

Learner requests labels for **selected** data

$(X_1, ?)$

(X_1, Y_1)

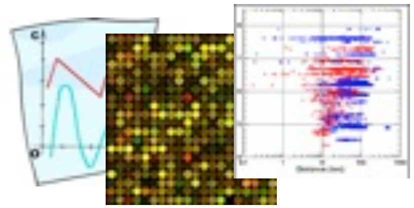
$(X_3, ?)$

active learner

expert/oracle
analyzes/experiments
to determine labels

Active Learning

Raw unlabeled data



$X_1, X_2, X_3, \ldots$

Learner requests labels for **selected** data

active learner

$(X_1, ?)$

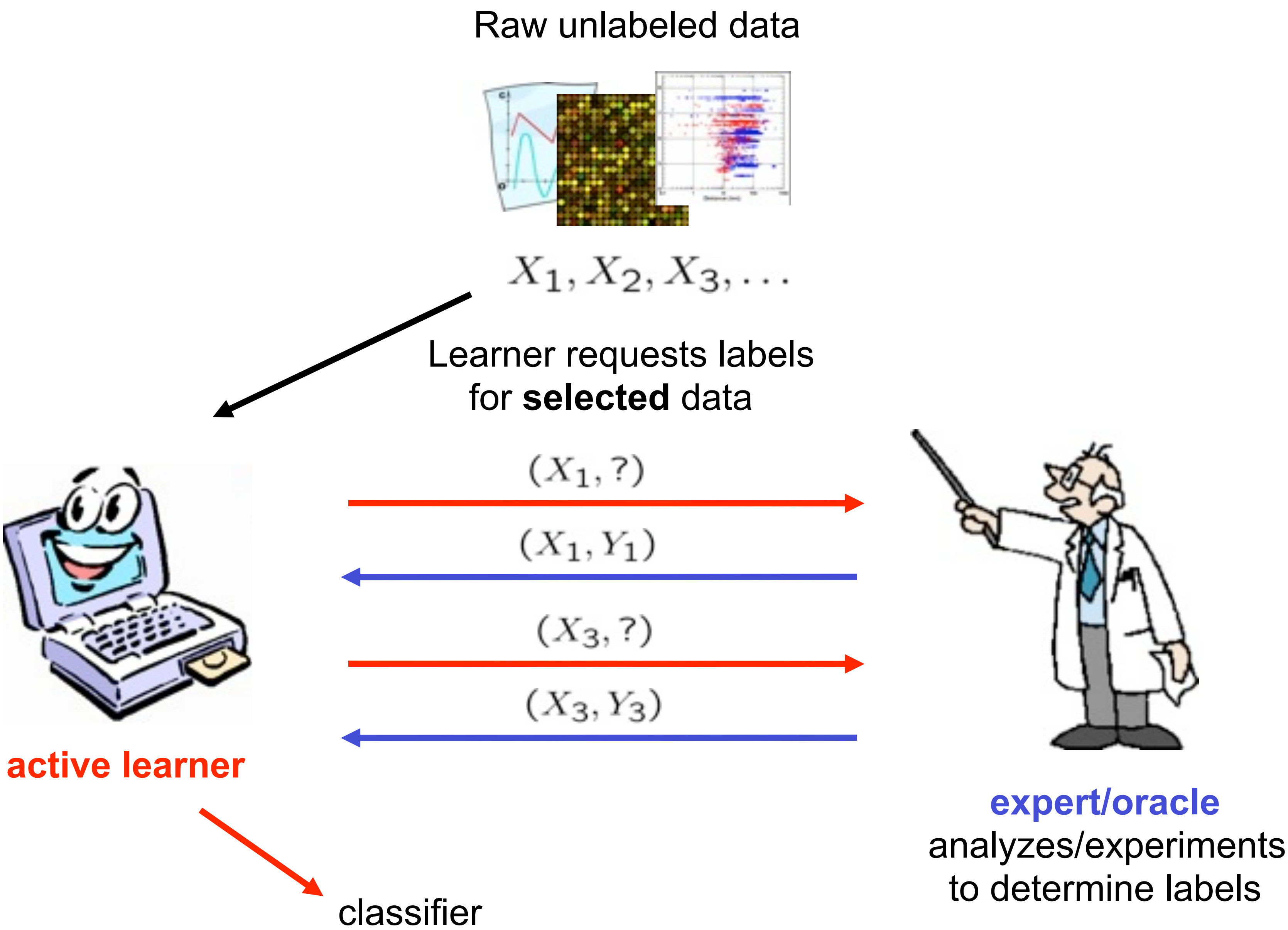
(X_1, Y_1)

$(X_3, ?)$

(X_3, Y_3)

expert/oracle
analyzes/experiments to determine labels

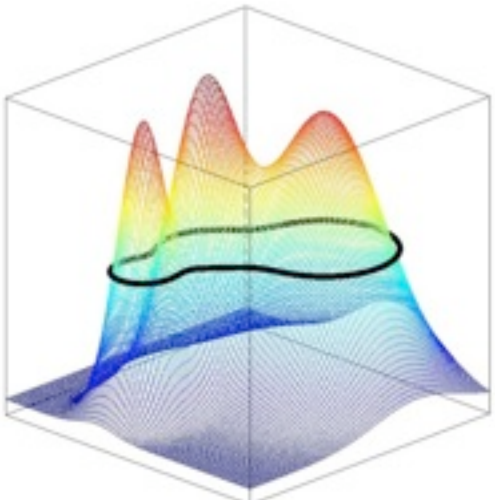
Active Learning



Binary Classification

$\mathcal{X}$:= *feature* space, typically $\mathbb{R}^d$

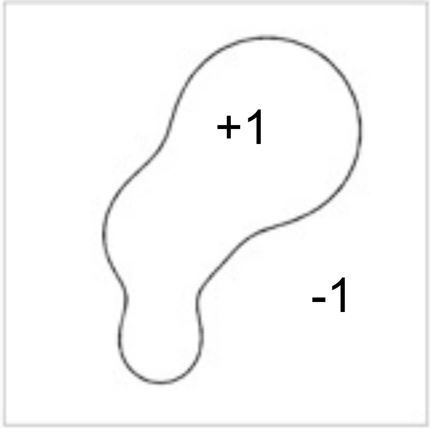
$\mathcal{Y}$:= $\{-1, +1\}$



$\mathbb{P}(Y = 1|X = x)$

unknown

1/2-level set is optimal decision boundary

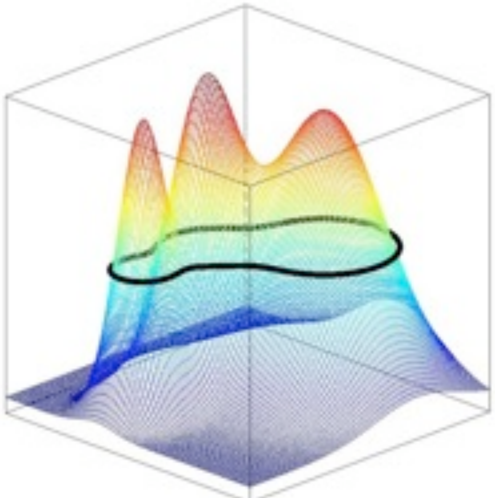


optimal decision set

Binary Classification

$\mathcal{X}$:= *feature* space, typically $\mathbb{R}^d$

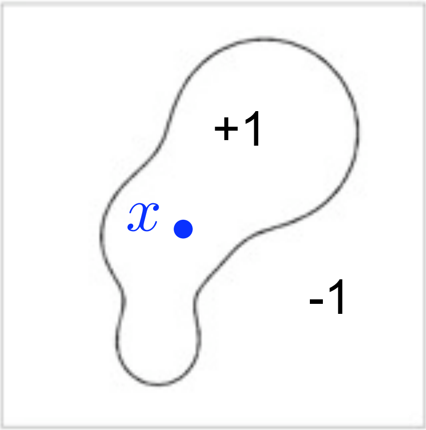
$\mathcal{Y}$:= $\{-1, +1\}$



$\mathbb{P}(Y = 1 | X = x)$

unknown

1/2-level set is optimal decision boundary



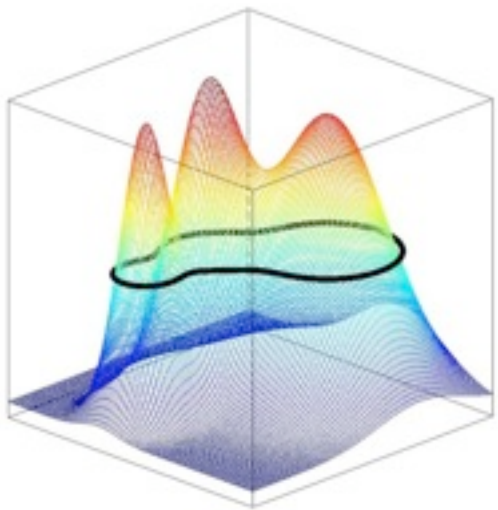
optimal decision set

allowable questions:
is x in the set?

Binary Classification

$\mathcal{X} := \textit{feature}$ space, typically $\mathbb{R}^d$

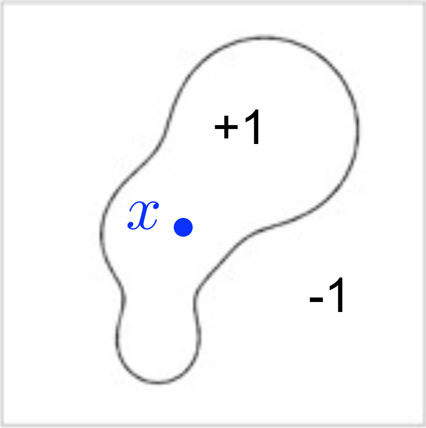
$\mathcal{Y} := \{-1, +1\}$



$\mathbb{P}(Y = 1|X = x)$

unknown

1/2-level set is optimal decision boundary



optimal decision set

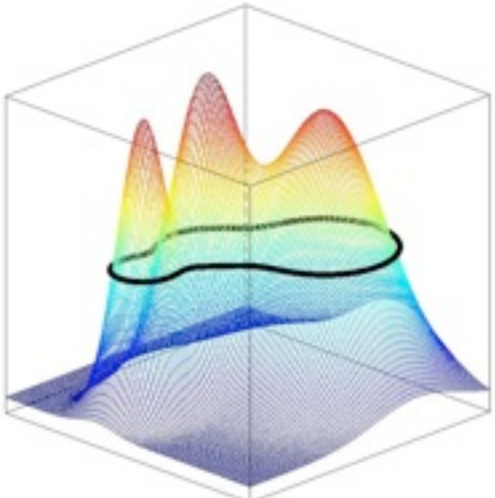
allowable questions: is x in the set?

Problem boils down to learning a set through simple “membership” queries

Binary Classification

$\mathcal{X}$:= *feature* space, typically $\mathbb{R}^d$

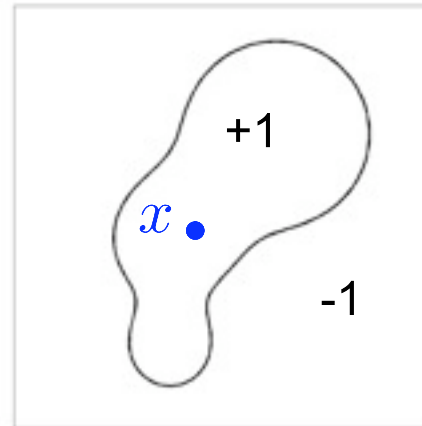
$\mathcal{Y}$:= $\{-1, +1\}$



$\mathbb{P}(Y = 1 | X = x)$

unknown

1/2-level set is optimal decision boundary



optimal decision set

allowable questions: is x in the set?

Problem boils down to learning a set through simple "membership" queries

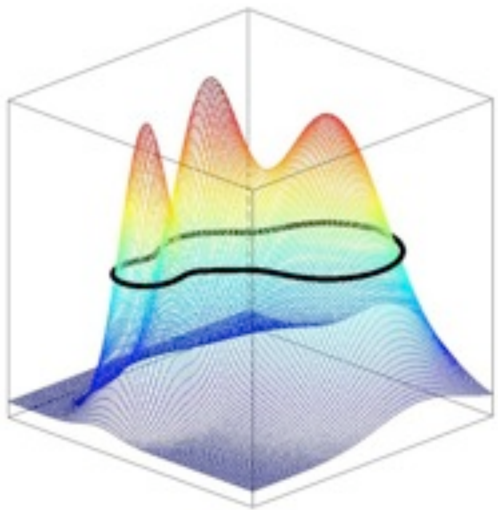
Key Questions:

1. When can active learning provide reductions in sample complexity?
2. What active learning strategies/policies are optimal?

Binary Classification

$\mathcal{X}$:= *feature* space, typically $\mathbb{R}^d$

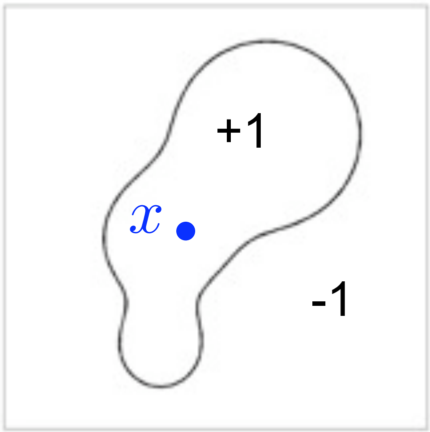
$\mathcal{Y}$:= $\{-1, +1\}$



$\mathbb{P}(Y = 1|X = x)$

unknown

1/2-level set is optimal decision boundary



optimal decision set

allowable questions: is x in the set?

Problem boils down to learning a set through simple "membership" queries

Key Questions:

1. When can active learning provide reductions in sample complexity?

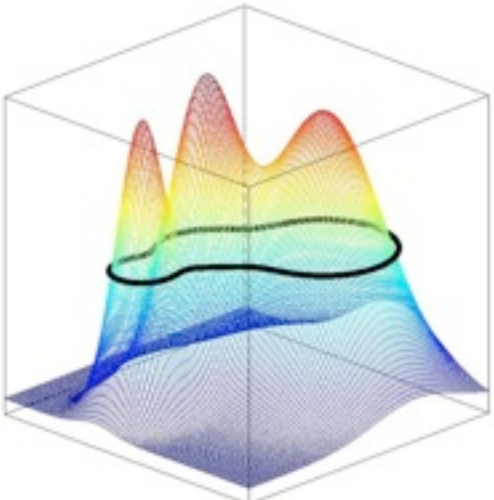
2. What active learning strategies/policies are optimal?

R. Castro, RN: *Minimax Bounds for Active Learning*. IEEE Transactions on Information Theory, 2008.

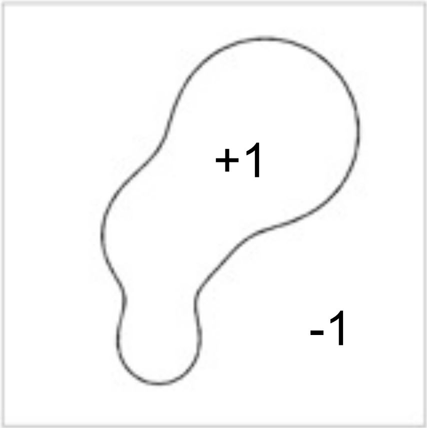
M. Raginsky and S. Rahklin: *Lower Bounds for Passive and Active Learning,* NIPS 2011

Lower Bounds on Sample Complexity

Key complexity parameters



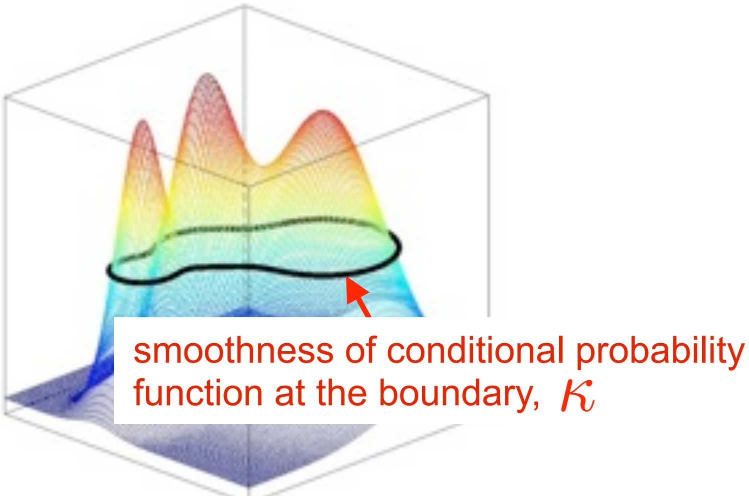
$\mathbb{P}(Y = 1 | X = x)$



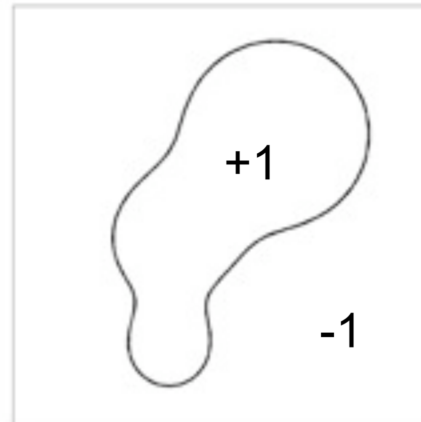
optimal decision set

Lower Bounds on Sample Complexity

Key complexity parameters



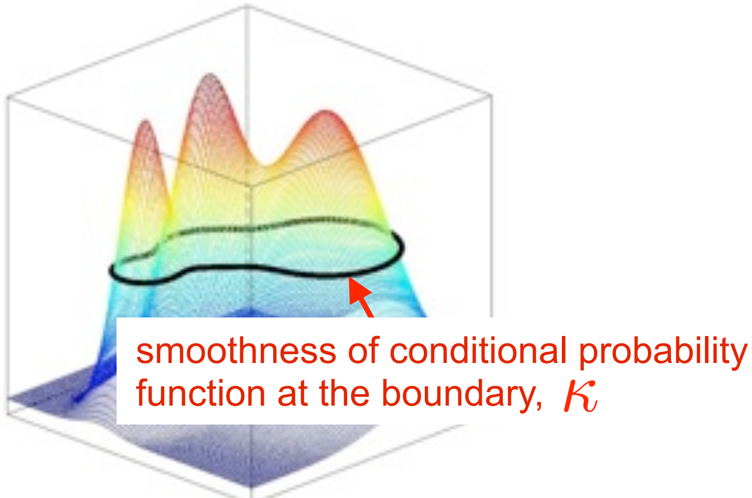
$\mathbb{P}(Y = 1|X = x)$



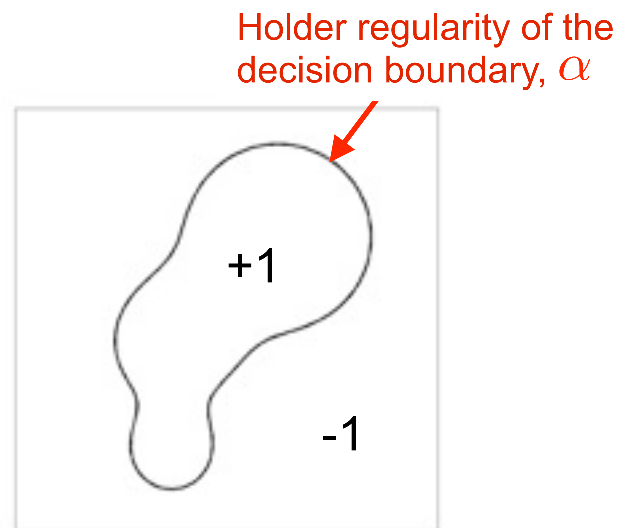
optimal decision set

Lower Bounds on Sample Complexity

Key complexity parameters



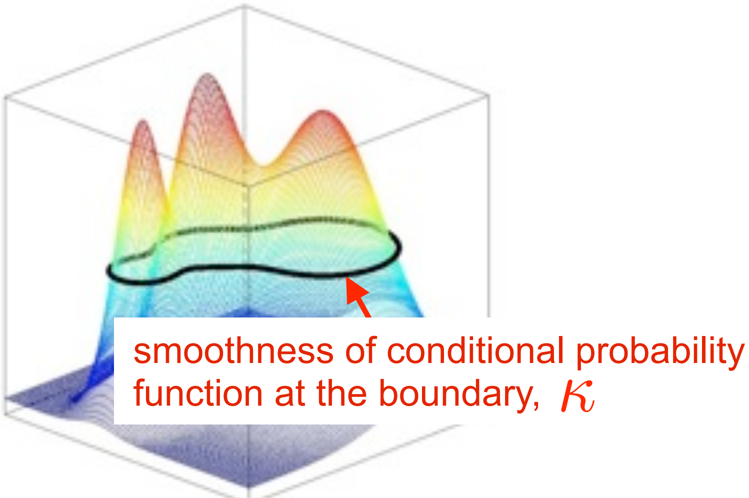
$\mathbb{P}(Y = 1 | X = x)$



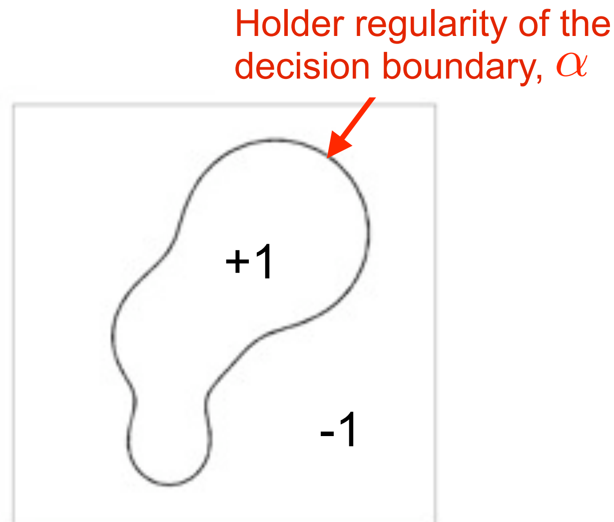
optimal decision set

Lower Bounds on Sample Complexity

Key complexity parameters



$\mathbb{P}(Y = 1|X = x)$

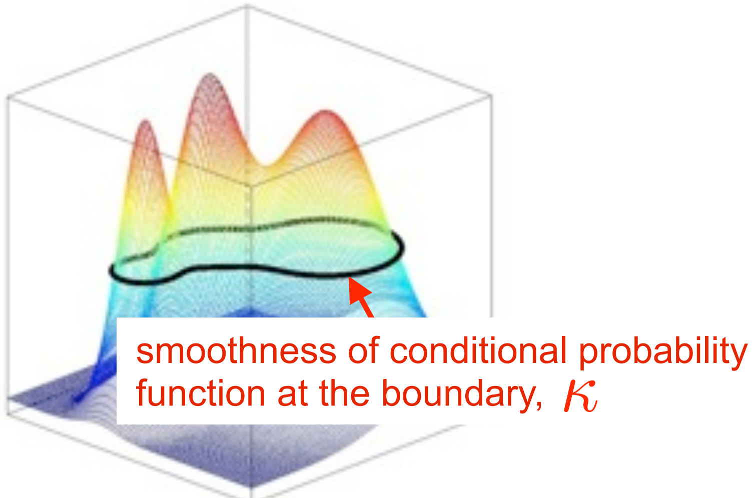


optimal decision set

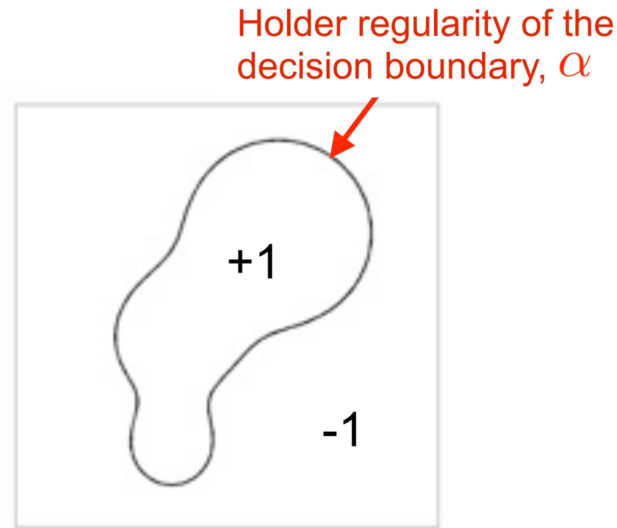
training examples: $\{(x_i, y_i)\}_{i=1}^n$ selected sequentially and adaptively (active learning) or at random (passive learning)

Lower Bounds on Sample Complexity

Key complexity parameters



$\mathbb{P}(Y = 1 | X = x)$



optimal decision set

training examples: $\{(x_i, y_i)\}_{i=1}^n$ selected sequentially and adaptively (active learning) or at random (passive learning)

minimax rate of convergence to Bayes error:

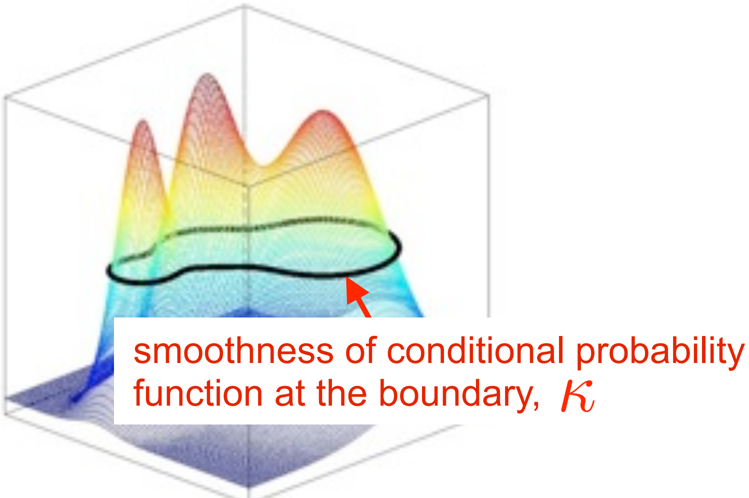
$$\text{Active:} \qquad n^{-\frac{\kappa}{2\kappa+\rho-2}} \qquad\qquad \rho := \frac{d-1}{\alpha}$$

$$\text{Passive:} \qquad n^{-\frac{\kappa}{2\kappa+\rho-1}}$$

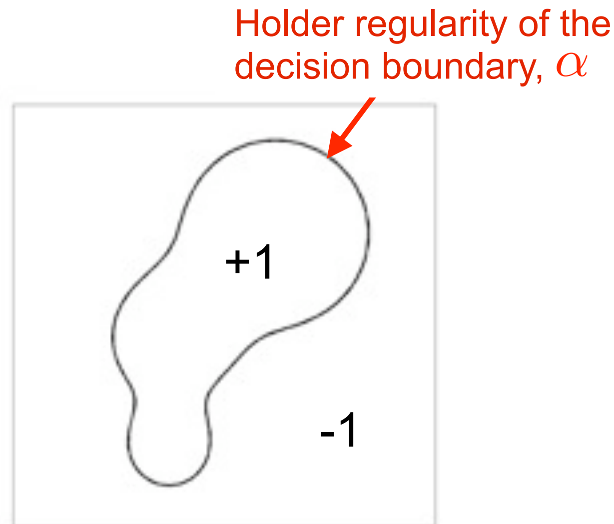
proof ingredients: Fano's inequality, Varshamov-Gilbert Bound

