

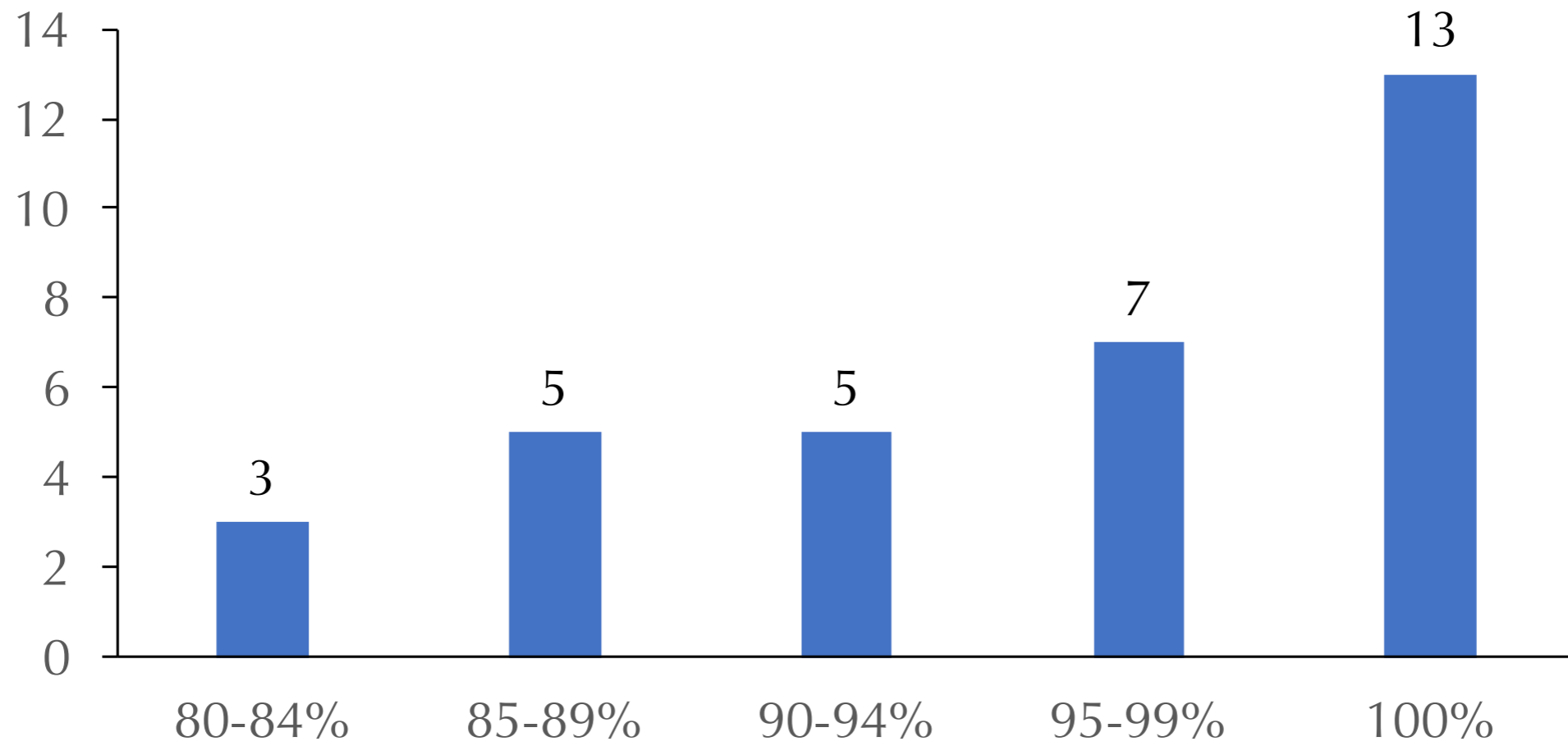
# Test Collection Experimentation

Jaime Arguello

INLS 509: Information Retrieval

[jarguell@email.unc.edu](mailto:jarguell@email.unc.edu)