Lower Bounds on Sample Complexity

Key complexity parameters



$\mathbb{P}(Y = 1 | X = x)$



optimal decision set

training examples: $\{(x_i, y_i)\}_{i=1}^n$ selected sequentially and adaptively (active learning) or at random (passive learning)

minimax rate of convergence to Bayes error:

Active: $n^{-\frac{\kappa}{2\kappa+\rho-2}}$

Passive: $n^{-\frac{\kappa}{2\kappa+\rho-1}}$

$\rho := \frac{d-1}{\alpha}$

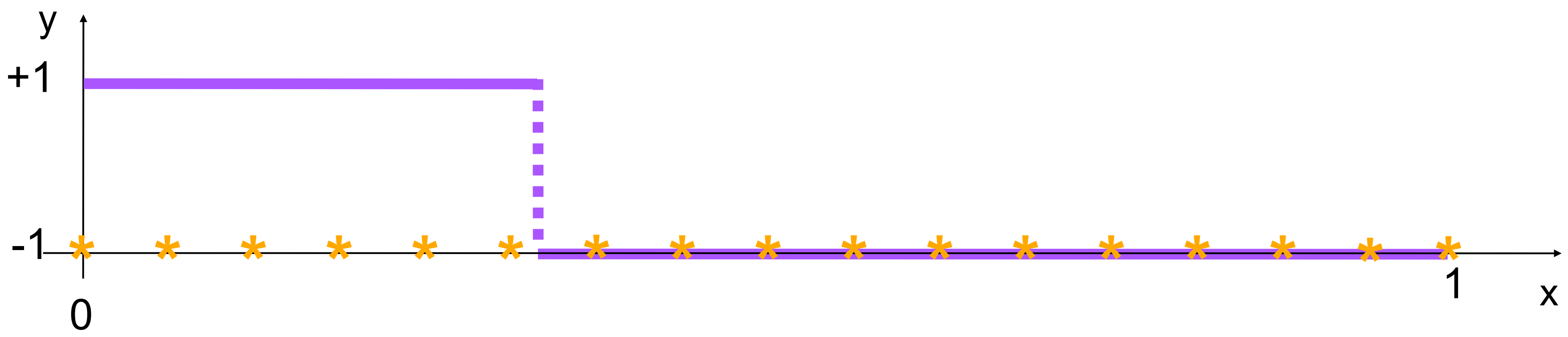
as $\rho \to 0$
and $\kappa \to 1$

active learning yields exponential improvement!

proof ingredients: Fano’s inequality, Varshamov-Gilbert Bound

Classic Binary Search

Classic Binary Search



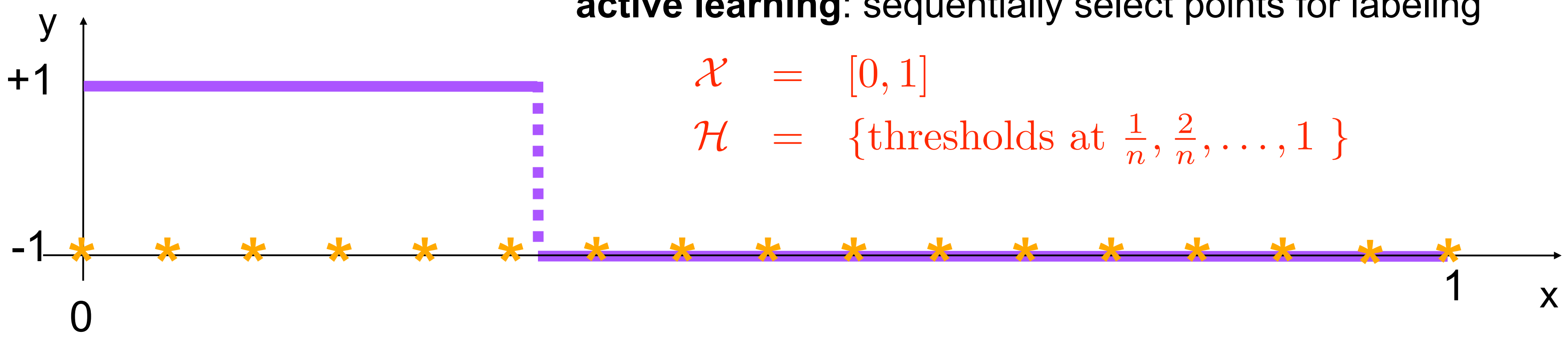
Classic Binary Search

active learning: sequentially select points for labeling

Classic Binary Search

active learning: sequentially select points for labeling

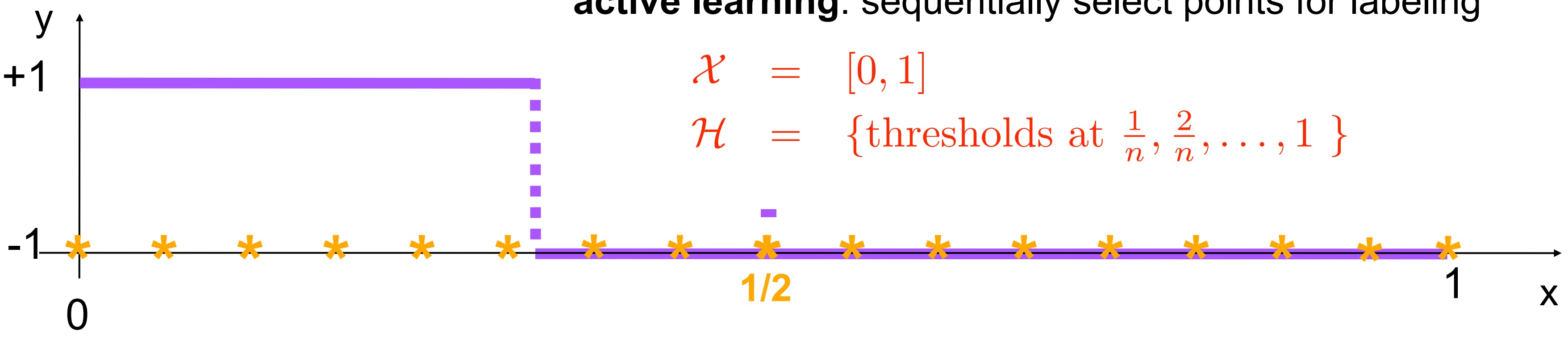
$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$



Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$

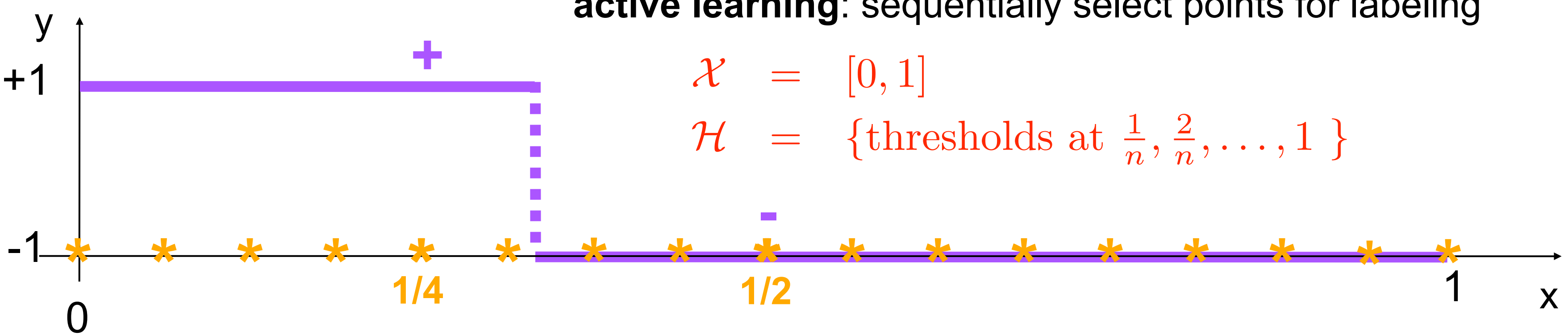


1/3 = 0…

Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0, 1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1 \}$$

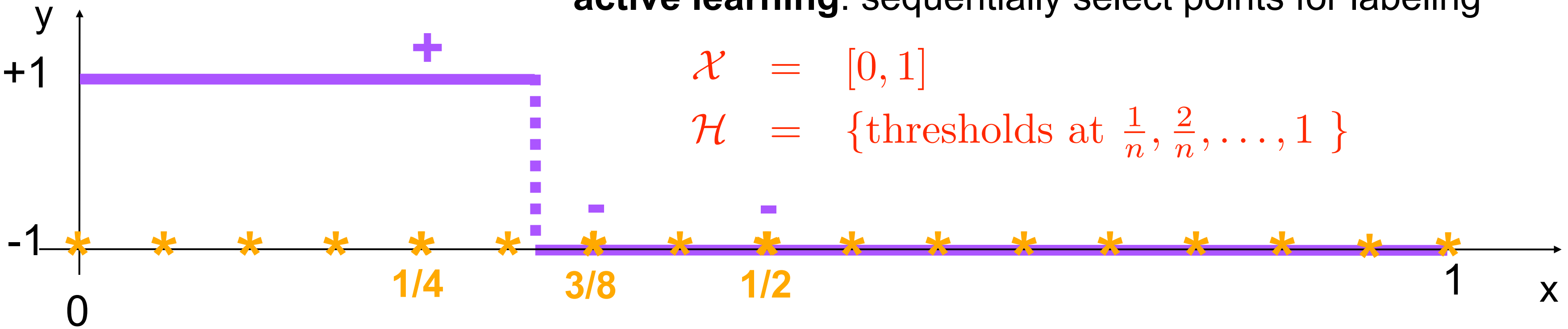


1/3 = 01…

Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$

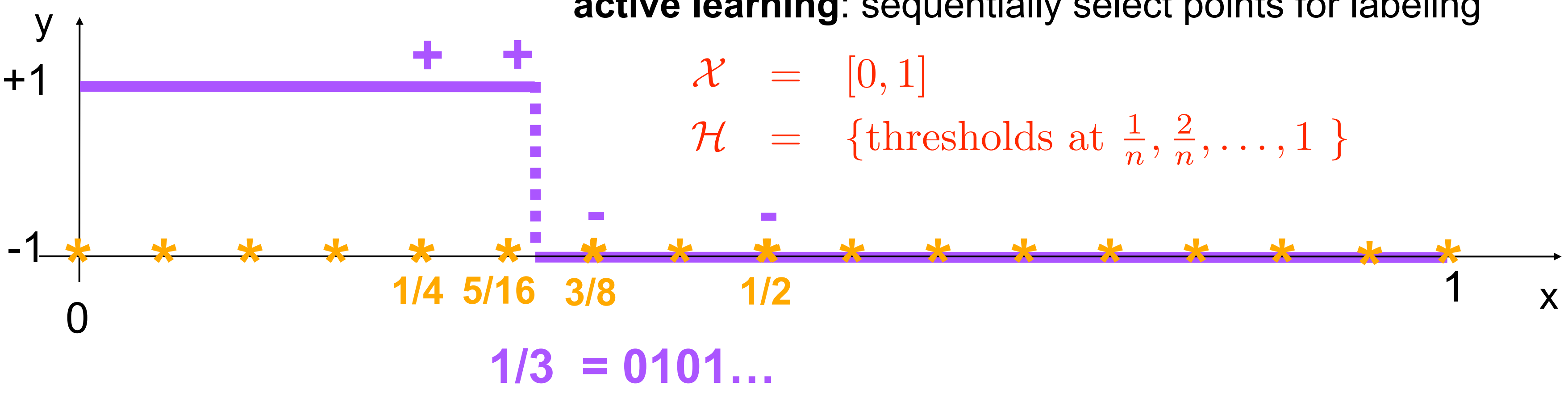


1/3 = 010…

Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$

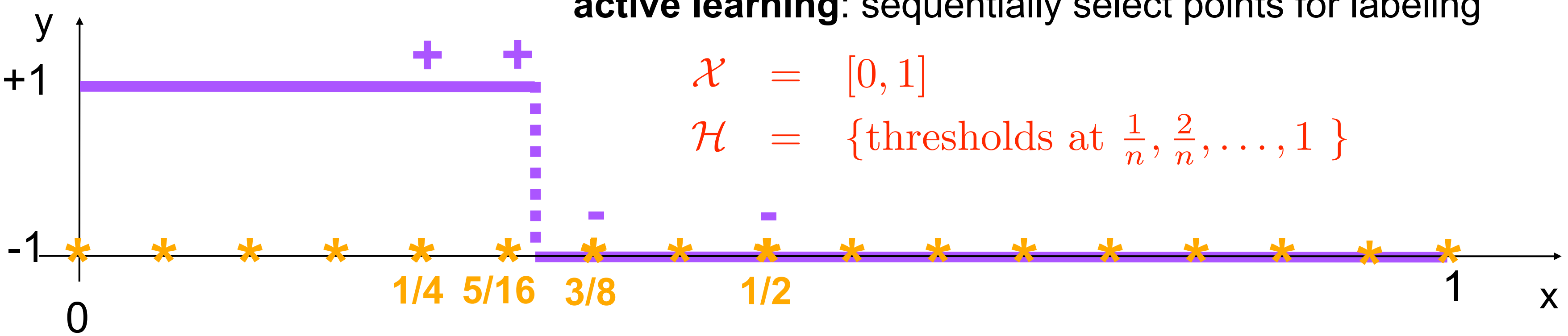


Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0,1]$$

$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$



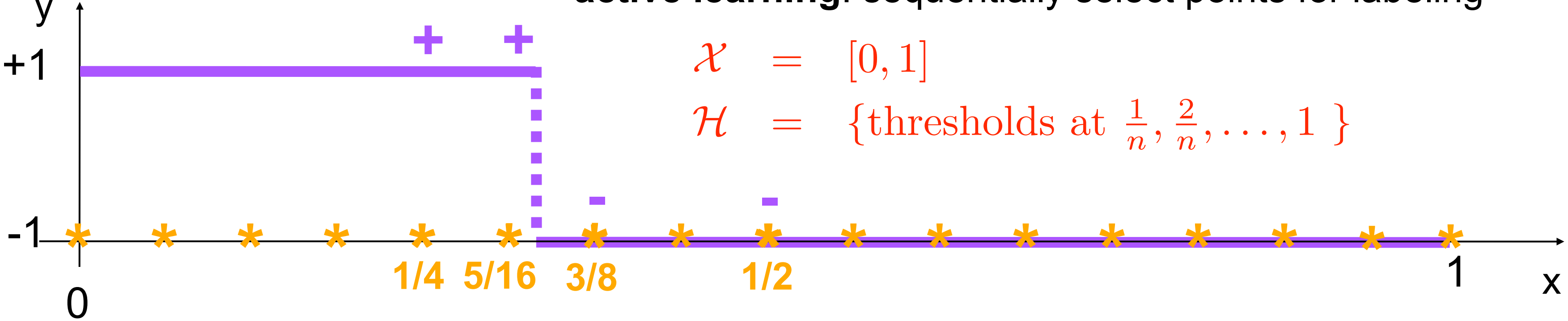
1/3 = 0101…

requires $\log_2 N$ queries

Classic Binary Search

active learning: sequentially select points for labeling

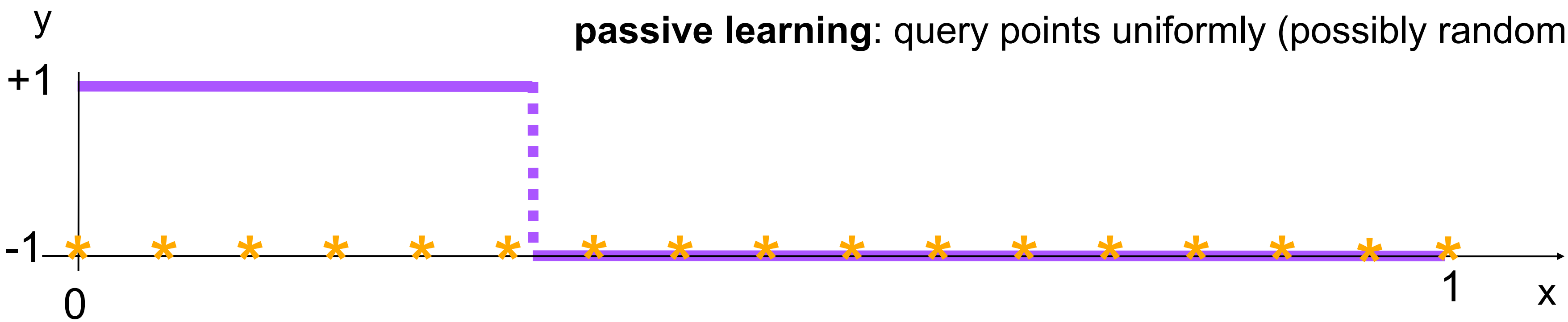
$$\mathcal{X} = [0, 1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$



1/3 = 0101…

requires $\log_2 N$ queries

passive learning: query points uniformly (possibly random)

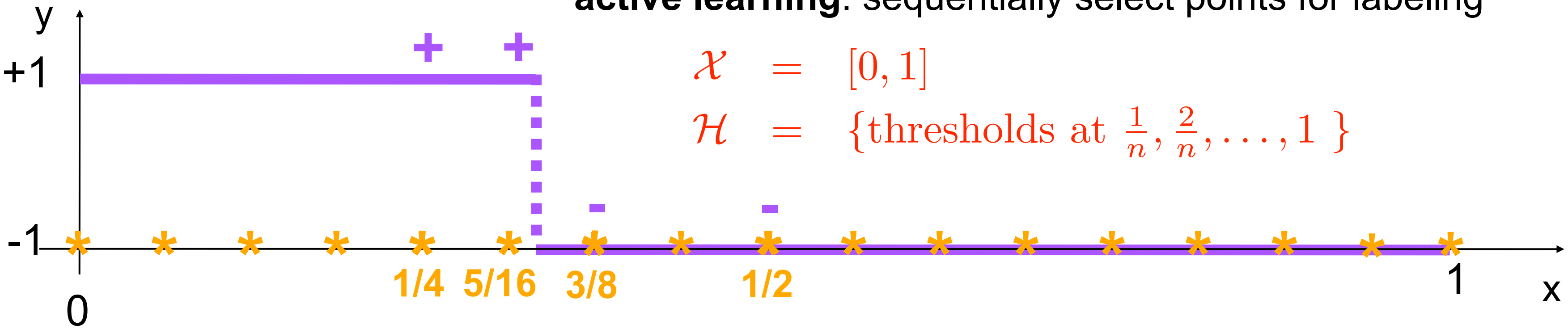


Classic Binary Search

active learning: sequentially select points for labeling

$$\mathcal{X} = [0, 1]$$

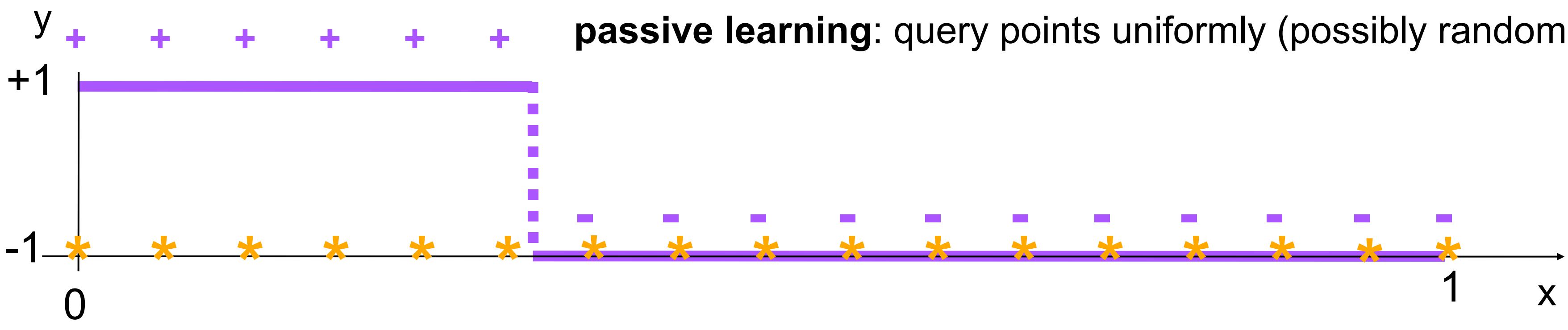
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$



1/3 = 0101…

requires $\log_2 N$ queries

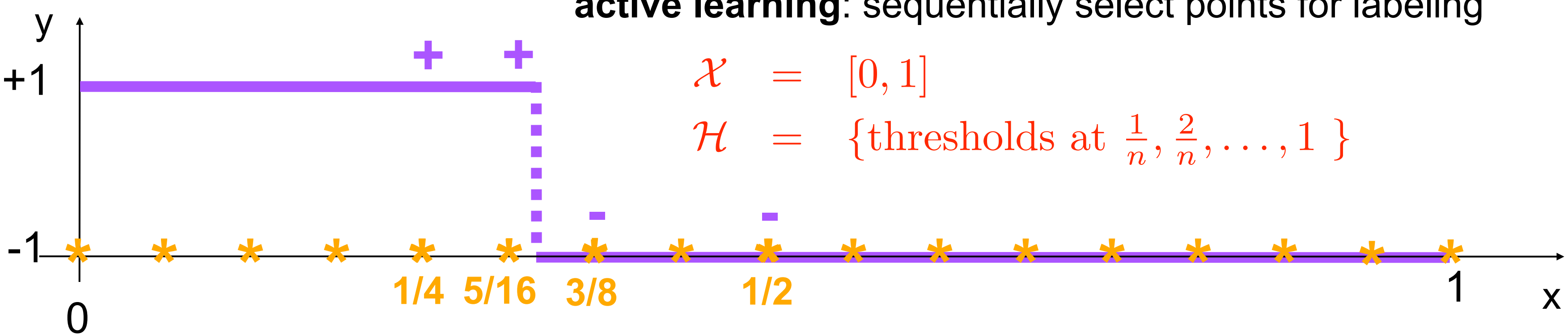
passive learning: query points uniformly (possibly random)



Classic Binary Search

active learning: sequentially select points for labeling

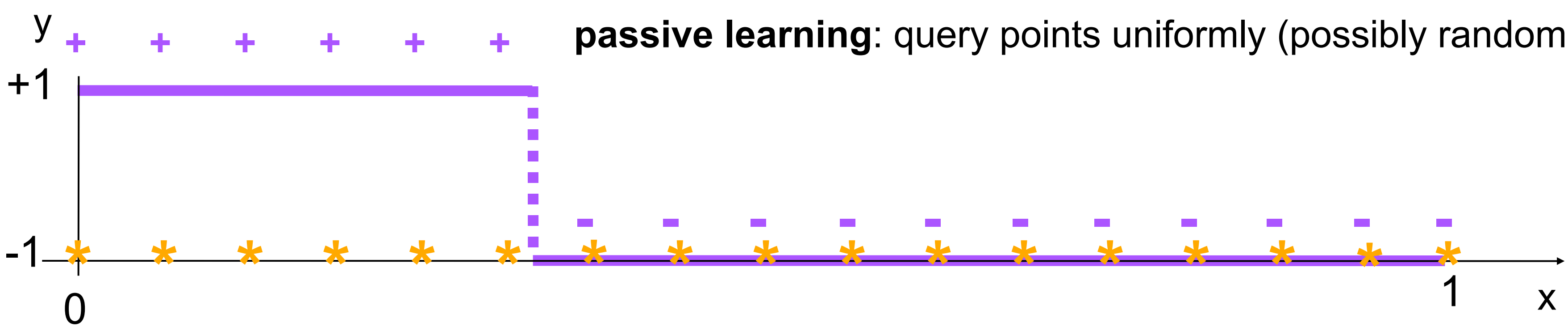
$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1\ \}$$



1/3 = 0101…

requires $\log_2 N$ queries

passive learning: query points uniformly (possibly random)