# Midterm



# Outline

Parameter Tuning

Cross-validation

Significance testing

# Test Collection Evaluation

## components

- **Corpus:** set of retrievable documents
- **Topics:** queries (input to system) and descriptions of what the hypothetical user is searching for
- **Relevance judgements:** a binary or graded indicator of relevance for each query-document pair
- **Metrics:** a measure of quality that operates on a ranking of known relevant and non-relevant documents

# Test Collection Evaluation

## queries

- **Query 435:** curbing population growth
- **Description:** What measures have been taken worldwide and what countries have been effective in curbing population growth? A relevant document must describe an actual case in which population measures have been taken and their results are known. Reduction measures must have been actively pursued. Passive events such as decrease, which involuntarily reduce population, are not relevant.

(TREC 2005 HARD Track)

# Test Collection Evaluation

## metrics

- P@N
- R@N
- Average Precision (AP)
- Normalized Discounted Cumulative Gain (NDCG)
- ....

# Parameter Tuning

## motivation

- Search algorithms have lots of moving parts
- We can think of these parameters as “knobs” that need to be tweaked or tuned
- Objective:
  - ▶ Find the parameter values that maximize performance (e.g., average P@10)
  - ▶ Estimate how well the system will perform using the optimal parameter values
- Can you think of some example parameters?

# Parameter Tuning

- Query-likelihood model with linear interpolation

$$\text{score}(Q, D) = \prod_{q \in Q} (\lambda P(q|\theta_D) + (1 - \lambda)P(q|\theta_C))$$

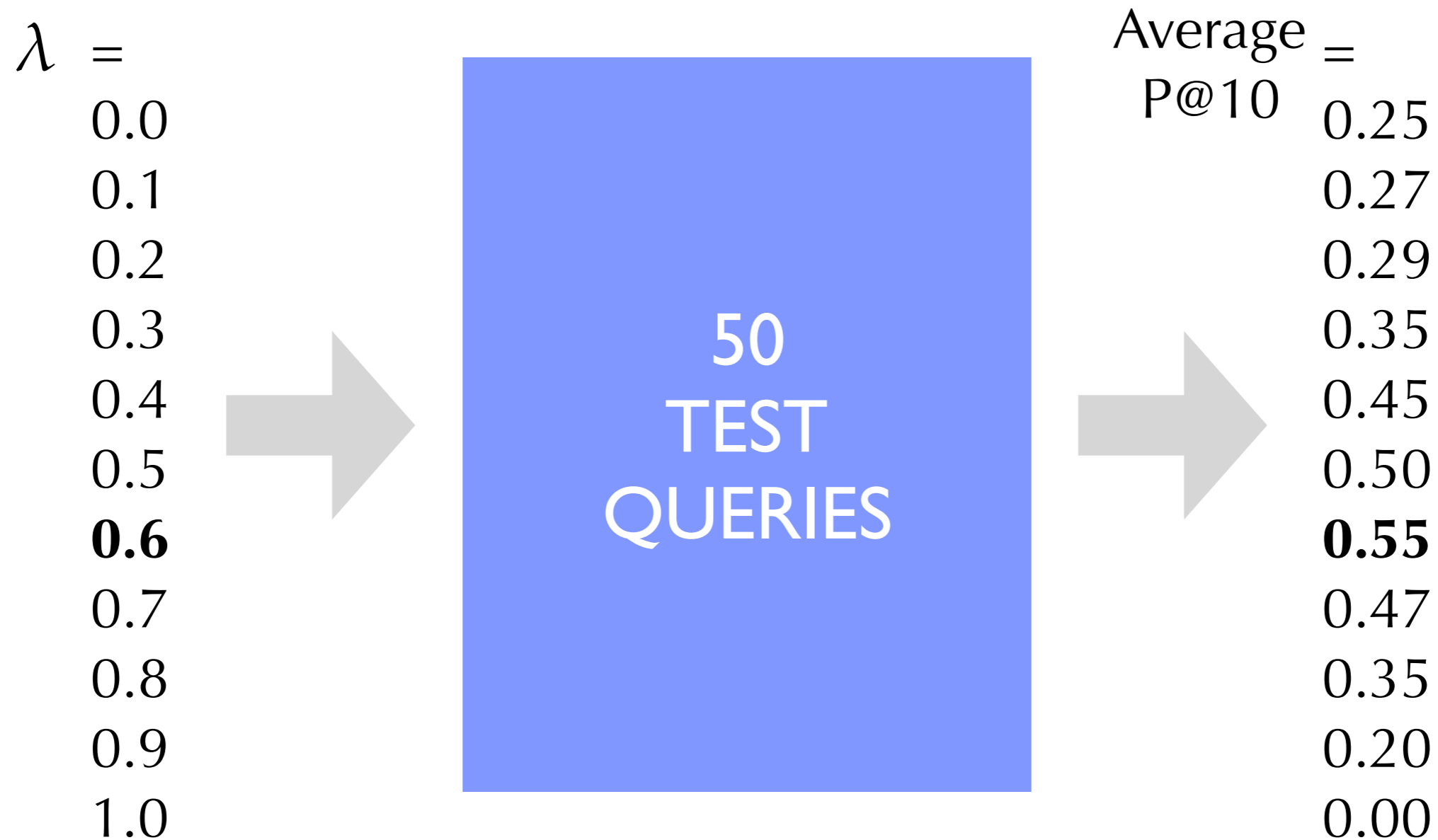
- Parameter  $\lambda$  avoids zero probabilities when a document is missing a query-term
- How should we determine the best value of  $\lambda$  and how should we estimate performance with this value?