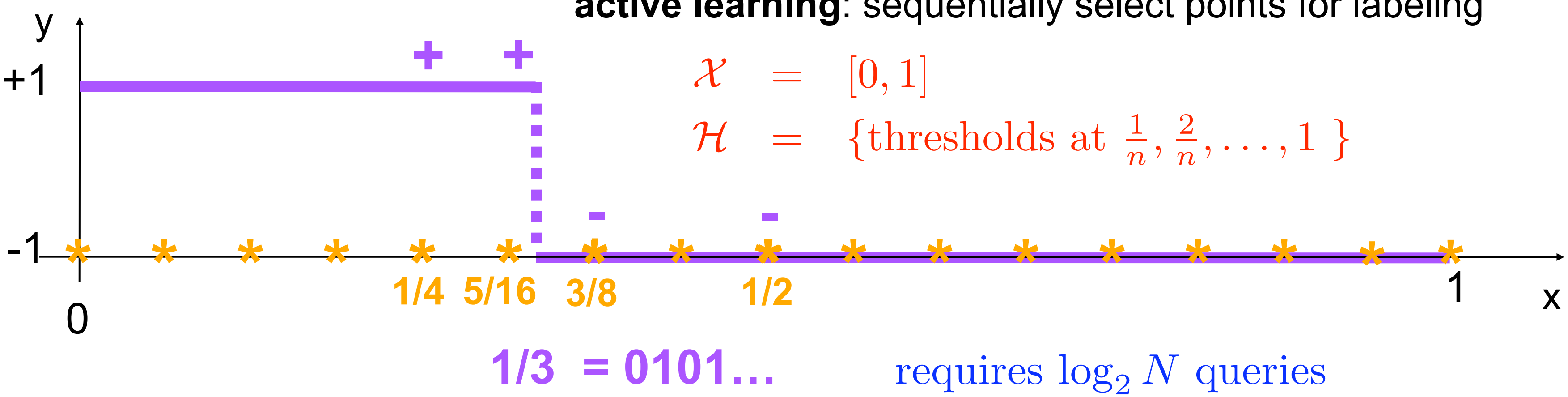


requires $O(N)$ queries

Classic Binary Search

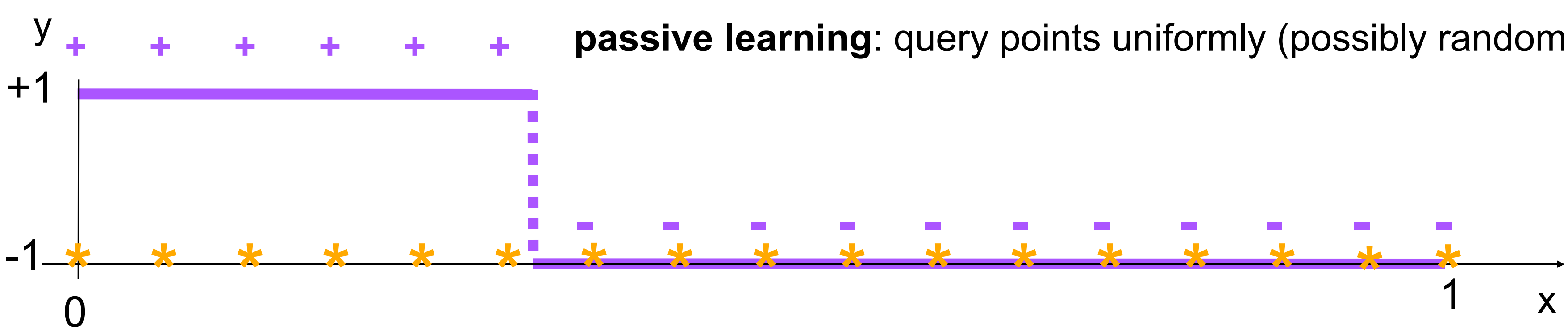
active learning: sequentially select points for labeling

$$\mathcal{X} = [0,1]$$
$$\mathcal{H} = \{\text{thresholds at } \tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1 \}$$



requires $\log_2 N$ queries

passive learning: query points uniformly (possibly random)



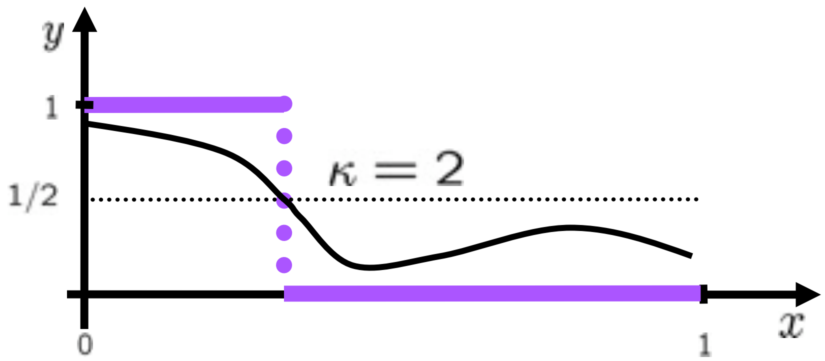
requires $O(N)$ queries

active learning is dramatically more efficient

Rates of Convergence to Bayes

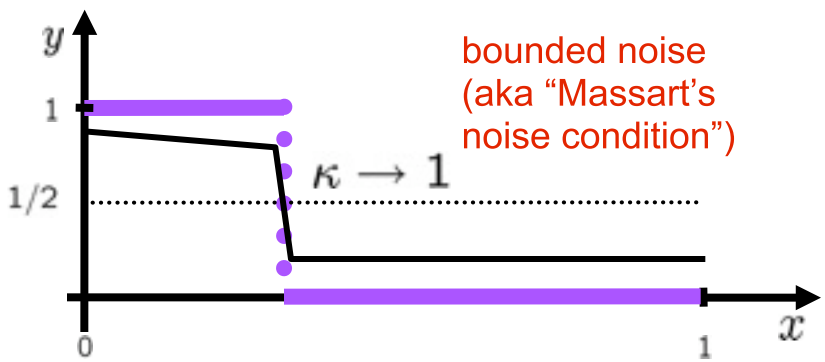
Active: $n^{-\frac{\kappa}{2\kappa-2}}$

Passive: $n^{-\frac{\kappa}{2\kappa-1}}$



passive: $n^{-2/3}$

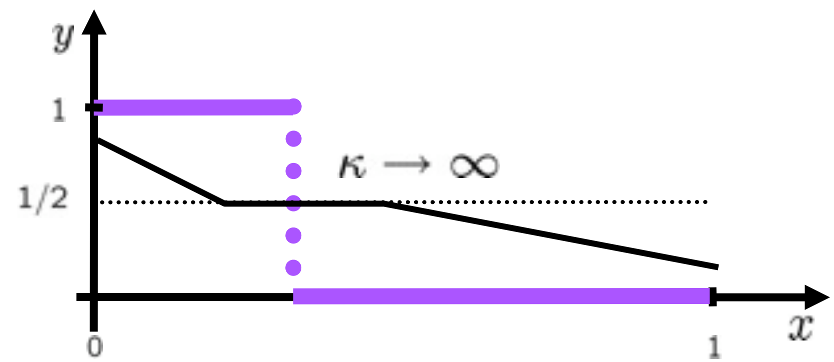
active: n^{-1}



bounded noise (aka “Massart’s noise condition”)

passive: $\to n^{-1}$

active: $\to e^{-cn}$

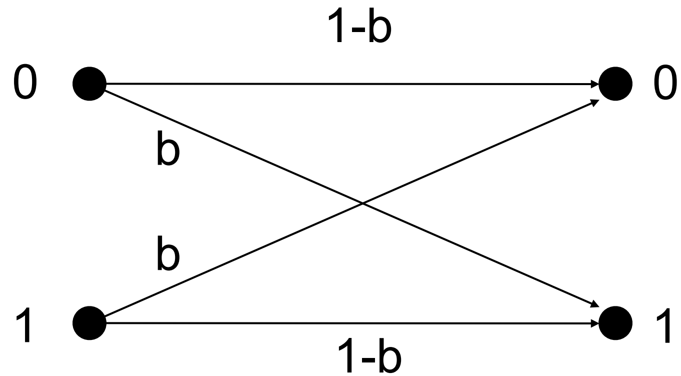


passive: $\to n^{-1/2}$

active: $\to n^{-1/2}$

Noisy Binary Search: Channel Coding with Noiseless Feedback

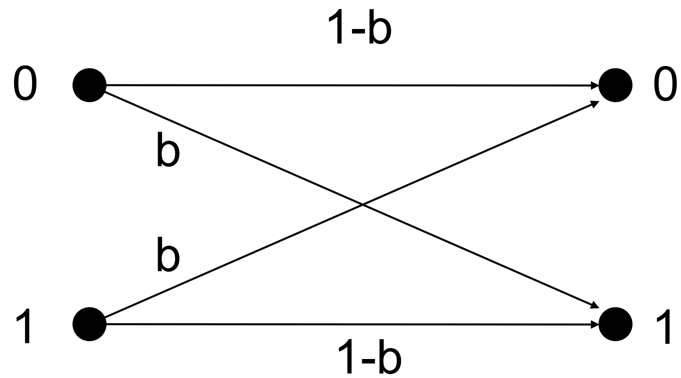
sender



receiver

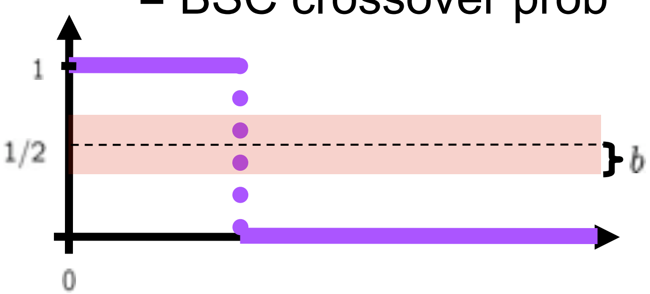
Noisy Binary Search: Channel Coding with Noiseless Feedback

sender



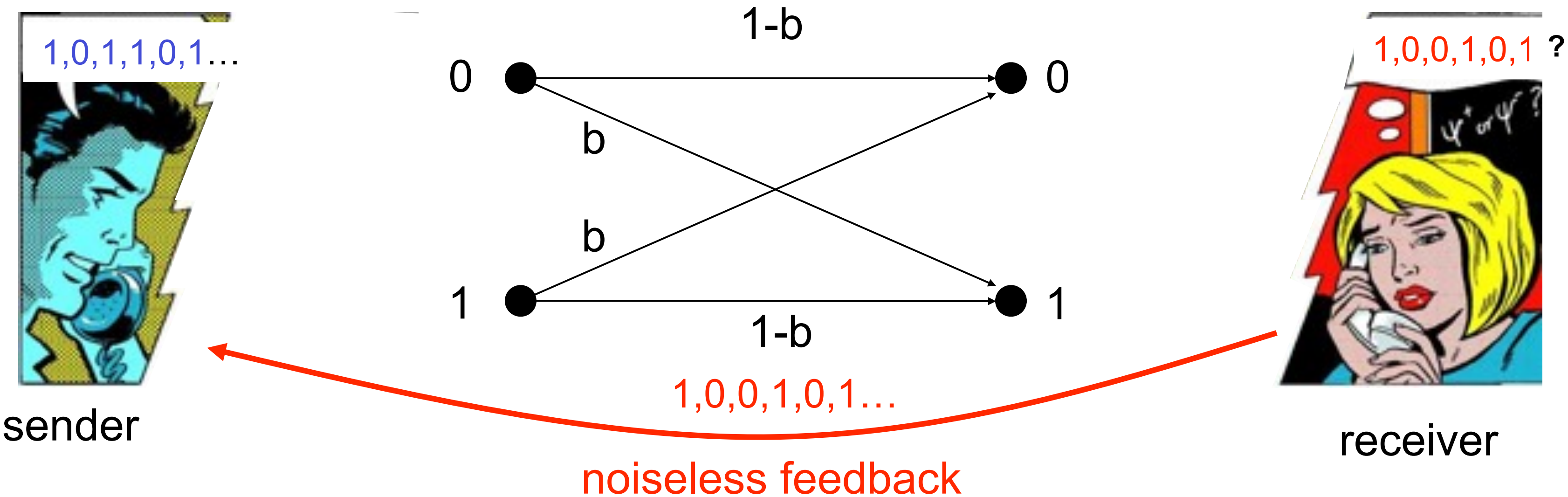
receiver

noise bound
= BSC crossover prob

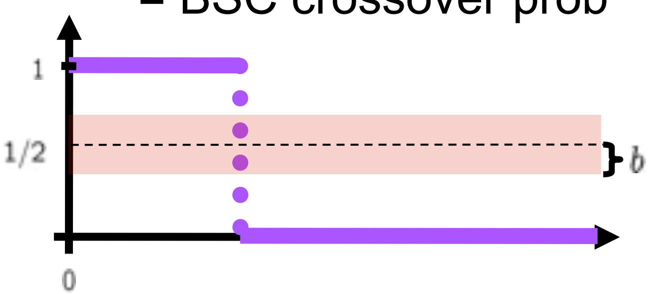


threshold location
= n bit message

Noisy Binary Search: Channel Coding with Noiseless Feedback

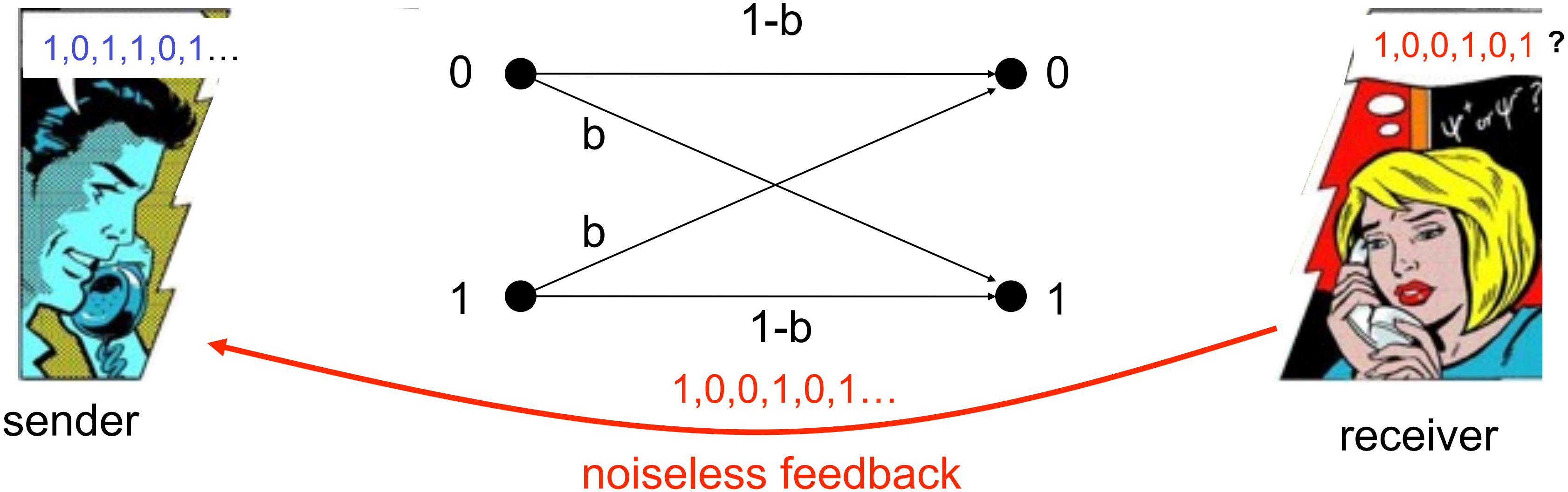


noise bound
= BSC crossover prob

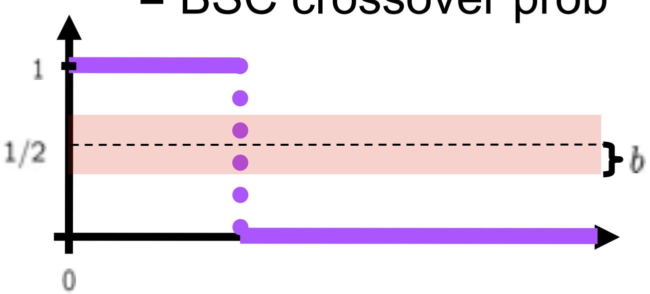


threshold location
= n bit message

Noisy Binary Search: Channel Coding with Noiseless Feedback



noise bound
= BSC crossover prob

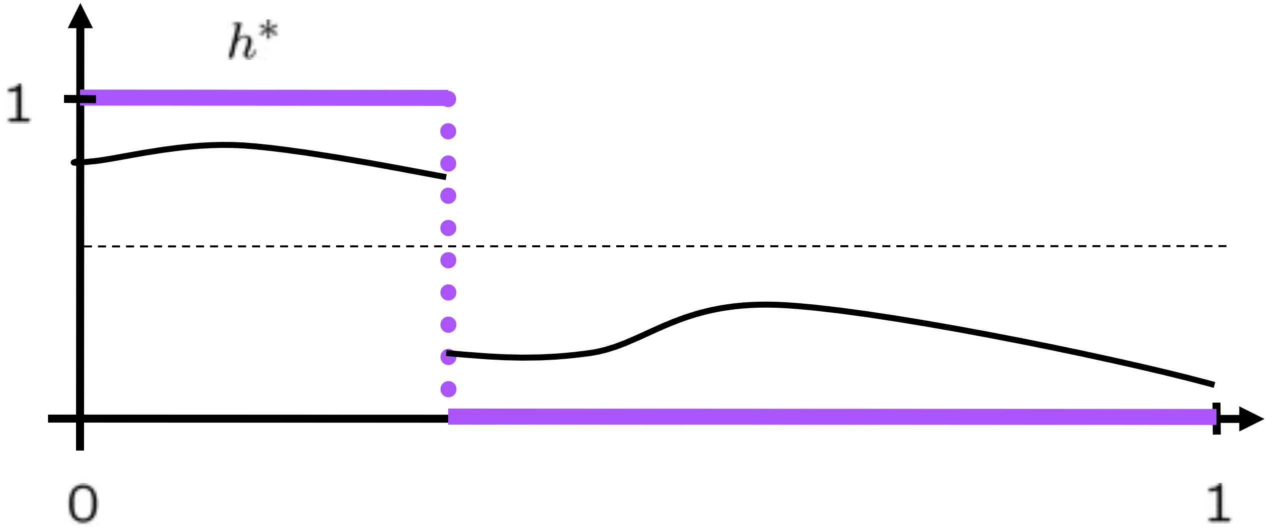


threshold location
= n bit message

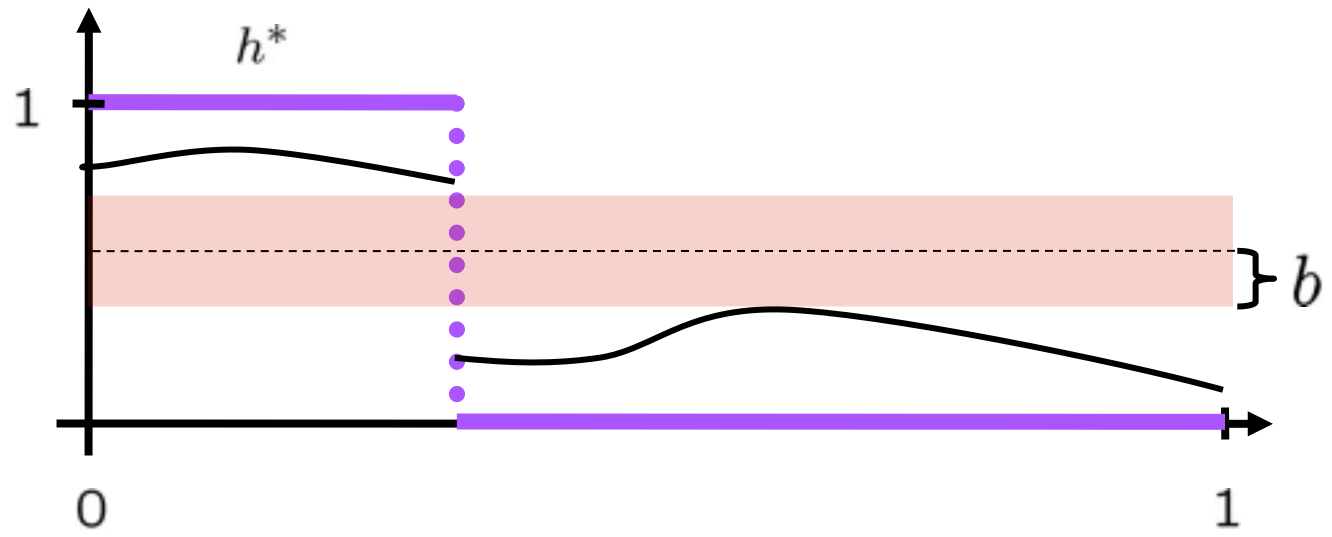
Both sender and receiver implement Horstein's algorithm

Sender deduces which binary symbol to send next in order to yield the greatest possible expected reduction in the receiver's uncertainty about n-bit message

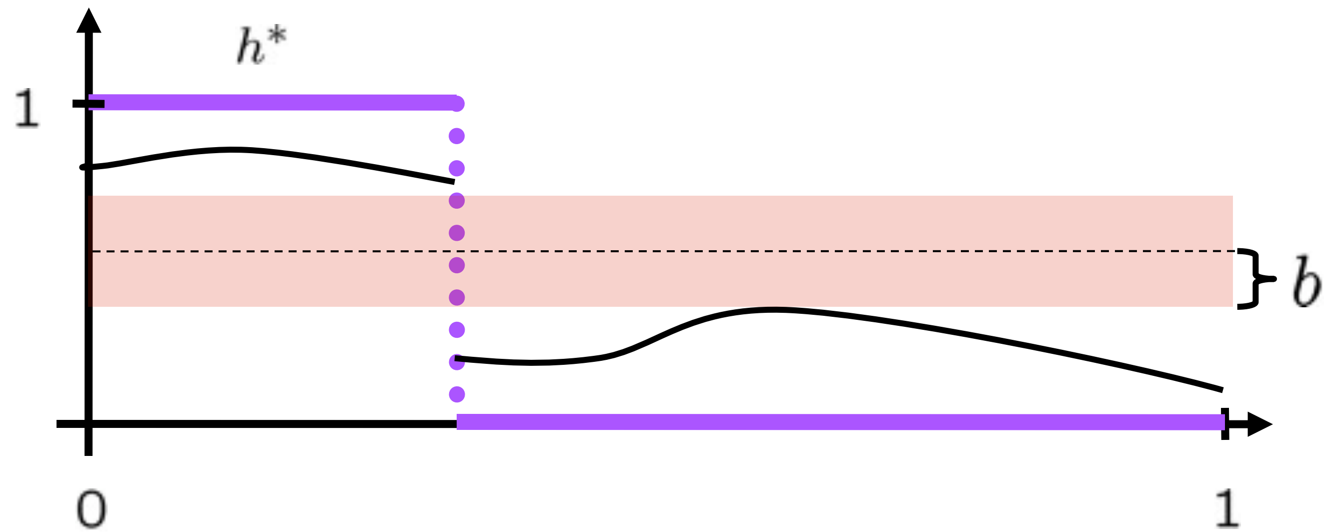
Horstein's Algorithm (incremental information gain)

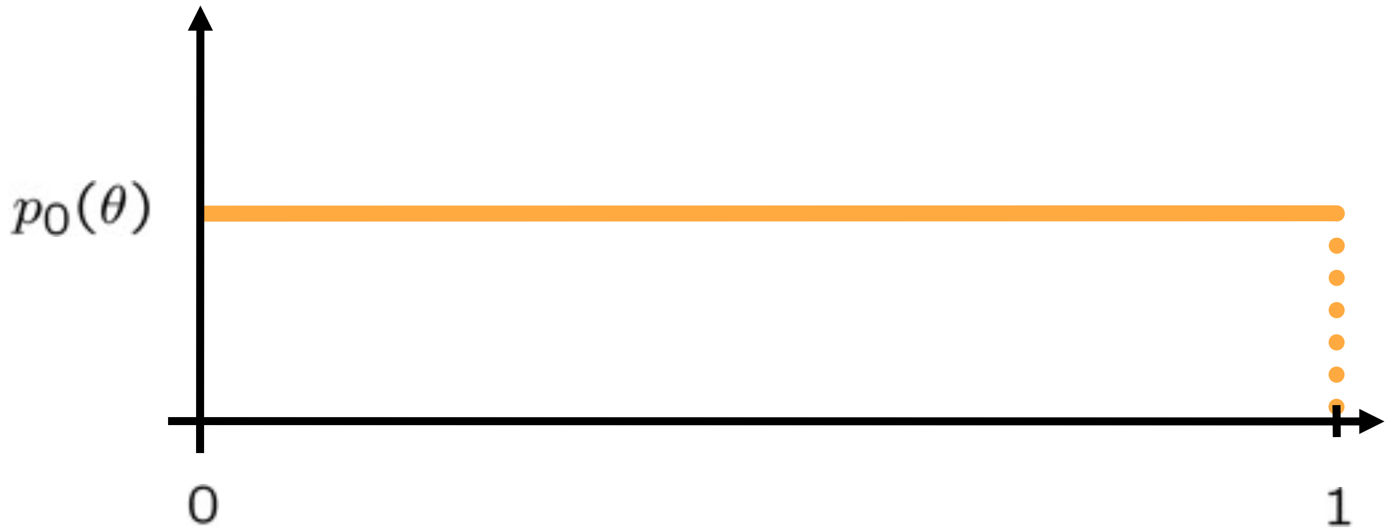


Horstein's Algorithm (incremental information gain)

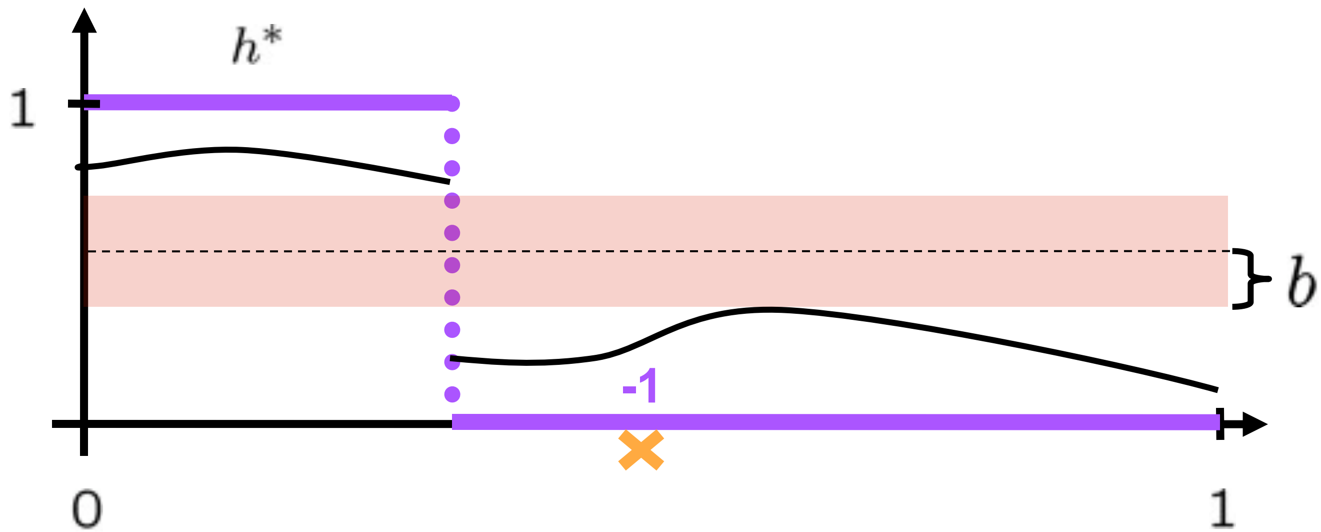


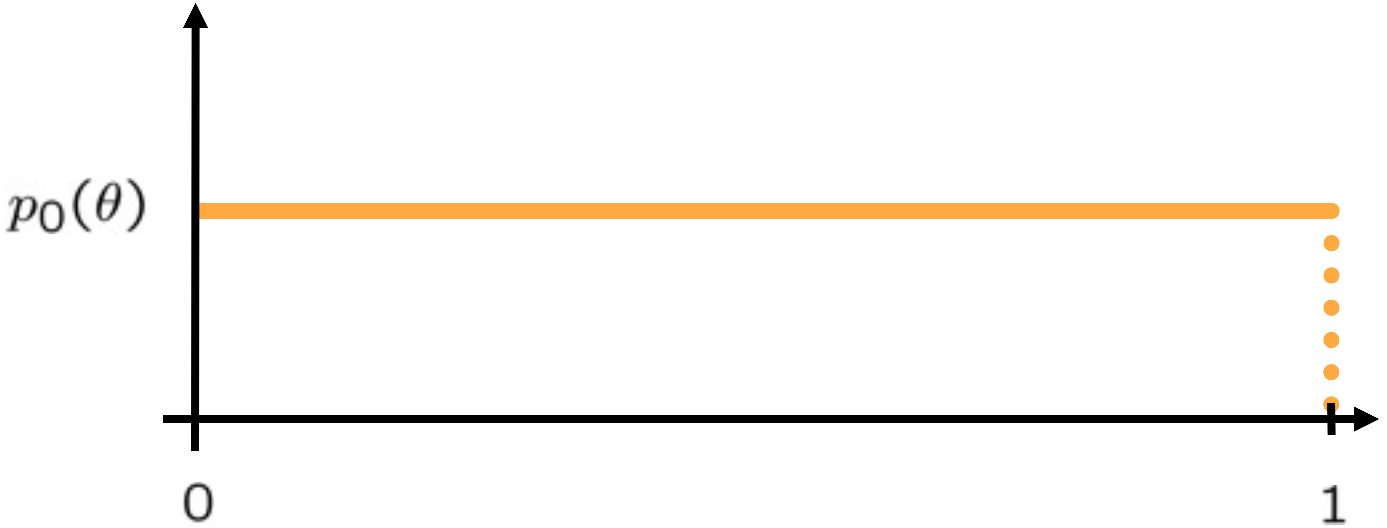
Horstein's Algorithm (incremental information gain)



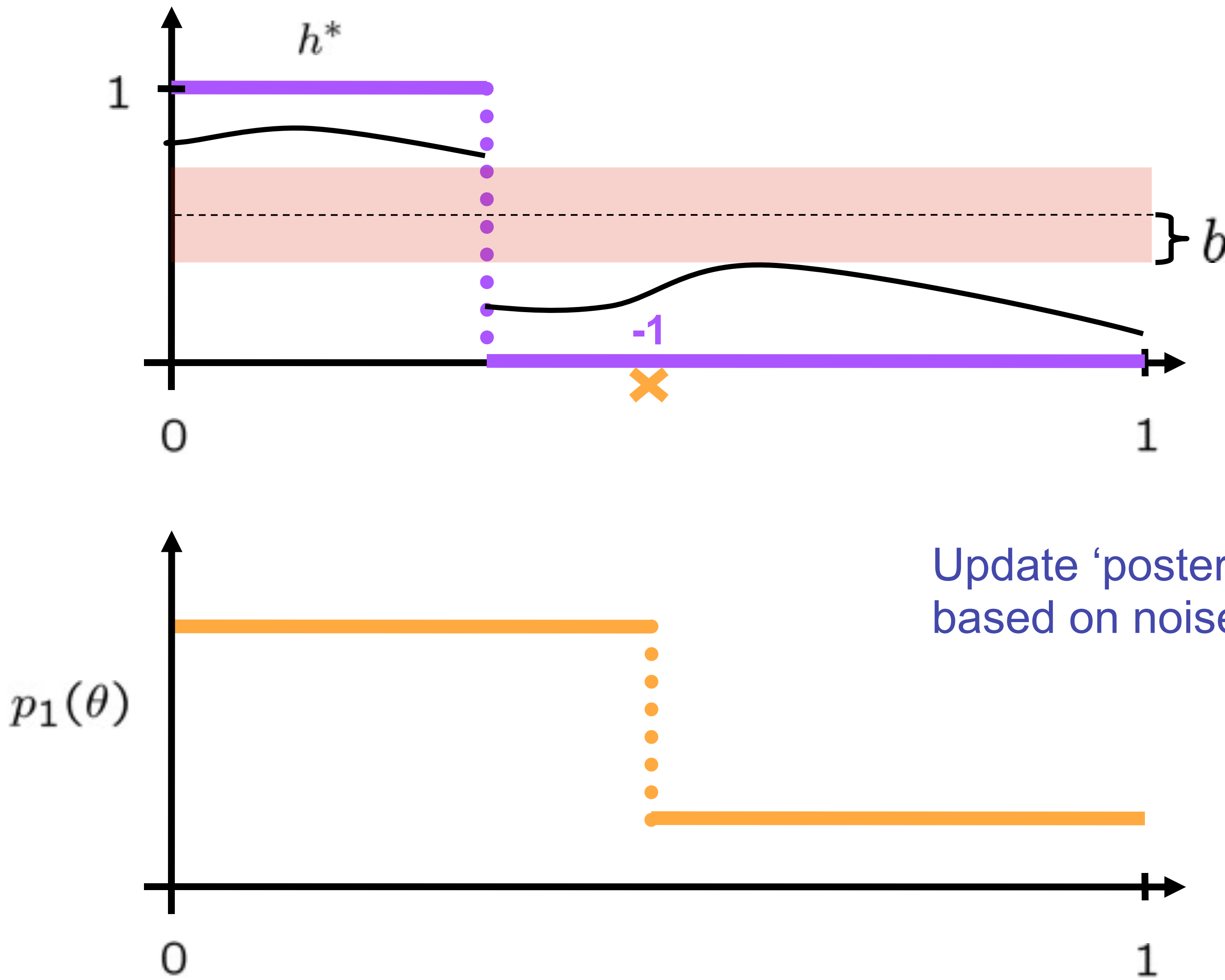


Horstein's Algorithm (incremental information gain)



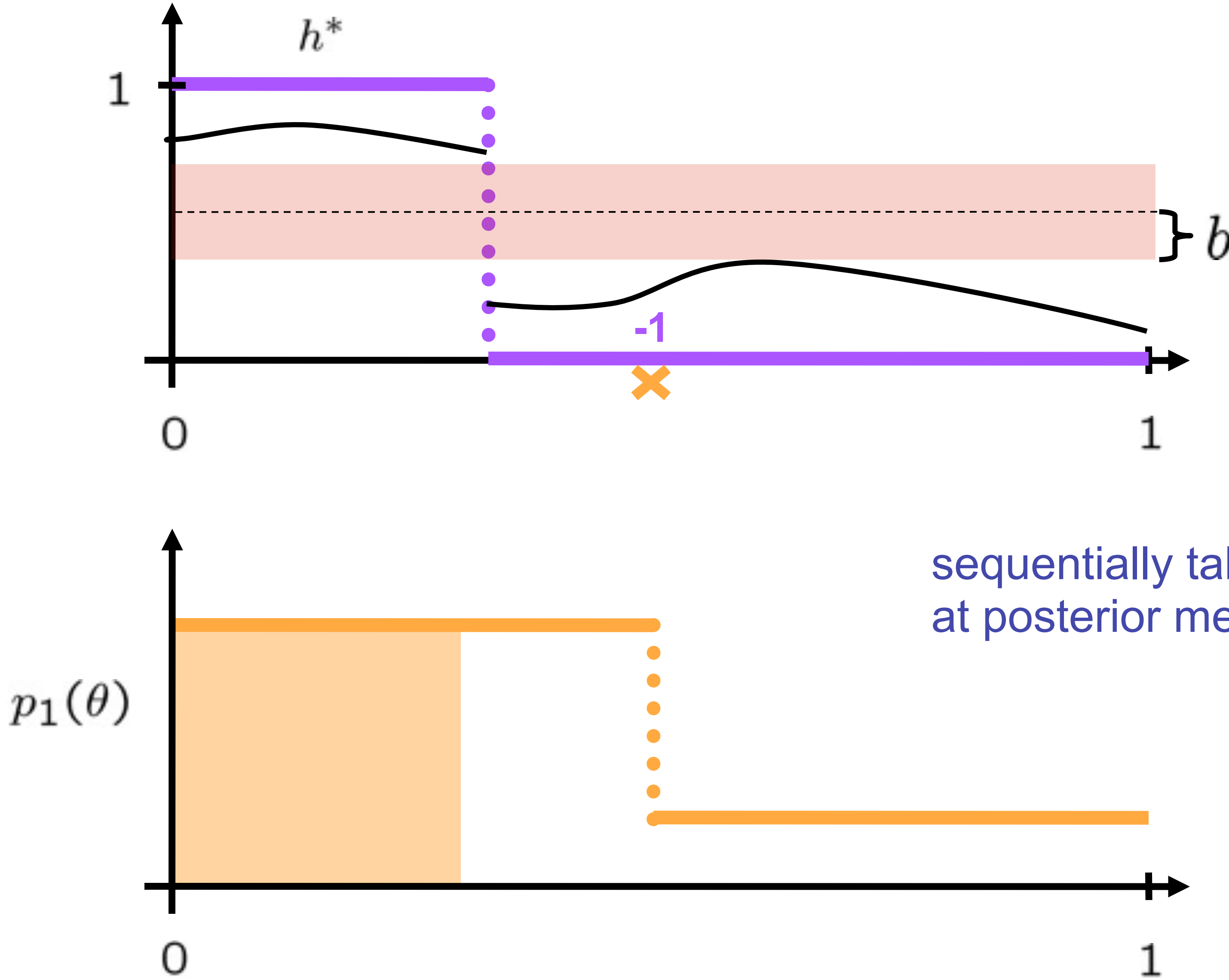


Horstein's Algorithm (incremental information gain)



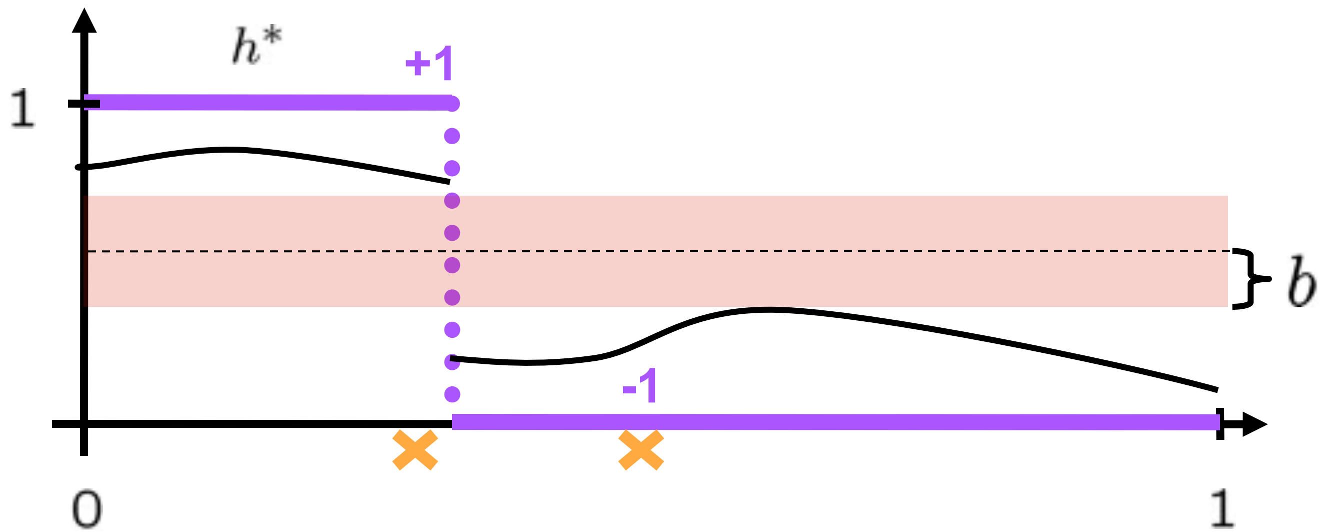
Update 'posterior' density based on noise bound b

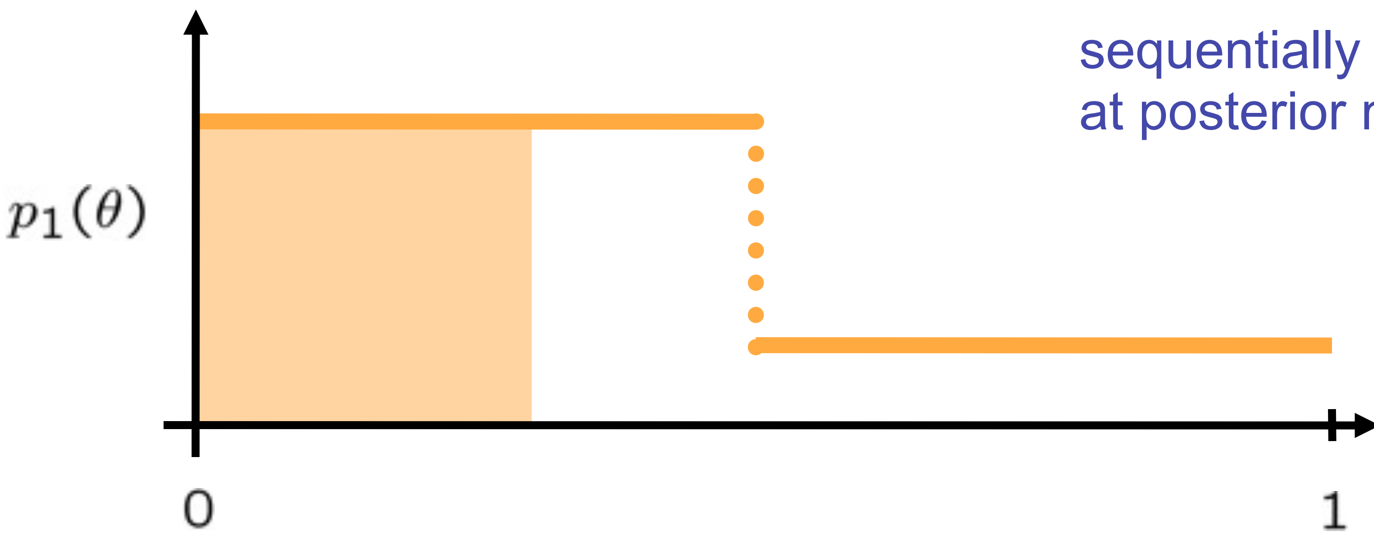
Horstein's Algorithm (incremental information gain)



sequentially take samples at posterior median

Horstein's Algorithm (incremental information gain)

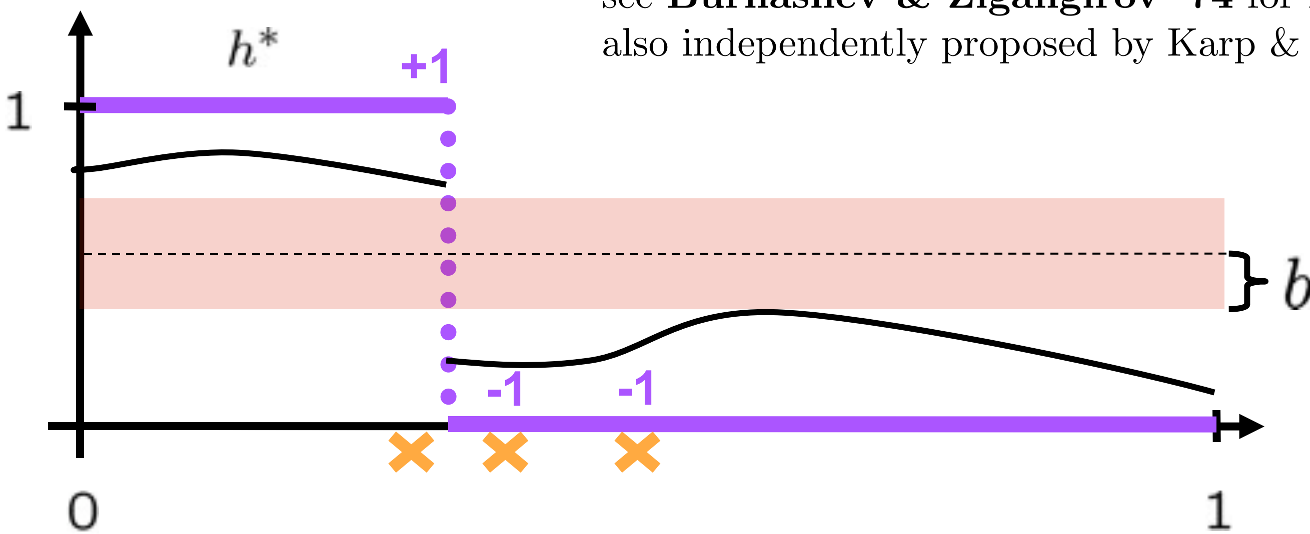


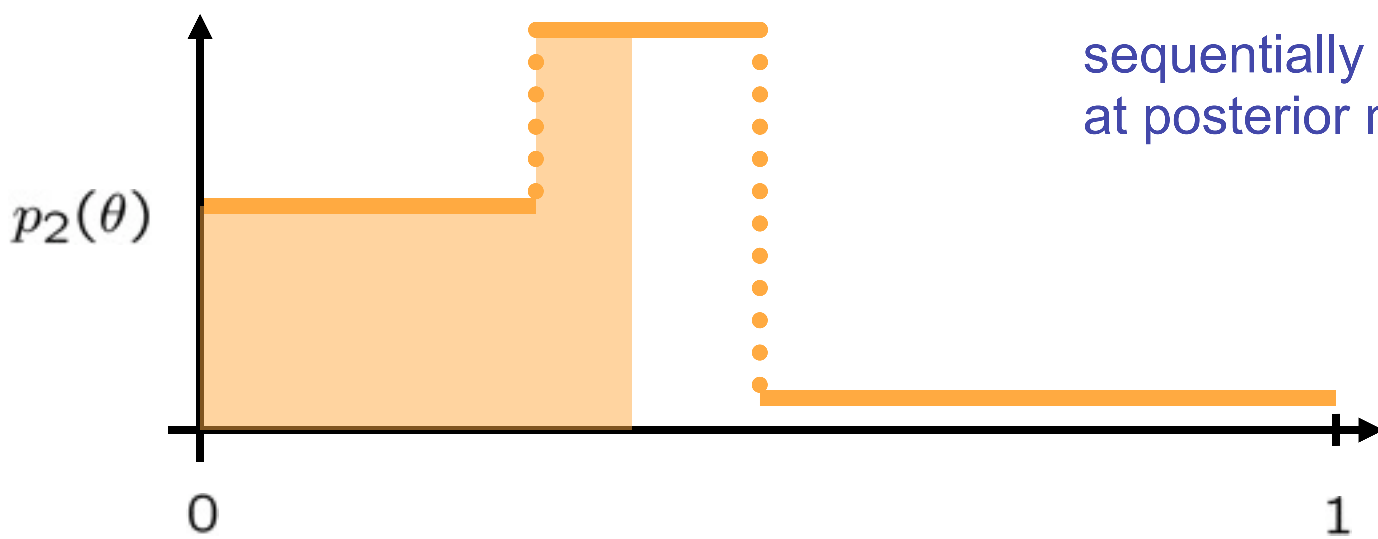


sequentially take samples
at posterior median

Horstein's Algorithm (incremental information gain)

see **Burnashev & Zigangirov '74** for rigorous analysis; also independently proposed by Karp & Kleinberg '07

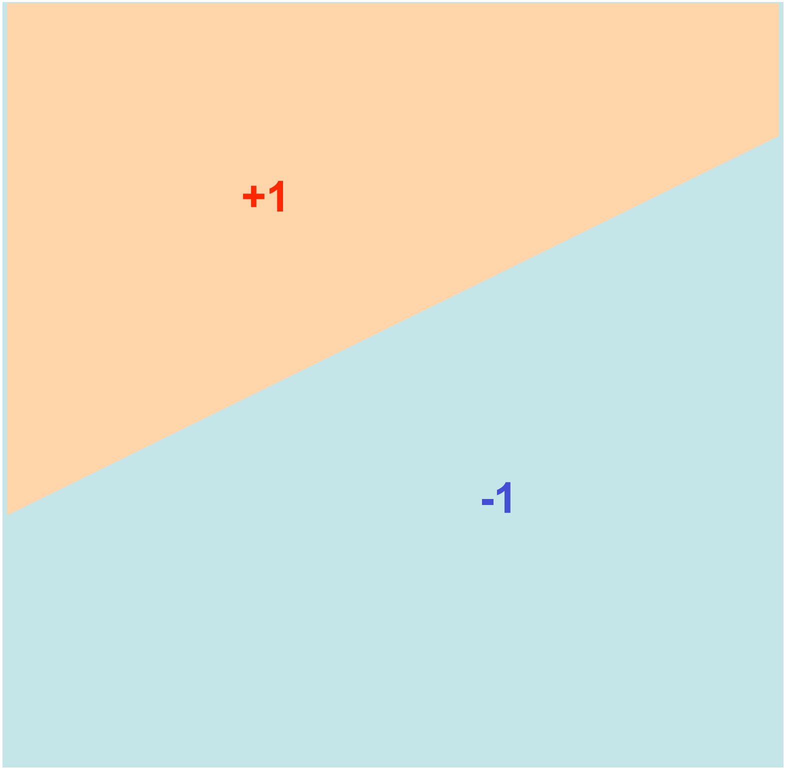




sequentially take samples at posterior median

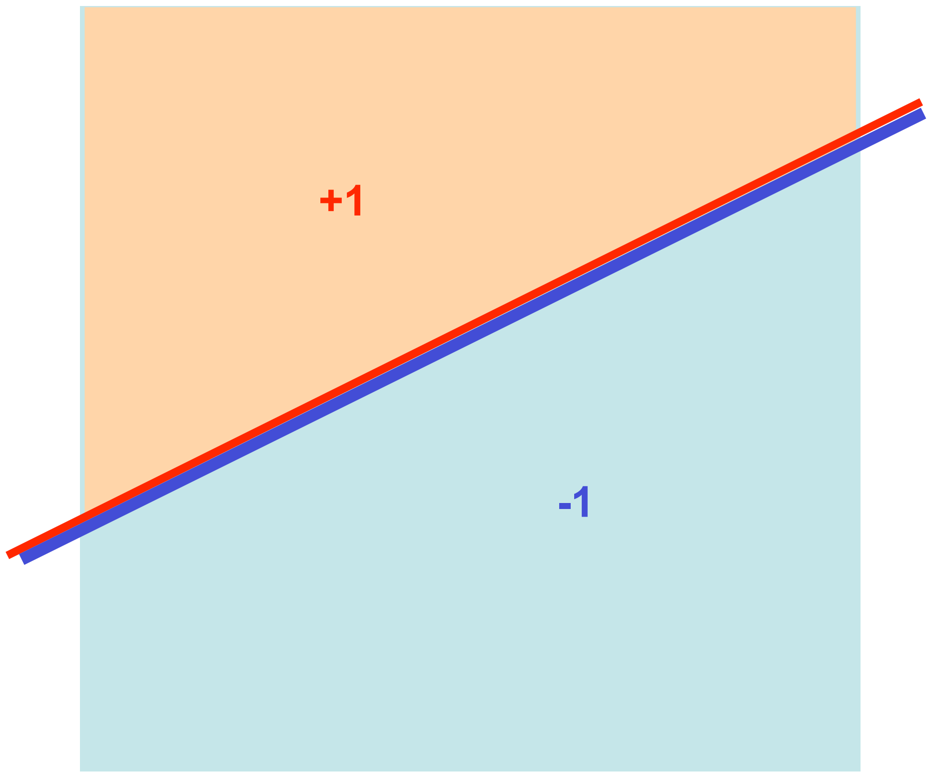
Halfspaces: Canonical Model for Multi-Dim Problems

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$$



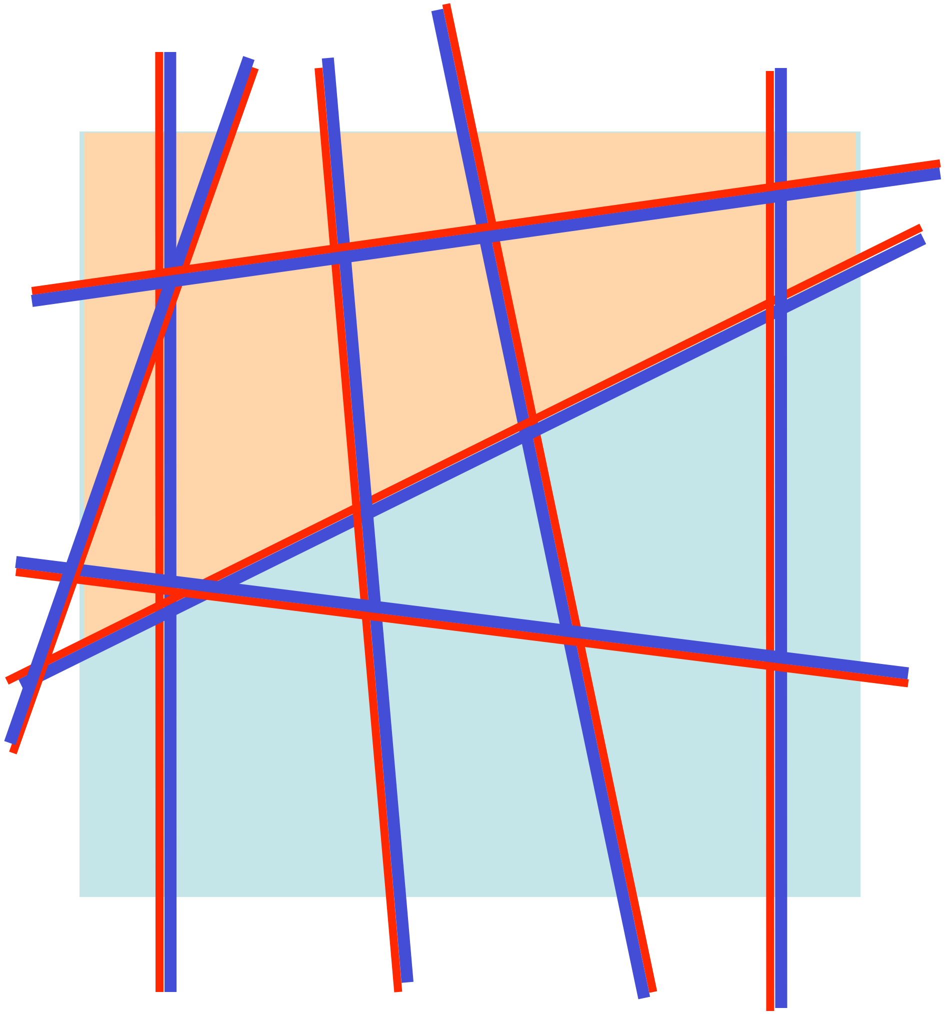
Halfspaces: Canonical Model for Multi-Dim Problems

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$$



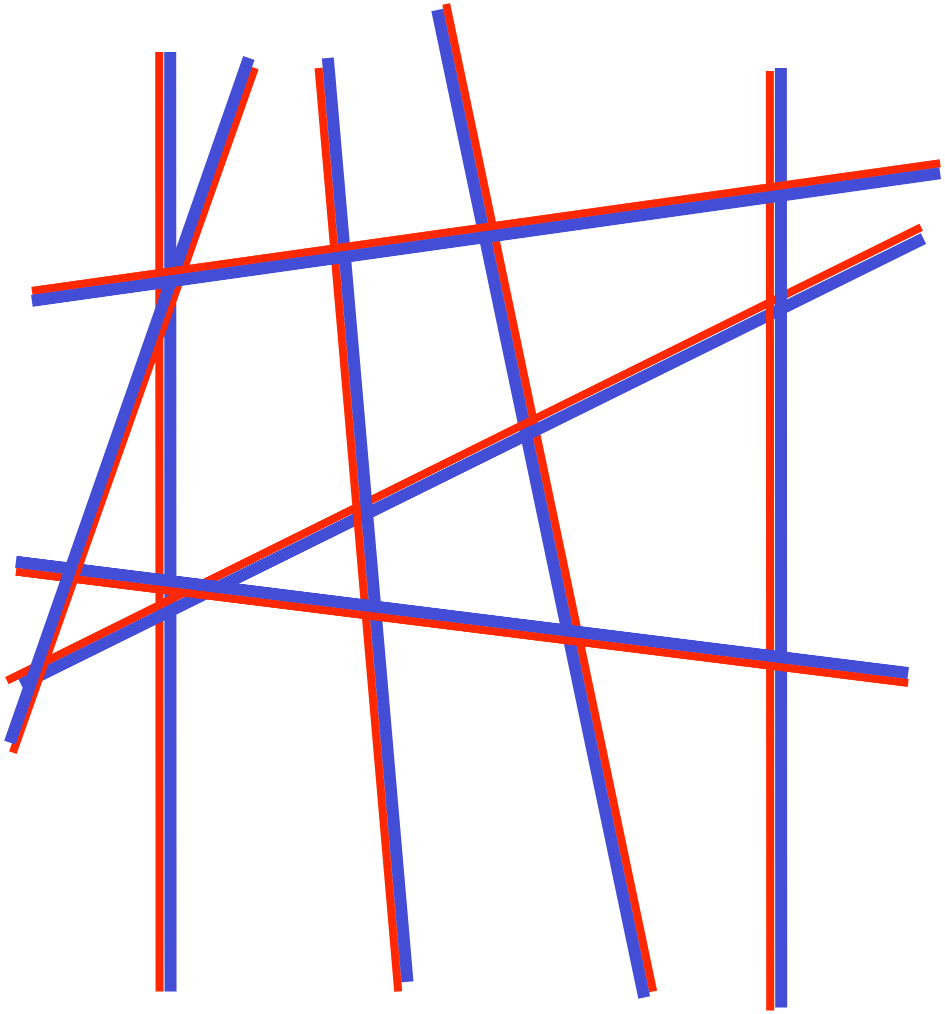
Halfspaces: Canonical Model for Multi-Dim Problems

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$$



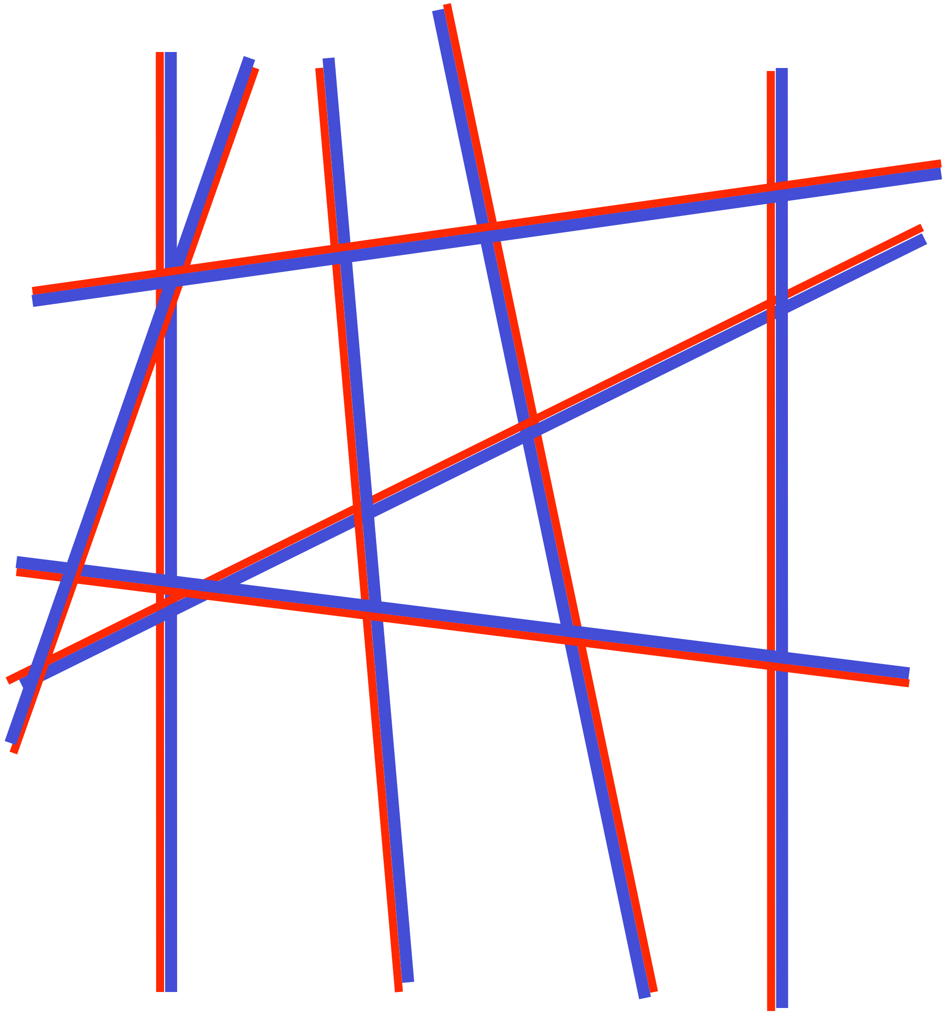
Halfspaces: Canonical Model for Multi-Dim Problems

$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$$



Halfspaces: Canonical Model for Multi-Dim Problems

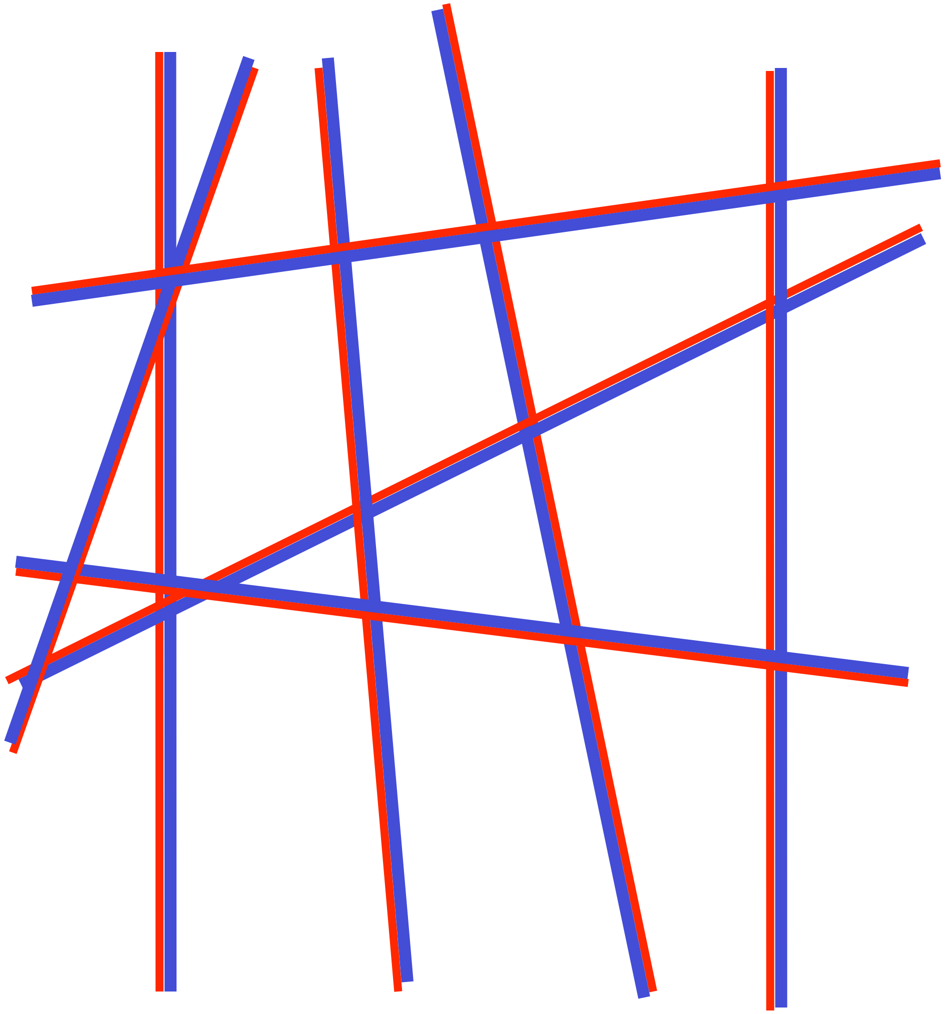
$$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$$



How to select queries?

Halfspaces: Canonical Model for Multi-Dim Problems

$\mathcal{X} = \mathbb{R}^d \quad \mathcal{H} = \{\text{finite number of halfspaces}\}$



How to select queries?

What is query complexity? Is it $\log_2 |\mathcal{H}|$?

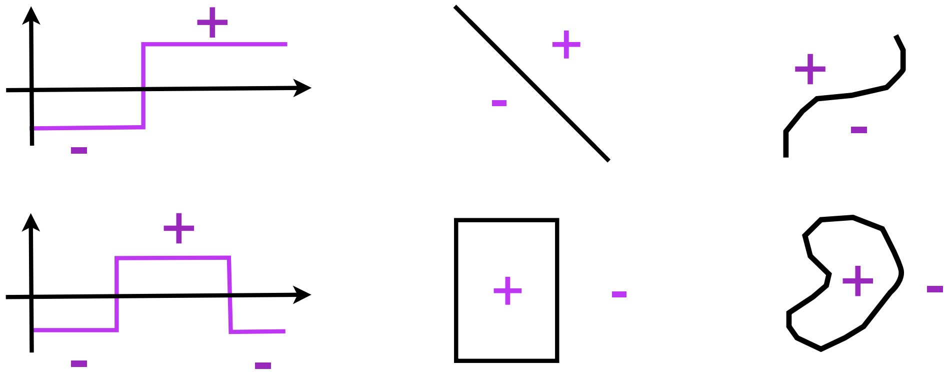
Incremental Information-Gain for Classification

$\mathcal{X}$:= feature or *query space*

$\mathcal{Y} := \{-1, +1\}$

$\mathcal{H}$:= *hypothesis space*

$\forall h \in H, \;\; h : \mathcal{X} \rightarrow \mathcal{Y}$



Assume labels y are deterministically related to features x, i.e., "noiseless"

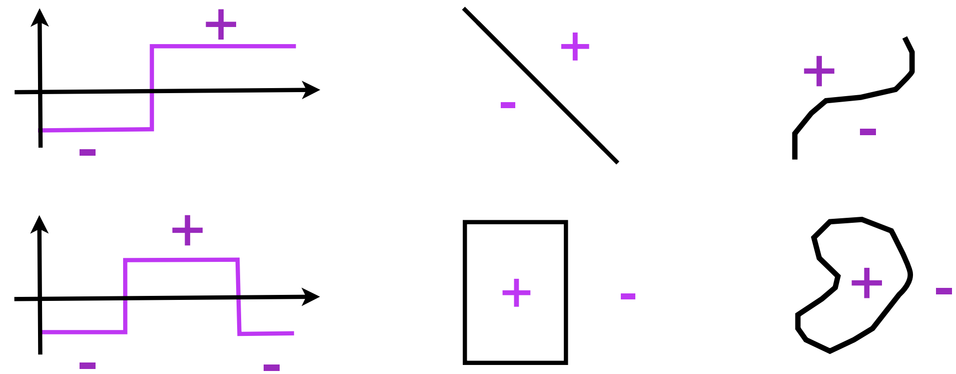
Incremental Information-Gain for Classification

$\mathcal{X}$:= feature or *query space*

$\mathcal{Y} := \{-1, +1\}$

$\mathcal{H}$:= *hypothesis space*

$\forall h \in H, \;\; h : \mathcal{X} \to \mathcal{Y}$



Assume labels y are deterministically related to features x, i.e., "noiseless"

Generalized Binary Search (GBS)

initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

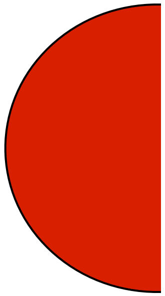
1) Select $x_n = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$

2) Query with x_n to obtain response $y_n = h^*(x_n)$

3) Set $\mathcal{H}_{n+1} = \{h \in \mathcal{H}_n : h(x_n) = y_n\}$, $n = n + 1$

$\mathcal{H}$

hypothesis space



“Is the person
wearing a hat ?”

“Is the person wearing a hat ?”

“Does the person have blue eyes ?”

“Is the person wearing a hat ?”

“Does the person have blue eyes ?”

GBS can be quite effective if responses are reliable

"Is the person wearing a hat ?"

"Does the person have blue eyes ?"

GBS can be quite effective if responses are reliable

Generalized Binary Search with Noise

Generalized Binary Search (GBS)

initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

1) Select $x_n = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$

2) Query with x_n to obtain response $y_n = h^*(x_n)$

3) Set $\mathcal{H}_{n+1} = \{h \in \mathcal{H}_n : h(x_n) = y_n\}$, $n = n + 1$

Generalized Binary Search with Noise

Generalized Binary Search (GBS)

initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

1) Select $x_n = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$

2) Query with x_n to obtain response $y_n = h^*(x_n)$

3) Set $\mathcal{H}_{n+1} = \{h \in \mathcal{H}_n : h(x_n) = y_n\}$, $n = n + 1$

Suppose that the binary response $y \in \{-1, 1\}$ to query $x \in \mathcal{X}$ is an independent realization of the random variable Y satisfying $\mathbb{P}(Y = h^*(x)) > \mathbb{P}(Y = -h^*(x))$, where $h^* \in \mathcal{H}$ is fixed but unknown (i.e., the response is only probably correct)

Generalized Binary Search with Noise

Generalized Binary Search (GBS)

initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

1) Select $x_n = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$

2) Query with x_n to obtain response $y_n = h^*(x_n)$

3) Set $\mathcal{H}_{n+1} = \{h \in \mathcal{H}_n : h(x_n) = y_n\}$, $n = n + 1$

Suppose that the binary response $y \in \{-1, 1\}$ to query $x \in \mathcal{X}$ is an independent realization of the random variable Y satisfying $\mathbb{P}(Y = h^*(x)) > \mathbb{P}(Y = -h^*(x))$, where $h^* \in \mathcal{H}$ is fixed but unknown (i.e., the response is only probably correct)

The *noise bound* is defined as $\alpha := \sup_{x \in \mathcal{X}} \mathbb{P}(Y \neq h^*(x))$

Generalized Binary Search with Noise

Noise-tolerant GBS

initialize: p_0 uniform over $\mathcal{H}$ and $\alpha < \beta < 1/2$.

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

1) $x_n = \arg\min_{x\in\mathcal{X}} |\sum_{h\in\mathcal{H}} p_n(h)h(x)|$

2) Obtain noisy response y_n

3) Bayes update: $\forall\, h$

$$p_{n+1}(h) \ \propto \ p_n(h) \times \begin{cases} 1-\beta & , h(x_n) = y_n \\ \beta & , h(x_n) \neq y_n \end{cases}$$

hypothesis selected at each step:

$\widehat{h}_n := \arg\max_{h\in H} p_n(h)$

Suppose that the binary response $y \in \{-1, 1\}$ to query $x \in \mathcal{X}$ is an independent realization of the random variable Y satisfying $\mathbb{P}(Y = h^*(x)) > \mathbb{P}(Y = -h^*(x))$, where $h^* \in \mathcal{H}$ is fixed but unknown (i.e., the response is only probably correct)

The *noise bound* is defined as $\alpha := \sup_{x\in\mathcal{X}} \mathbb{P}(Y \neq h^*(x))$

Noise-tolerant GBS is a generalized version of Horstein's algorithm

When is Noisy GBS Information-Theoretically Optimal?

Theorem 1 Let $\mathbb{P}$ denotes the underlying probability measure (governing errors and randomization). Under mild conditions, noise-tolerant GBS generates a sequence of hypotheses satisfying

$$\mathbb{P}(\widehat{h}_n \neq h^*) \;\leq\; |\mathcal{H}|\,(1-\lambda)^n \;\leq\; |\mathcal{H}|\,e^{-\lambda n} \;\;,\; n = 0, 1, \ldots$$

with exponential constant $\lambda = \frac{1}{2}\left(1 - \frac{\beta(1-\alpha)}{1-\beta} - \frac{\alpha(1-\beta)}{\beta}\right)$

N

When is Noisy GBS Information-Theoretically Optimal?

Theorem 1 Let $\mathbb{P}$ denotes the underlying probability measure (governing errors and randomization). Under mild conditions, noise-tolerant GBS generates a sequence of hypotheses satisfying

$$\mathbb{P}(\widehat{h}_n \neq h^*) \;\leq\; |\mathcal{H}|\,(1-\lambda)^n \;\leq\; |\mathcal{H}|\,e^{-\lambda n} \;\;,\;\; n = 0, 1, \ldots$$

with exponential constant $\lambda = \frac{1}{2}\left(1 - \frac{\beta(1-\alpha)}{1-\beta} - \frac{\alpha(1-\beta)}{\beta}\right)$

$\Rightarrow \;\; n = O(\log|\mathcal{H}|)$ will suffice

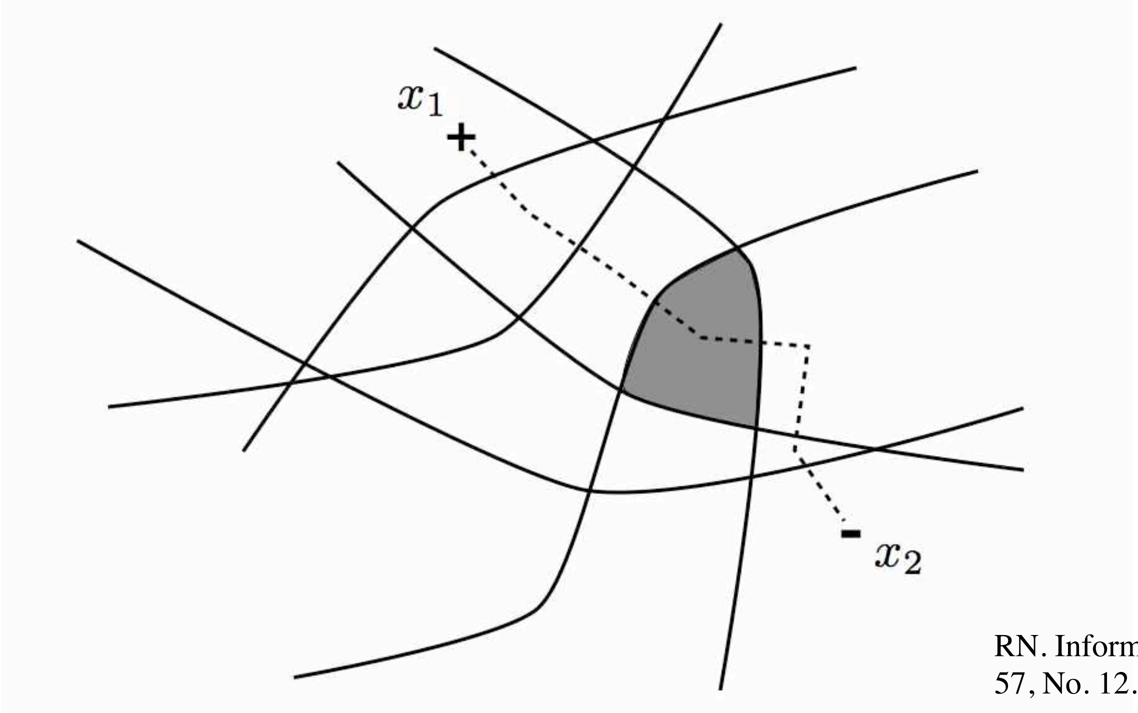
When is Noisy GBS Information-Theoretically Optimal?

Theorem 1 Let $\mathbb{P}$ denotes the underlying probability measure (governing errors and randomization). Under mild conditions, noise-tolerant GBS generates a sequence of hypotheses satisfying

$$\mathbb{P}(\widehat{h}_n \neq h^*) \;\leq\; |\mathcal{H}|\,(1-\lambda)^n \;\leq\; |\mathcal{H}|\,e^{-\lambda n} \quad , \; n = 0, 1, \ldots$$

with exponential constant $\lambda = \frac{1}{2}\left(1 - \frac{\beta(1-\alpha)}{1-\beta} - \frac{\alpha(1-\beta)}{\beta}\right)$

$\Rightarrow \;\; n = O(\log|\mathcal{H}|)$ will suffice

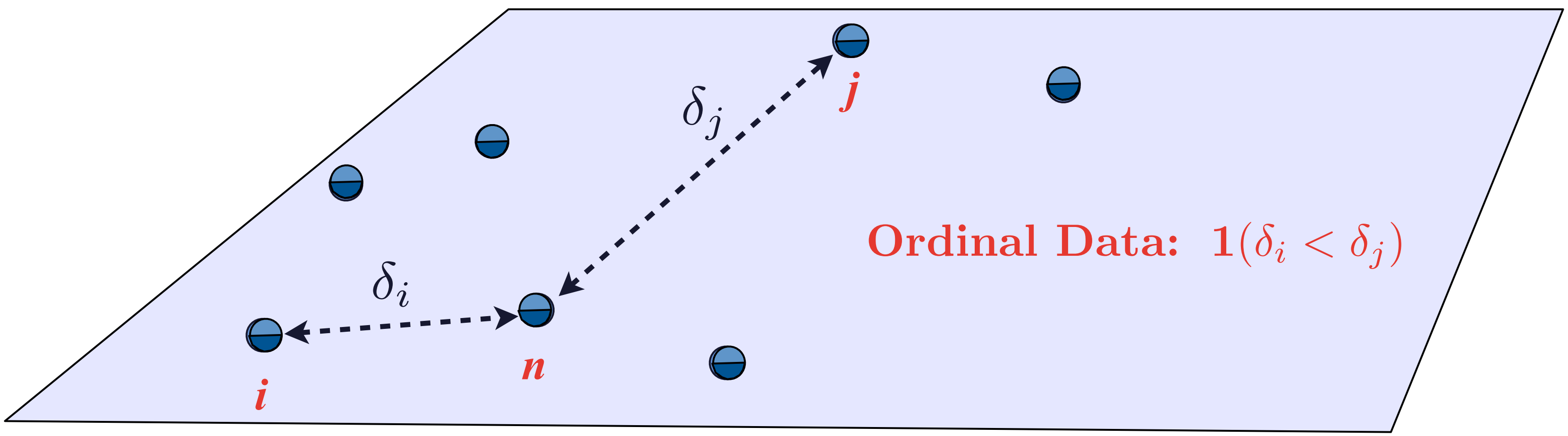


GBS isn't all that different from classic binary search

RN. Information Theory, IEEE Transactions on, Vol. 57, No. 12. (December 2011), pp. 7893-7906.

Ranking Based on Pairwise Comparisons

Ranking Problem: Consider a set of n objects $x_1, \ldots, x_n \in \mathbb{R}^d$. The locations of $x, \ldots, x_{n-1}$ are known, but location of x_n is unknown. To gather information about x_n, we can only ask questions of the form "is object x_n closer to x_i than x_j?" The goal is to rank $x_1, \ldots, x_{n-1}$ relative to distances to x_n by asking as few questions as possible.



Standard sorting methods require $n \log n$ comparisons, but this can be prohibitive when n is large, especially since it is often humans who are judging the comparisons (e.g., database search).

However, 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.


AVENTINUS
WOODCHUCK
DRAFT CIDER


Bartender: “What beer would you like?”

Bartender: “What beer would you like?”

Natasha: “Hmm... actually I’m more of wine drinker”

Bartender: “What beer would you like?”

Natasha: “Hmm... actually I’m more of wine drinker”

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

Bartender: “What beer would you like?”

Natasha: “Hmm... actually I’m more of wine drinker”

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

Natasha: “B”

Bartender: “What beer would you like?”

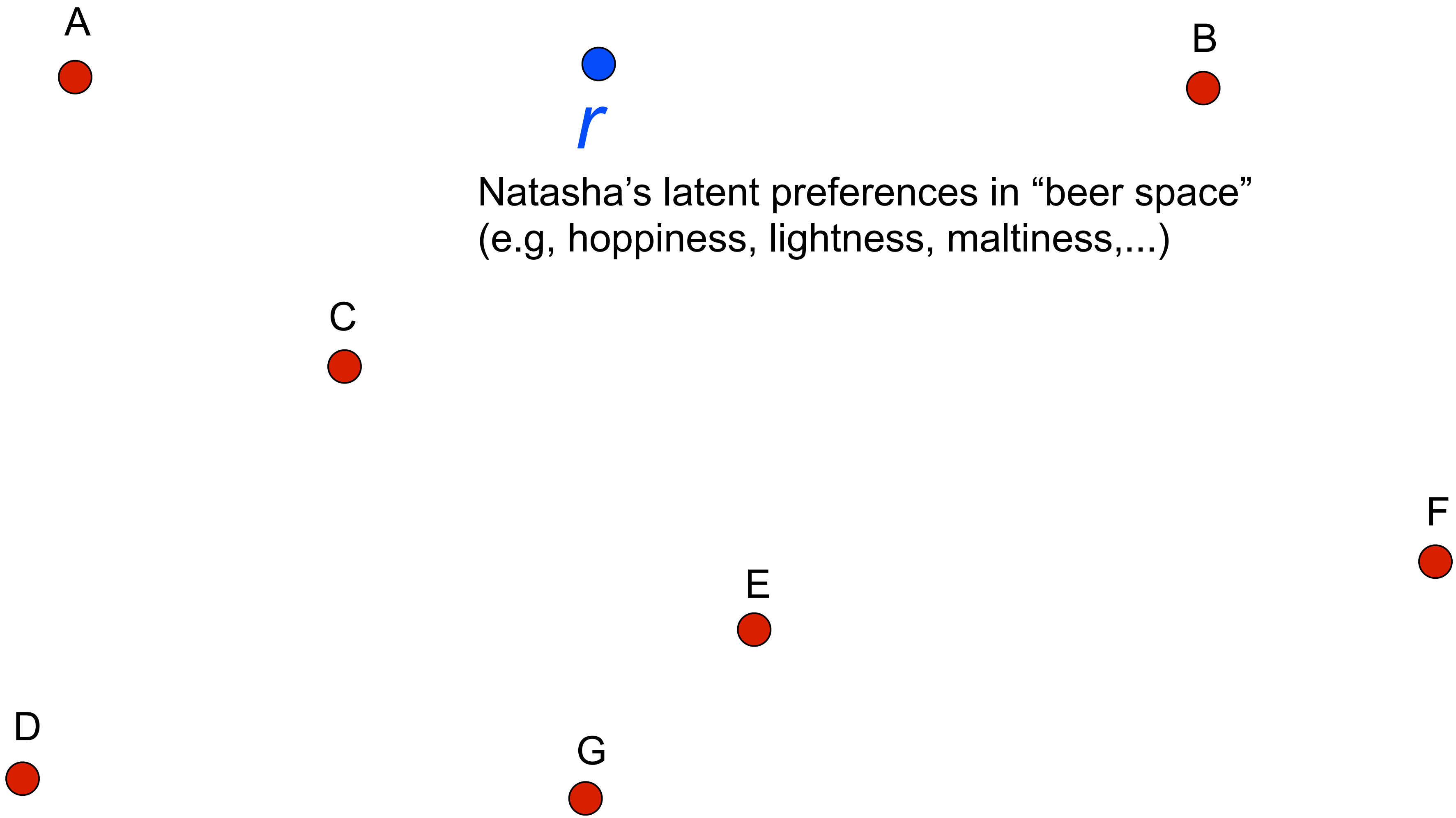
Natasha: “Hmm... actually I’m more of wine drinker”

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

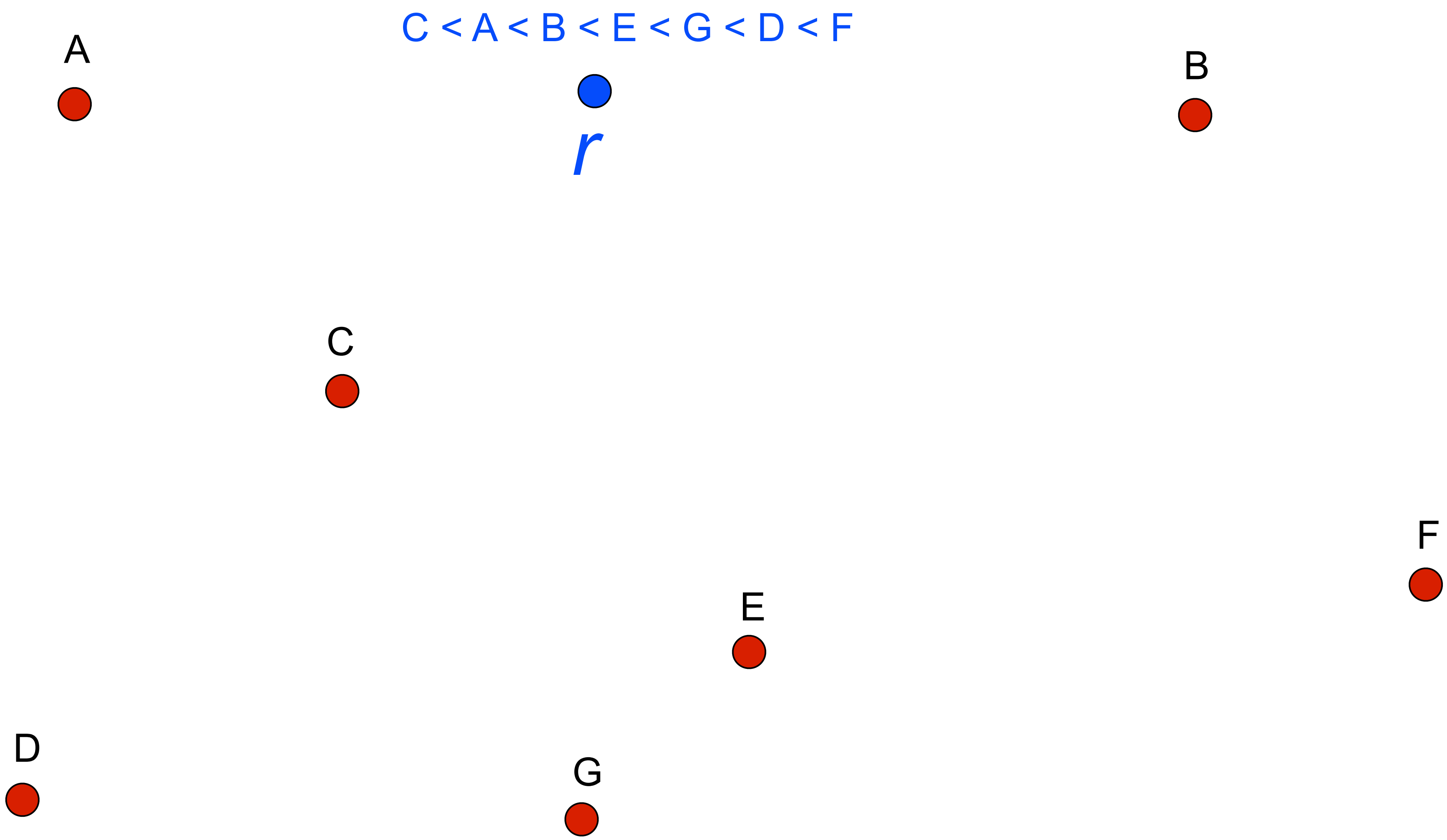
Natasha: “B”

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

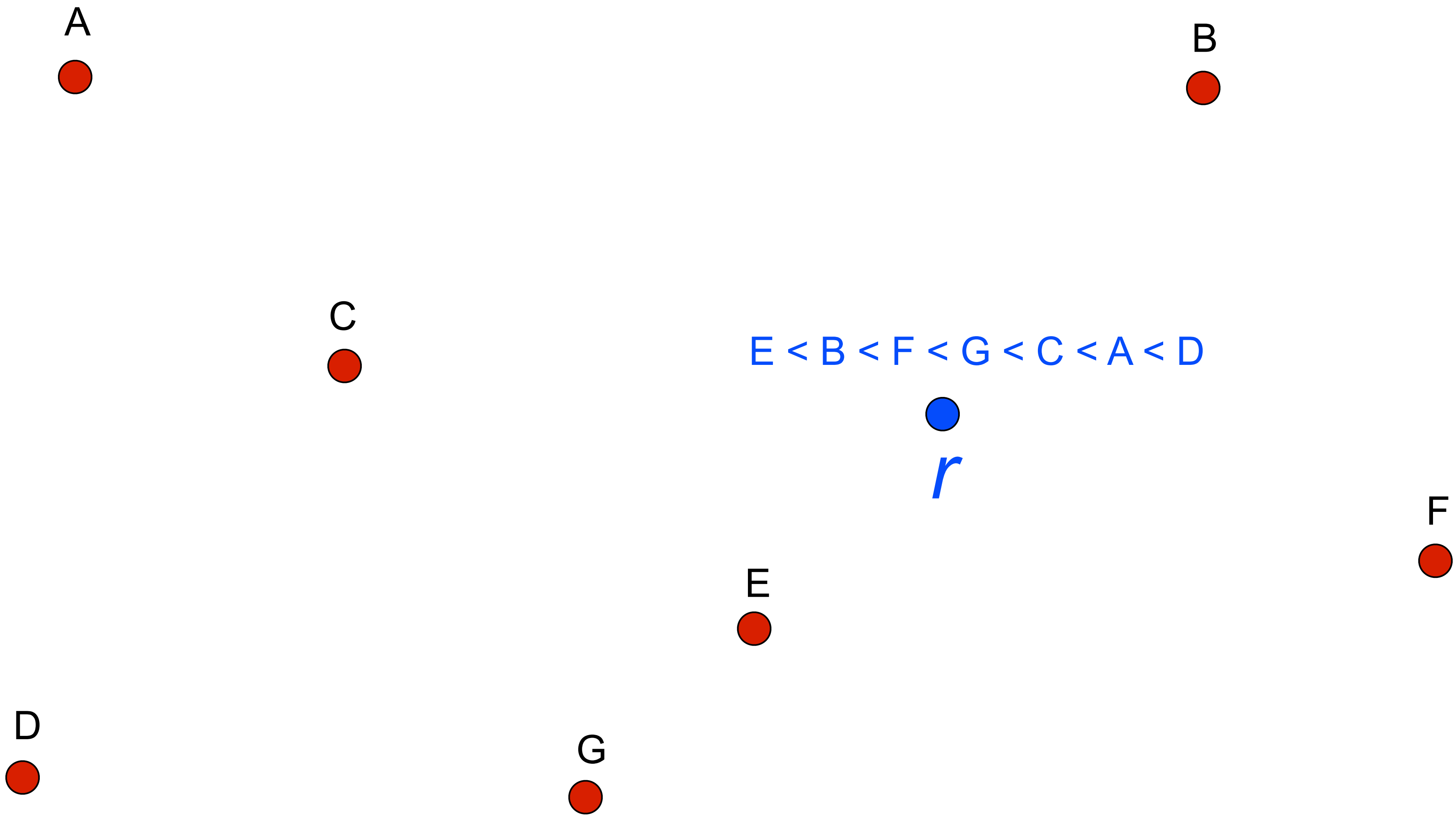
Ranking Relative to Distance



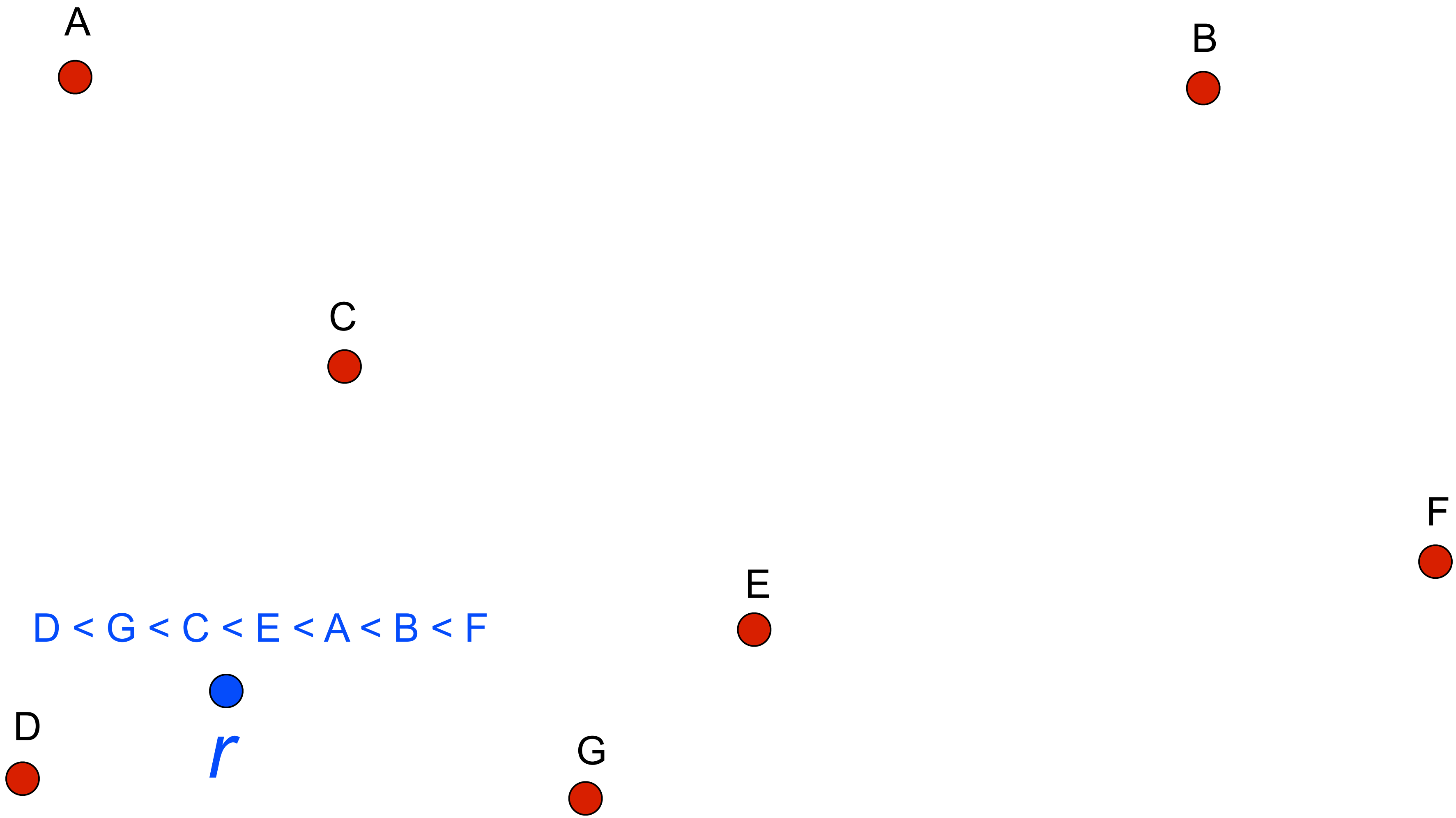
Ranking Relative to Distance



Ranking Relative to Distance



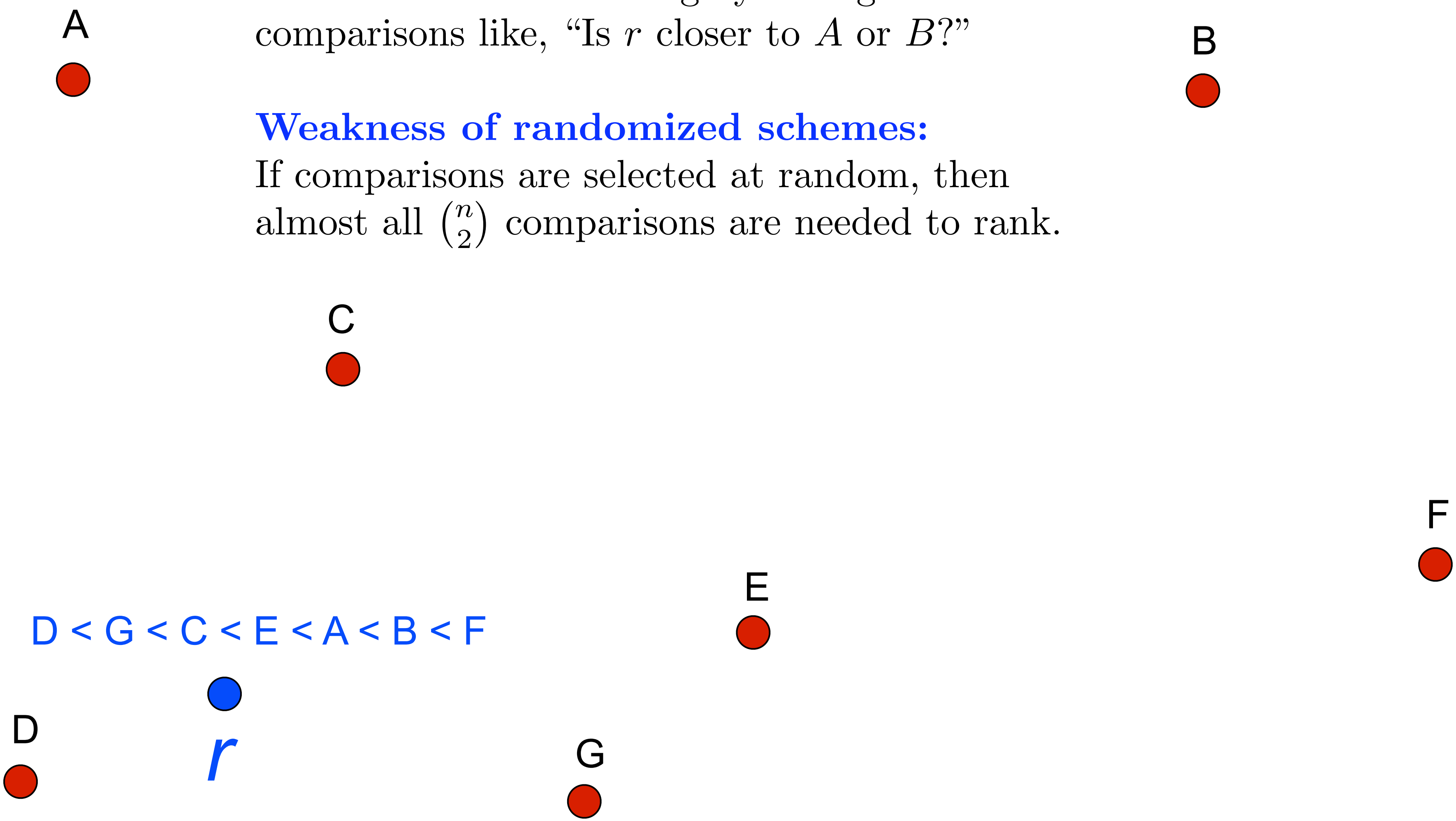
Ranking Relative to Distance



Ranking Relative to Distance

Goal: Determine ranking by asking comparisons like, "Is r closer to A or B?"

Weakness of randomized schemes:
If comparisons are selected at random, then almost all $\binom{n}{2}$ comparisons are needed to rank.



Ranking Relative to Distance

Goal: Determine ranking by asking comparisons like, "Is r closer to A or B?"

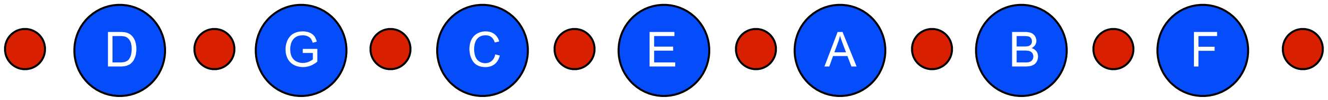
Weakness of randomized schemes:
If comparisons are selected at random, then almost all $\binom{n}{2}$ comparisons are needed to rank.

... but there are at most $n!$ rankings, and so in principle no more than $n \log n$ bits of information are needed.

D < G < C < E < A < B < F

Ranking with Adaptively Selected Queries

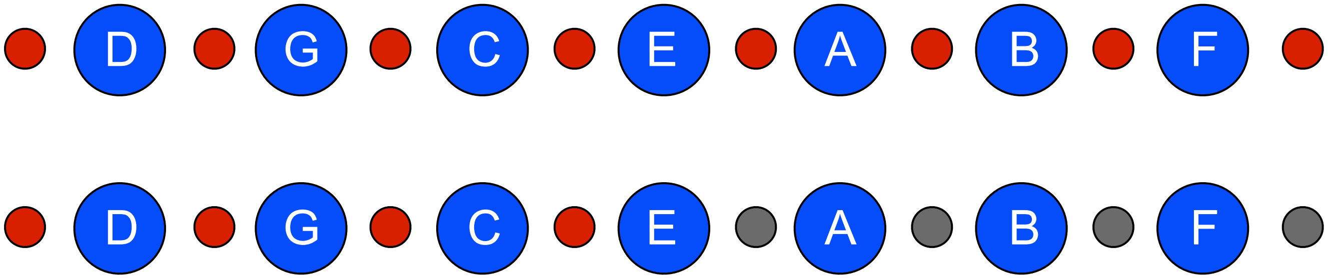
Insert H into: D < G < C < E < A < B < F



{ }

Ranking with Adaptively Selected Queries

Insert H into: D < G < C < E < A < B < F

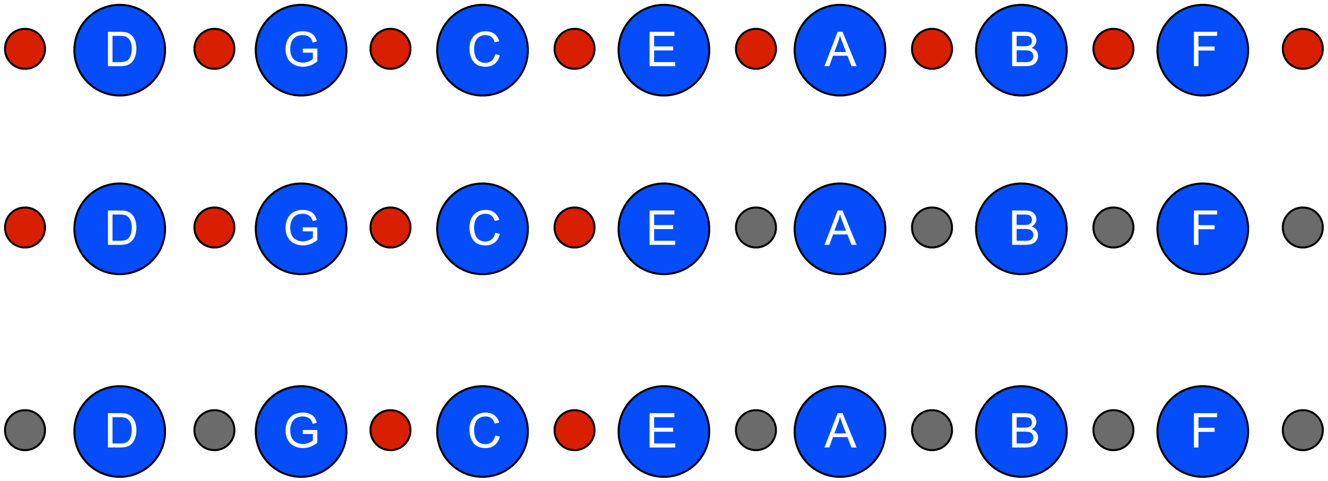


{ }

{H < E}

Ranking with Adaptively Selected Queries

Insert H into: D < G < C < E < A < B < F



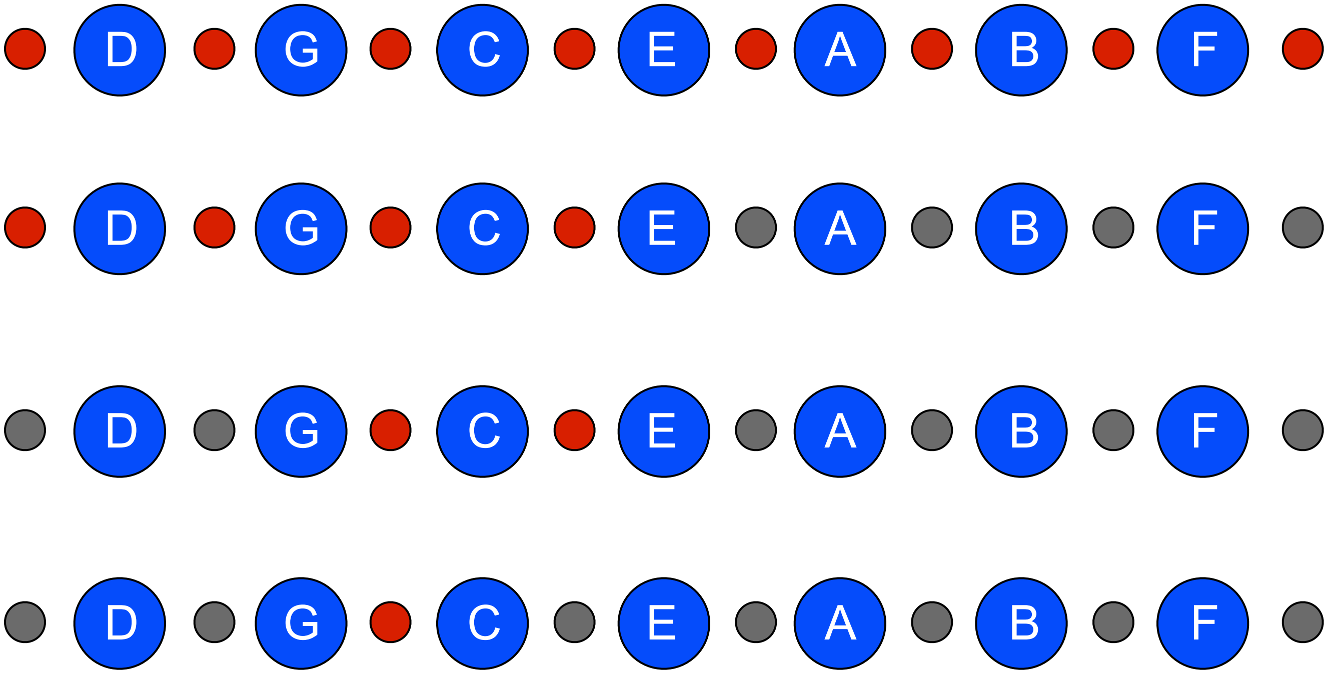
{ }

{H < E}

{H < E},{G < H}

Ranking with Adaptively Selected Queries

Insert H into: D < G < C < E < A < B < F



{ }

{H < E}

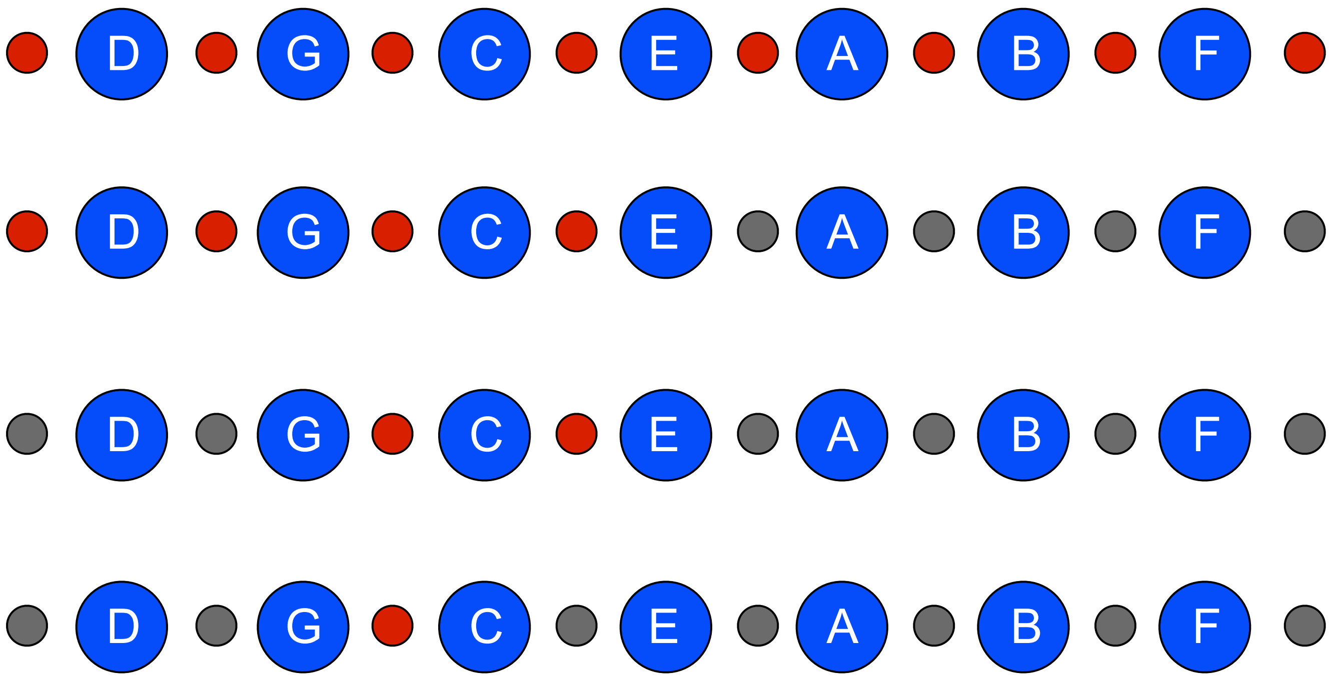
{H < E},{G < H}

{H < E},{G < H},{H < C}

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

Ranking with Adaptively Selected Queries

Insert H into: D < G < C < E < A < B < F



{ }

{H < E}

{H < E},{G < H}

{H < E},{G < H},{H < C}

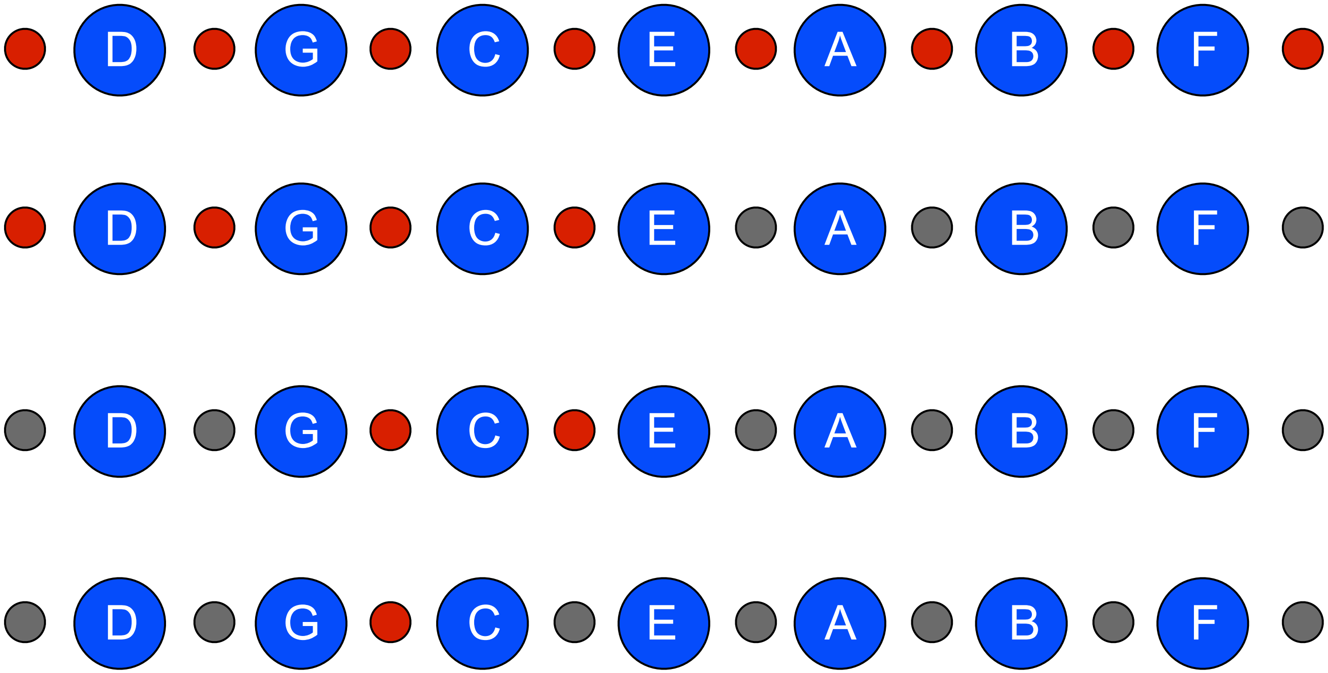
D < G < H < C < E < A < B < F

$\log_2 k$ comparisons to insert an item into a list of k objects

$\implies n \log_2 n$ comparisons to rank n objects

Ranking with Adaptively Selected Queries

Insert H into: D < G < C < E < A < B < F



{ }

{H < E}

{H < E},{G < H}

{H < E},{G < H},{H < C}

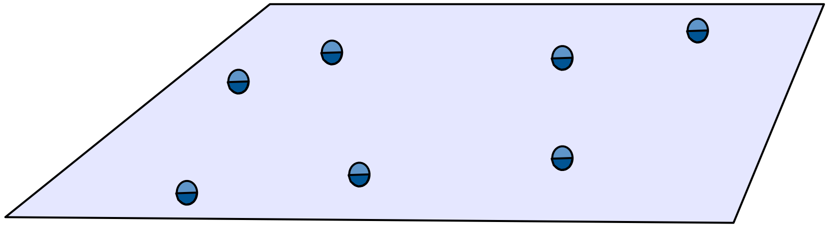
D < G < H < C < E < A < B < F

$\log_2 k$ comparisons to insert an item into a list of k objects

$\implies n \log_2 n$ comparisons to rank n objects

... but does embedding dimension d affect the sample complexity?

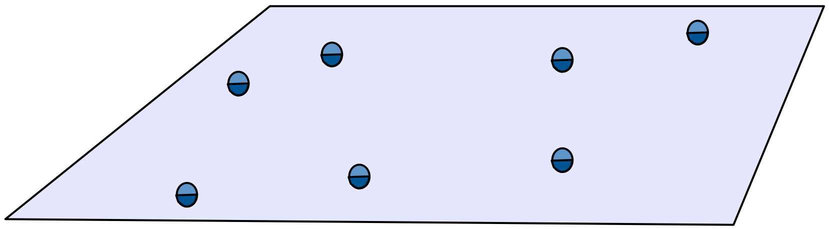
Ranking by Exploiting Low-Dimensional Geometry



In fact, there are only $O(n^{2d})$ possible rankings, and so we should only need $O(d \log n)$ bits.

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.

Ranking by Exploiting Low-Dimensional Geometry

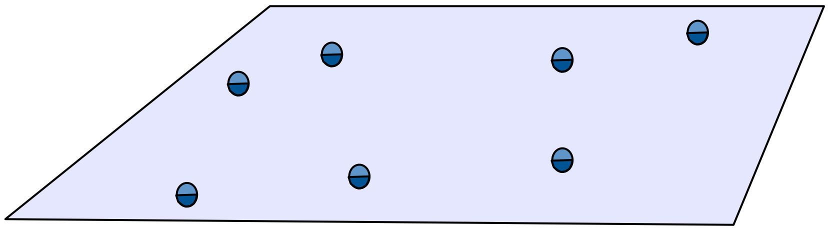


In fact, there are only $O(n^{2d})$ possible rankings, and so we should only need $O(d \log n)$ bits.

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: $q_{i,j} \equiv x_n$ **is closer to** x_i **than** x_j

Ranking by Exploiting Low-Dimensional Geometry



In fact, there are only $O(n^{2d})$ possible rankings, and so we should only need $O(d \log n)$ bits.

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: $q_{i,j} \equiv x_n$ **is closer to** x_i **than** x_j

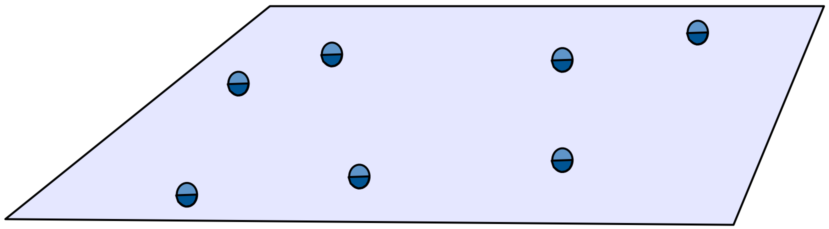
Sequential Data Selection

input: $x_1, \ldots, x_{n-1} \in \mathbb{R}^d$ and x_n at unknown position in $\mathbb{R}^d$
initialize: $x_1, \ldots, x_{n-1}$ in uniformly random order

for k=2,...,n-1
  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 $\{q_{i,j}\}_{i,j<k}$

output: ranking of $x_1, \ldots, x_{n-1}$ consistent with *all* pairwise comparisons

Ranking by Exploiting Low-Dimensional Geometry



In fact, there are only $O(n^{2d})$ possible rankings, and so we should only need $O(d \log n)$ bits.

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: $q_{i,j} \equiv x_n$ **is closer to** x_i **than** x_j

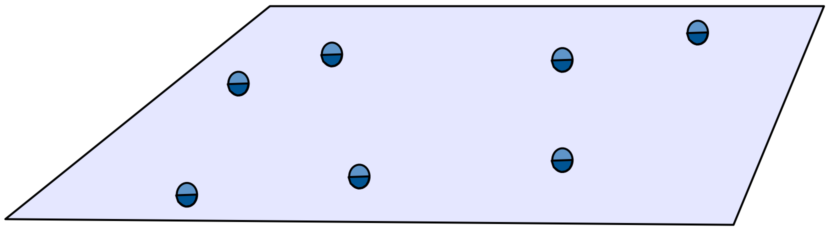
Sequential Data Selection

input: $x_1, \ldots, x_{n-1} \in \mathbb{R}^d$ and x_n at unknown position in $\mathbb{R}^d$
initialize: $x_1, \ldots, x_{n-1}$ in uniformly random order

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

output: ranking of $x_1, \ldots, x_{n-1}$ consistent with *all* pairwise comparisons

Ranking by Exploiting Low-Dimensional Geometry



In fact, there are only $O(n^{2d})$ possible rankings, and so we should only need $O(d \log n)$ bits.

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: $q_{i,j} \equiv x_n$ **is closer to** x_i **than** x_j

Sequential Data Selection

input: $x_1, \ldots, x_{n-1} \in \mathbb{R}^d$ and x_n at unknown position in $\mathbb{R}^d$
initialize: $x_1, \ldots, x_{n-1}$ in uniformly random order

for k=2,...,n-1
  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 $\{q_{i,j}\}_{i,j<k}$

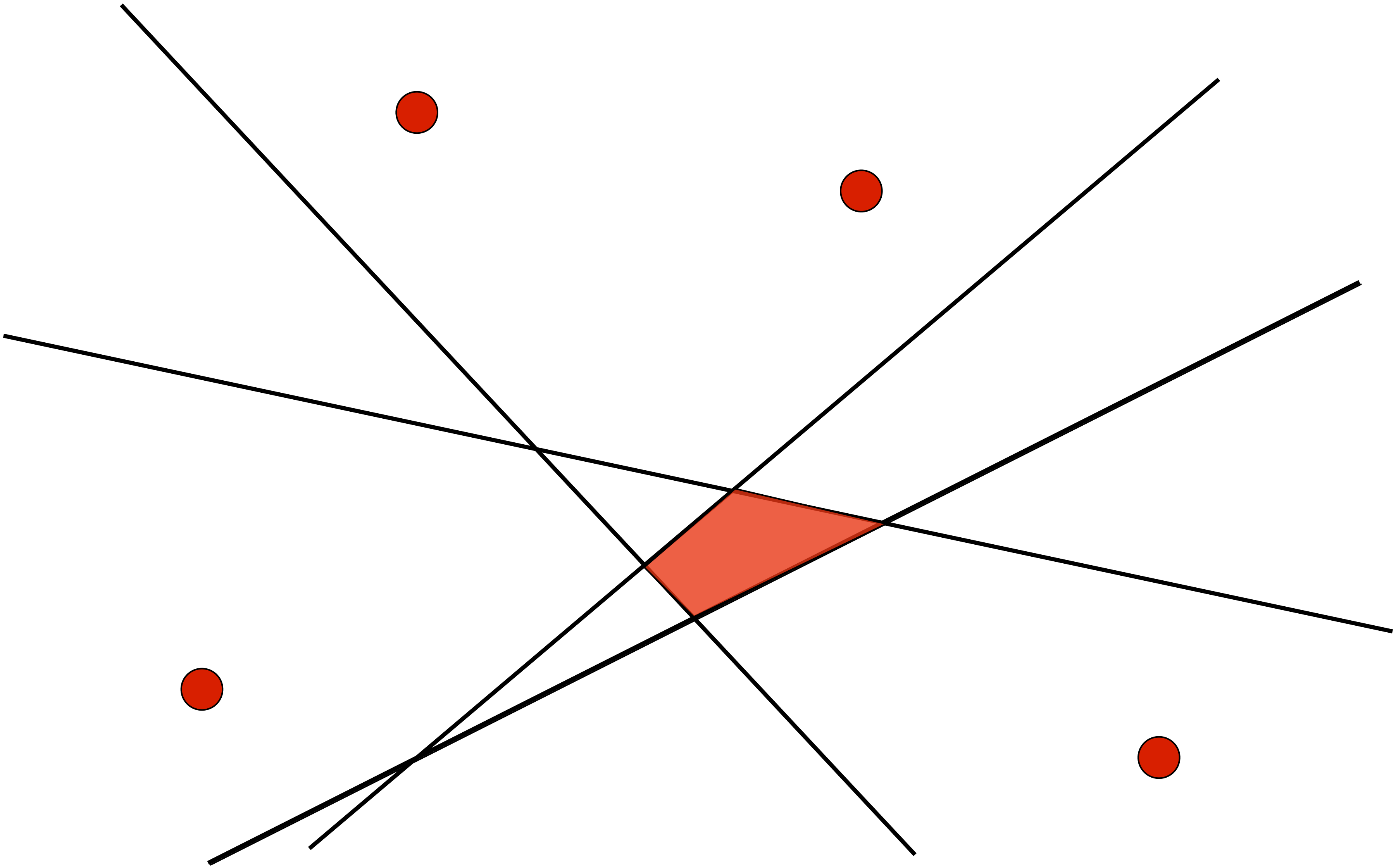
simple linear program

positive info-gain

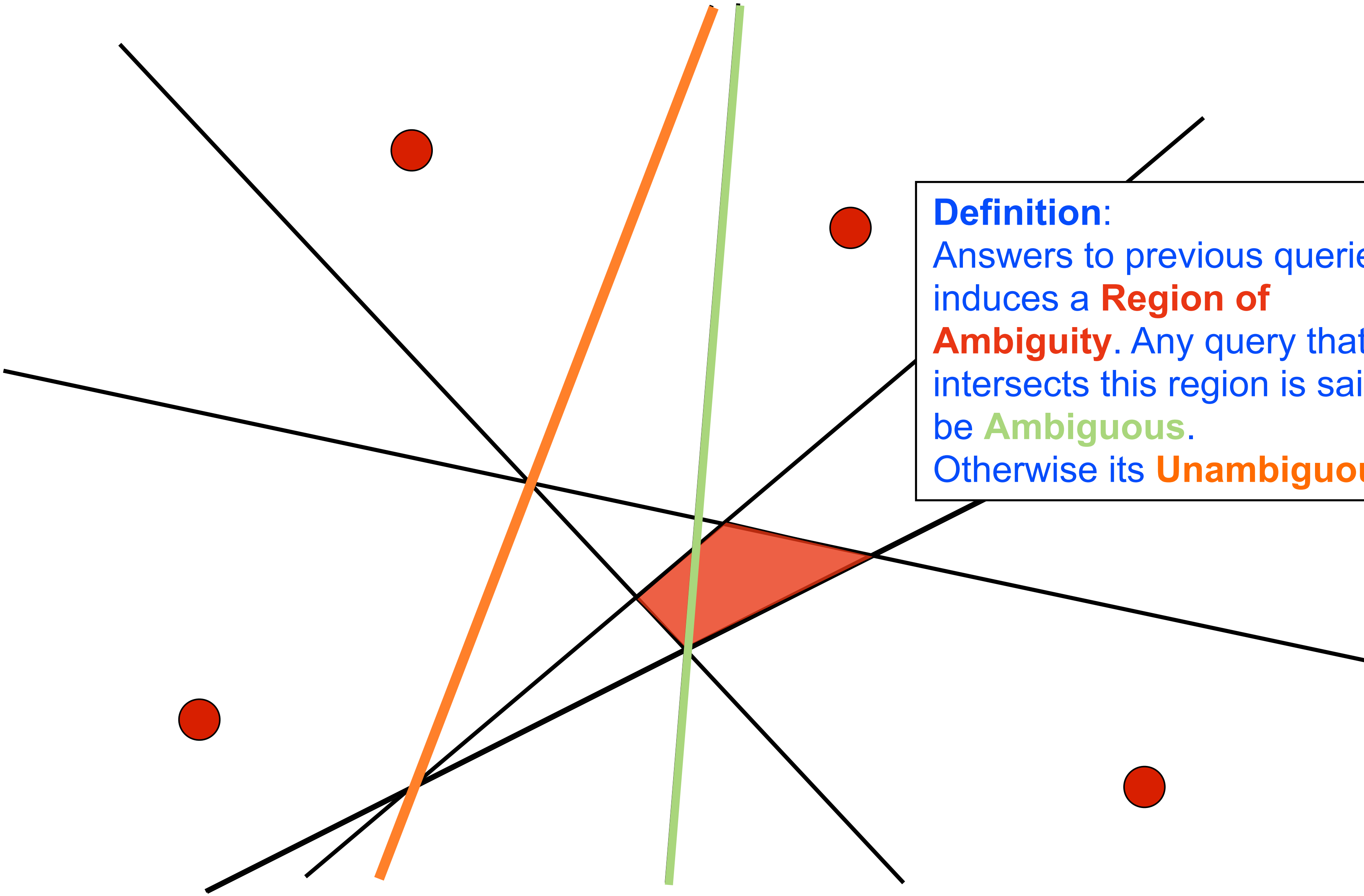
zero info-gain

output: ranking of $x_1, \ldots, x_{n-1}$ consistent with *all* pairwise comparisons

Ranking and Geometry

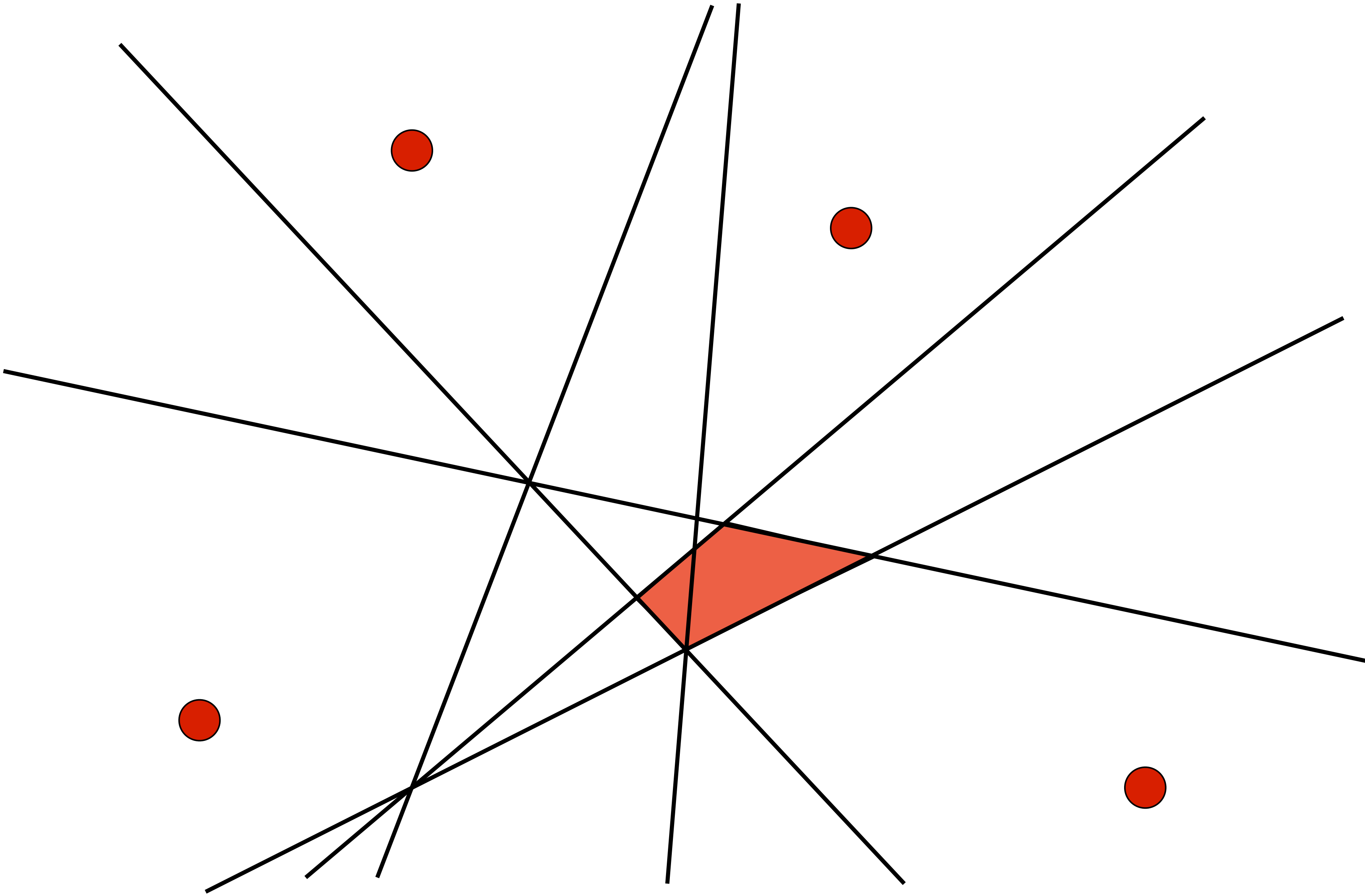


Ranking and Geometry



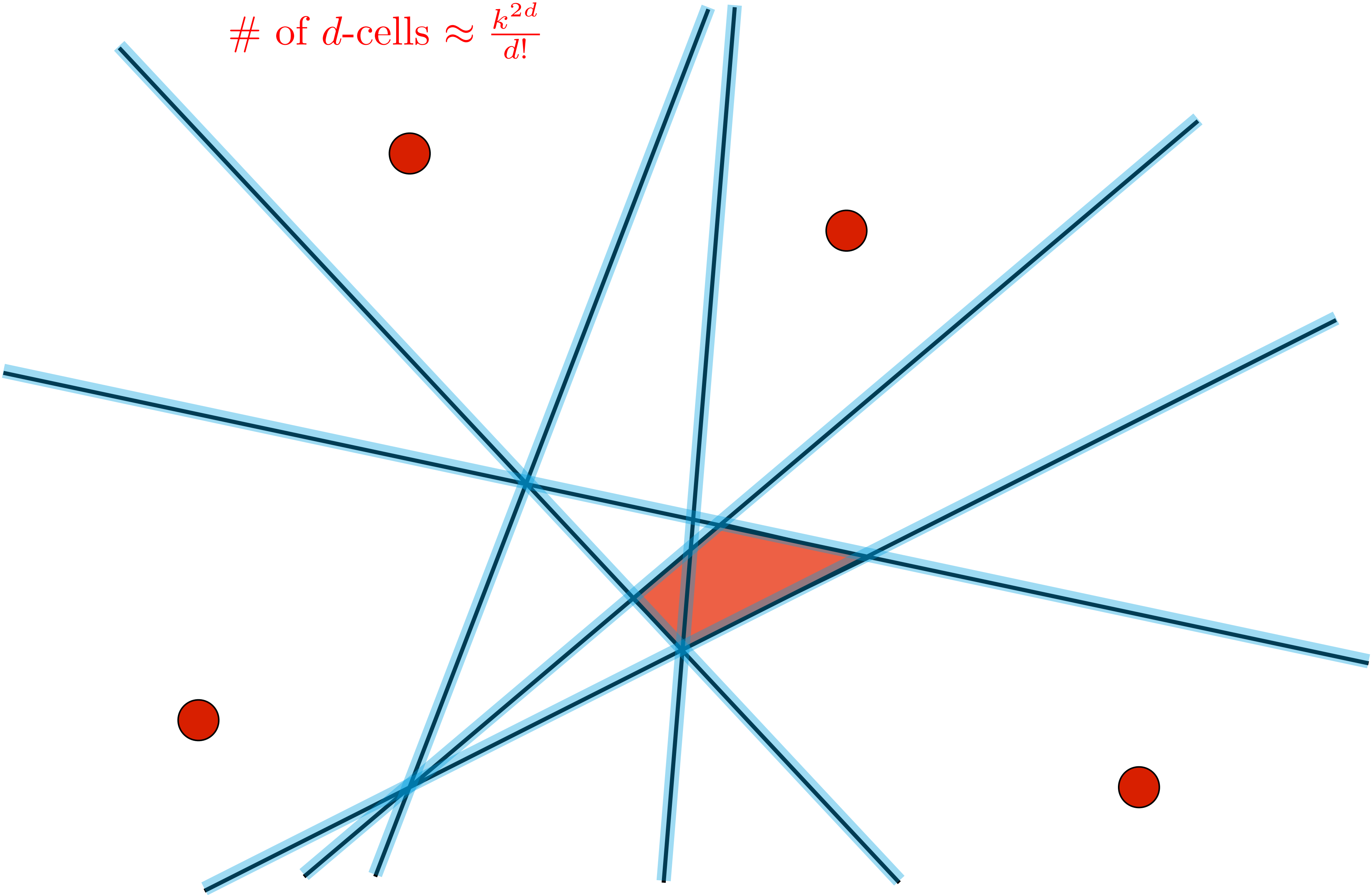
Definition:
Answers to previous queries induces a **Region of Ambiguity**. Any query that intersects this region is said to be **Ambiguous**.
Otherwise its **Unambiguous**

Ranking and Geometry

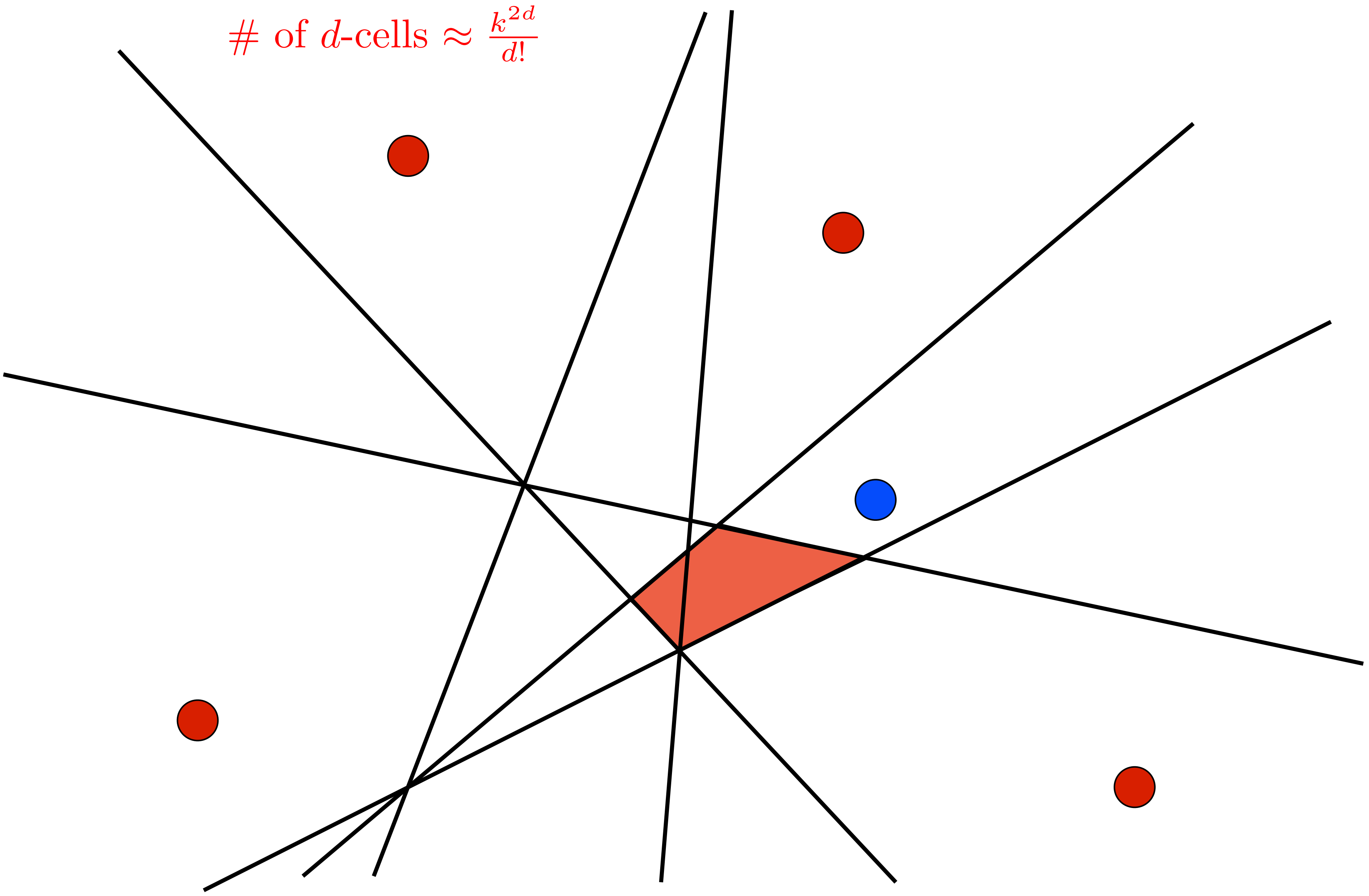


Ranking and Geometry

$\#$ of d-cells $\approx \frac{k^{2d}}{d!}$



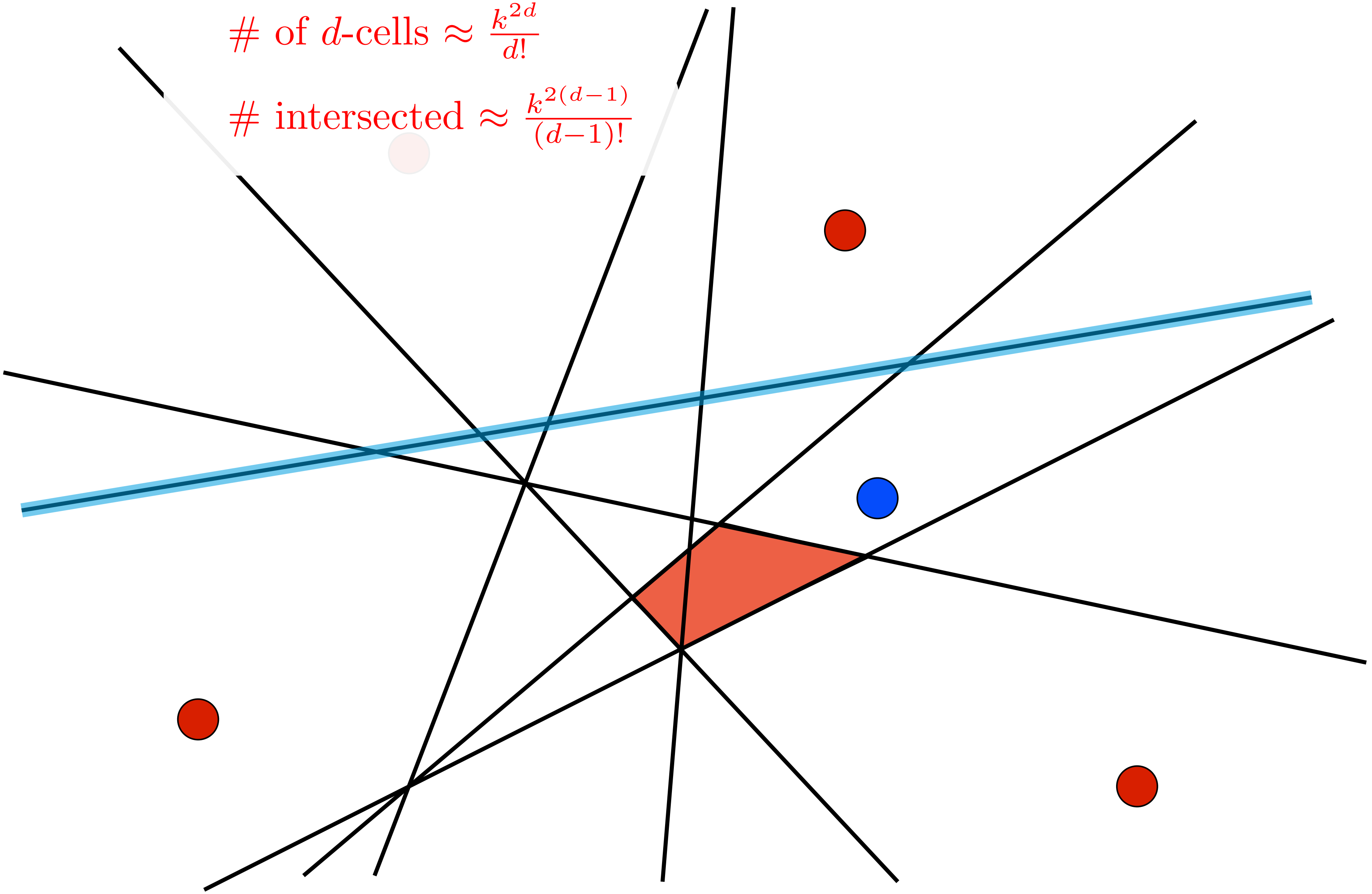
Ranking and Geometry



Ranking and Geometry

$$\#\text{ of }d\text{-cells} \approx \frac{k^{2d}}{d!}$$

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

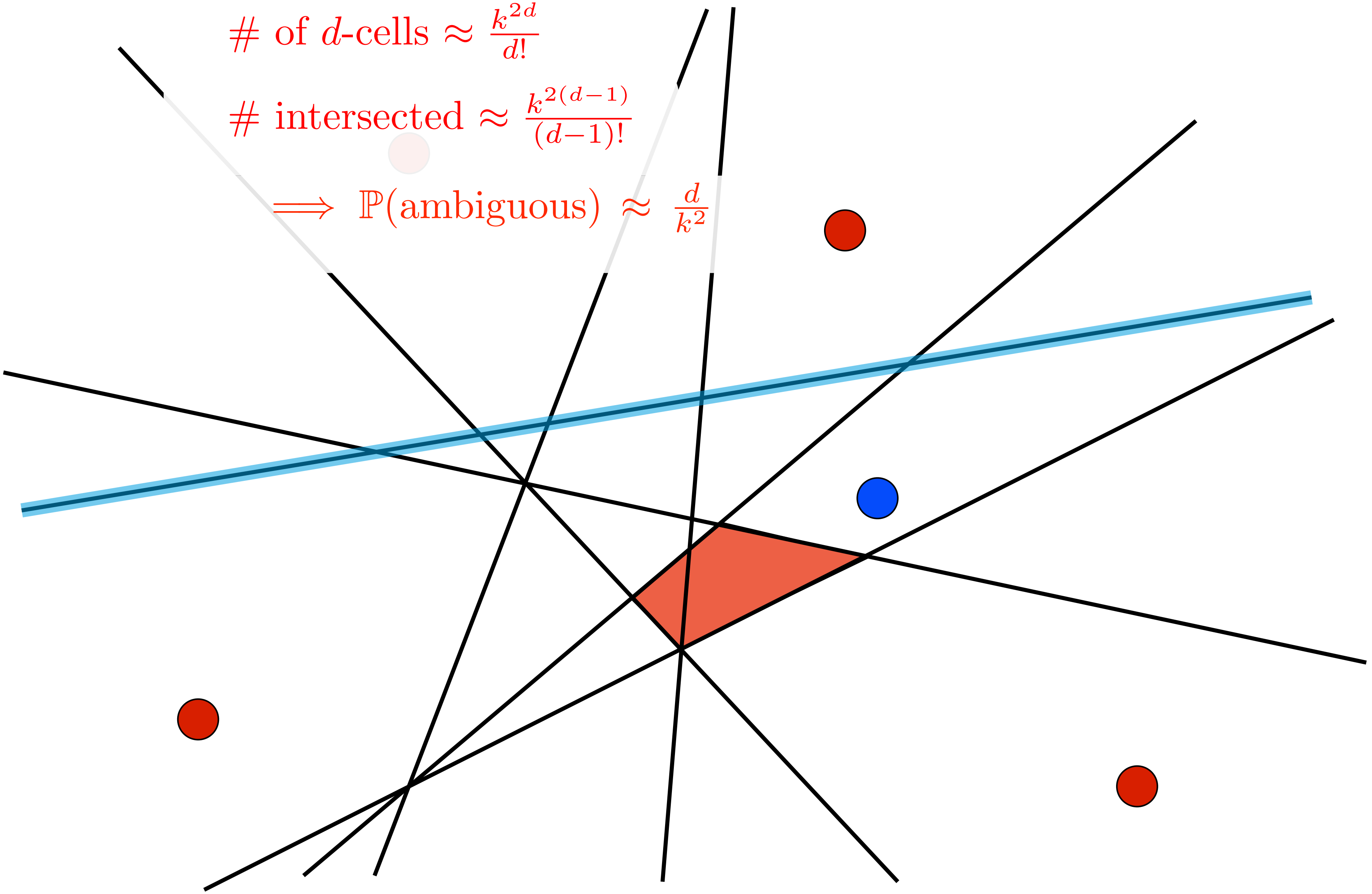


Ranking and Geometry

$$\# \text{ of } d\text{-cells} \approx \frac{k^{2d}}{d!}$$

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

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



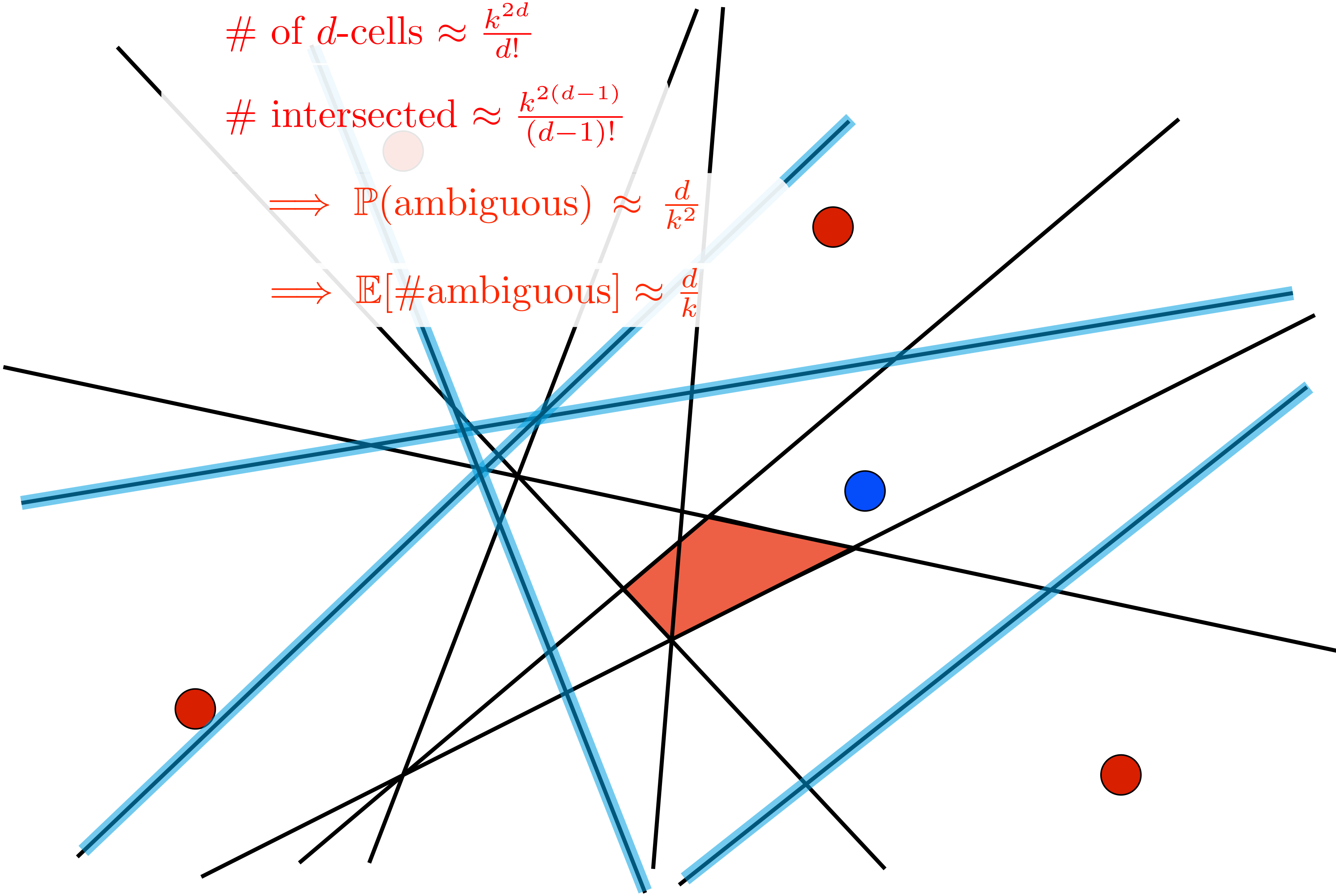
Ranking and Geometry

$$\# \text{ of } d\text{-cells} \approx \frac{k^{2d}}{d!}$$

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

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

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



Ranking and Geometry

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

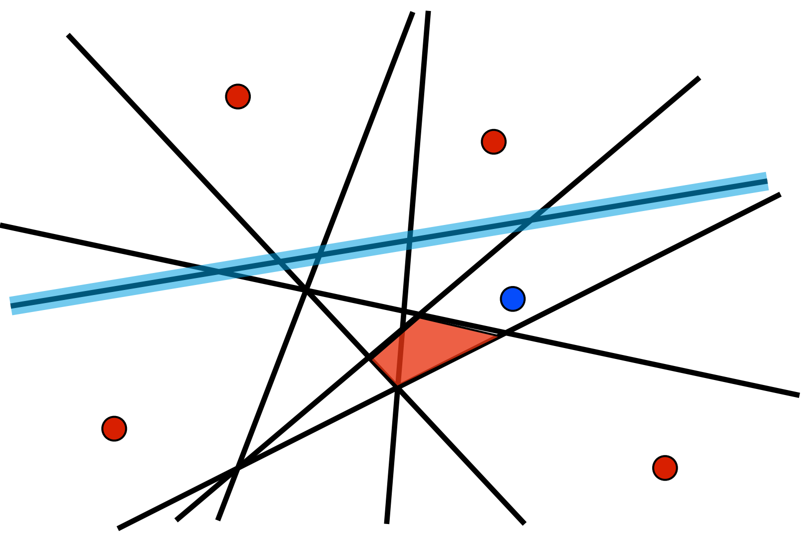
$$\#\text{ intersected} \approx \frac{k^{2(d-1)}}{(d-1)!} \qquad \text{(Buck 1943)}$$

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

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

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

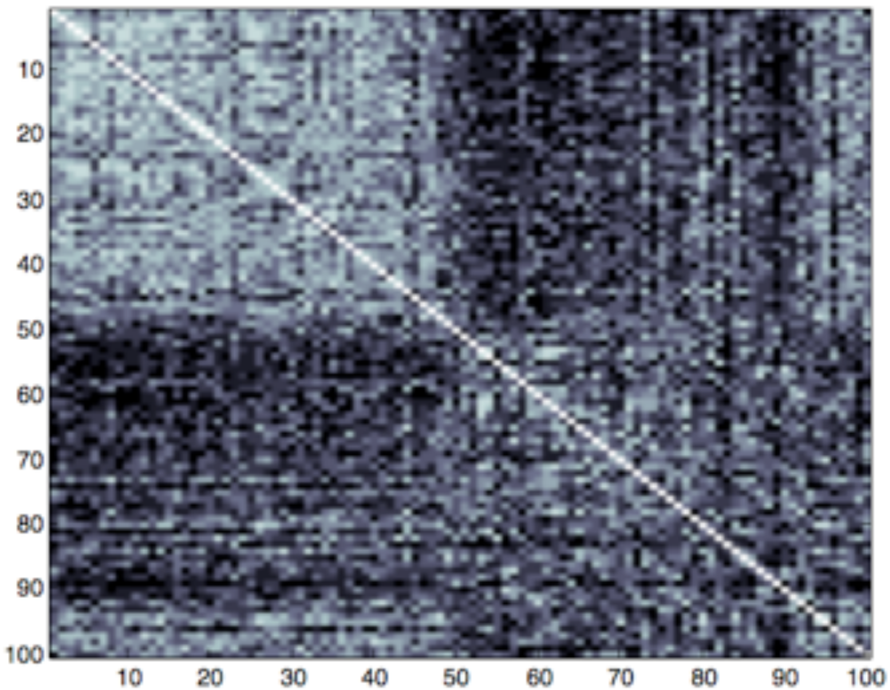
$$\approx d \log n$$



Sonar Example

Sonar echo audio signals bounced off: {50 targets, 50 rocks }

$S_{i,j} = \{\text{human-judged similarity between signals } i \text{ and } j\}$



Learning task:
Leave one signal out of the set and rank the other 99 using comparisons: $q_{i,j} \equiv \{S_{i,*} < S_{j,*}\}$

Compute d-dim embedding using MDS with similarity matrix.
$S_{i,*} < S_{j,*} \not\Leftrightarrow ||x_i - r|| < ||x_j - r||$
because embedding is approximate

Dimension		2	3
% of queries requested		14.5	18.5
Average error Kendell Tau	$d(y, \tilde{y})$	0.23	0.21
	$d(y, \hat{y})$	0.31	0.29

% of queries we requested

best achievable error

our algorithm's error

Summary

# of comparisons needed to rank n objects in d dimensions

random selection	$O(n^2)$
sequential w/o geometry	$O(n \log n)$
exploiting geometry	$O(d \log n)$
noise-tolerant	$O(d \log^2 n)$

K. Jamieson and RN. *Active ranking using pairwise comparisons*. Neural Information Processing Systems (NIPS), 2011

Summary

# of comparisons needed to rank n objects in d dimensions

random selection $O(n^2)$

sequential w/o geometry $O(n \log n)$

exploiting geometry $O(d \log n)$

noise-tolerant $O(d \log^2 n)$

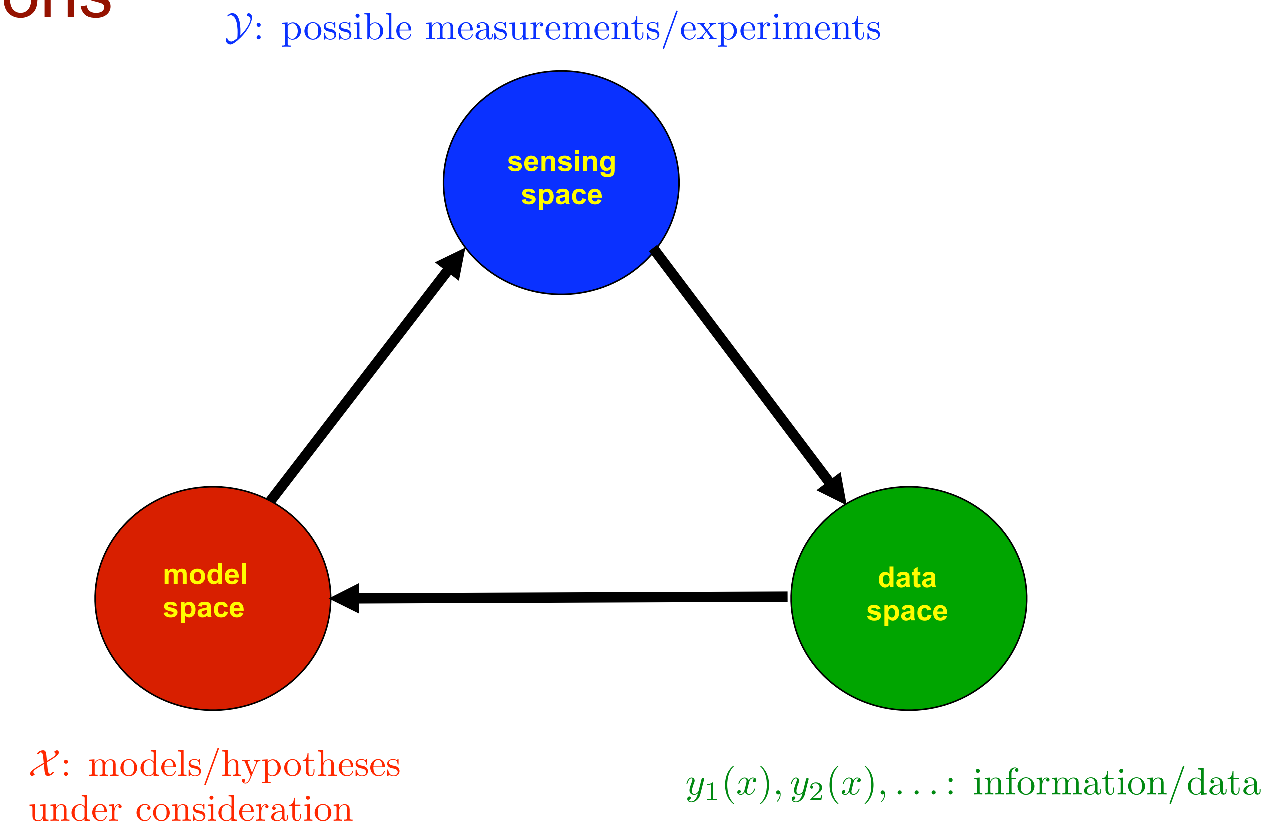
K. Jamieson and RN. *Active ranking using pairwise comparisons*. Neural Information Processing Systems (NIPS), 2011

There are other ways to limit the complexity of ranks. The combinatorial disorder D quantifies approximate triangle inequalities on ranks, and this has been used to devise more efficient ranking schemes of a similar nature

D. Tschopp, P. Delgosha, S. Mohajer, S. Diggavi. *Randomized Algorithms for Comparison-based Search.* Neural Information Processing Systems (NIPS), 2011

$$\text{ranking requires about } O(D^3 \log^2 n) \text{ pairwise comparisons}$$

Conclusions



* many learning tasks can be accelerated using interactive information gathering

* gains are often achieved because, unlike in conventional coding/information theory, there are restrictions on how information can be obtained/conveyed

* incremental information gain algorithms can be effective and sometimes optimal