# Parameter Tuning

- How should we determine the value of  $\lambda$  ?
- **Option -2:** roll the dice, close your eyes, and hope for the best
- **Option -1:** take a conservative guess (e.g.,  $\lambda = 0.5$ )?
- **Option 0:** take an “intuitive” guess (e.g.,  $\lambda = 0.7$ )?
- **Option 1:** try out a range of values (e.g.,  $\lambda = 0.0, 0.1, 0.2, \dots, 1.0$ ) and set it to the value that maximizes performance based on a sensible metric?

# Parameter Tuning



How well will the QL model do after parameter tuning?

# Parameter Tuning

## toy example

- **Objective:** distinguish between stars, squares, and circles



- **Parameters:** the relative importance between (1) size, (2) color, and (3) number of sides

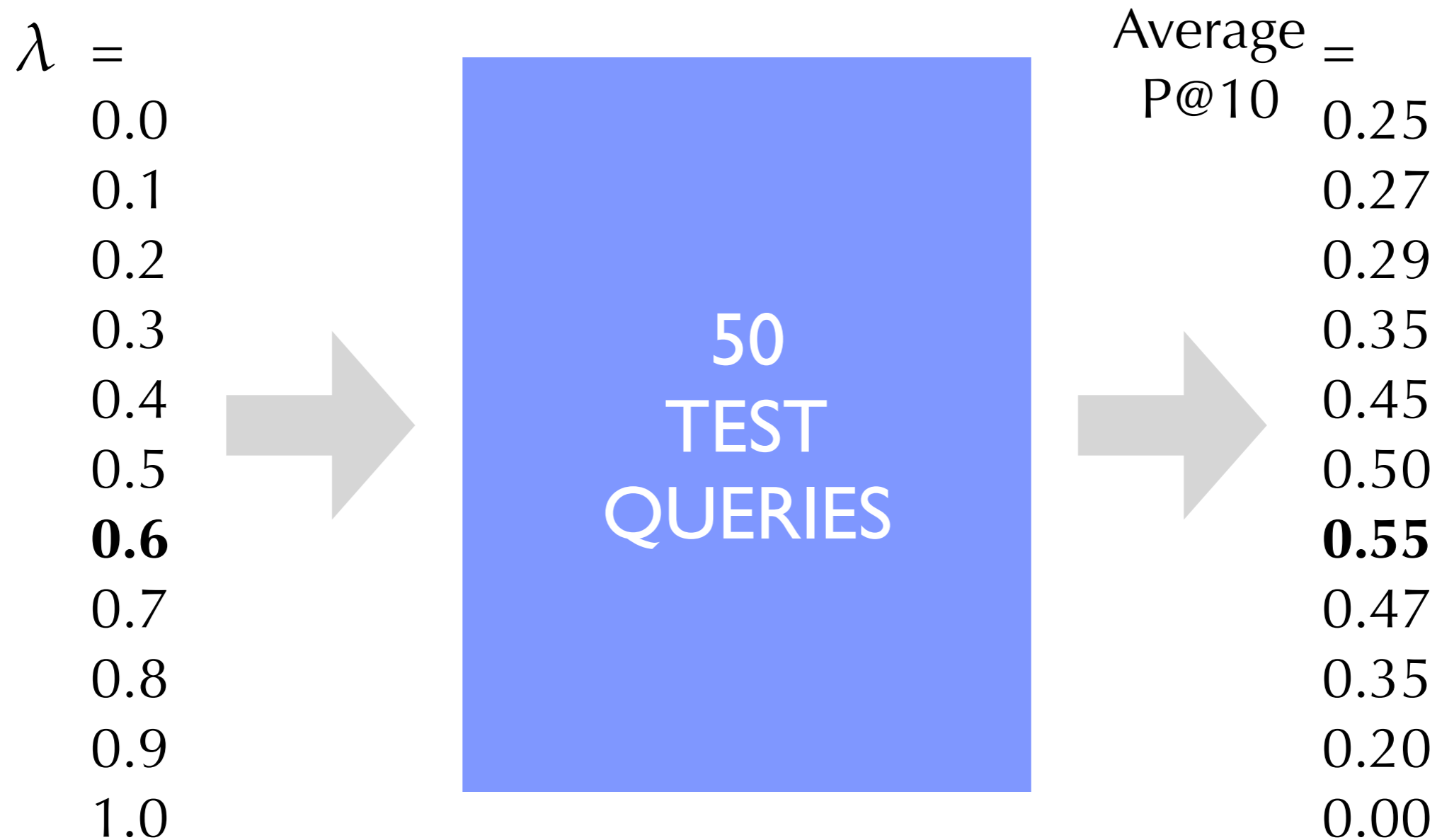
# Parameter Tuning

- The goal is to estimate the model performance using the optimal parameter values
- What is the performance that we are really interested in?

# Parameter Tuning

- The goal is to estimate the model performance using the optimal parameter values
- What is the performance that we are really interested in?
- Performance on previously unseen queries!
- We care about generalization performance!
- Our sample of queries may contain regularities that are not meaningful
- We care about those regularities that generalize to new queries!

# Parameter Tuning



Why is **0.55** a bad estimate of performance on new queries?

# Parameter Tuning

- Option 2:
  1. divide the set of 50 queries into two sets:
    - ▶ **training set:** a set of queries used to find the best parameter values (e.g., 40 queries)
    - ▶ **test set:** a held-out set used to evaluate model performance (e.g., 10 queries)
  2. **train:** find the parameter value that maximizes average performance on the training set
  3. **test:** evaluate the model (with the best training-set parameter value) on the test set

# Parameter Tuning

DATASET  
(50 queries)



# Parameter Tuning

- Split the data into two sets.
- Find the parameter value that maximize average performance on the training set.
- Evaluate the system with that parameter value on the test set.



$\lambda = 0.6$

P@10 = 0.50

# Parameter Tuning

- Split the data into two sets.
- Find the parameter value that maximize average performance on the training set.
- Evaluate the system with that parameter value on the test set.



$\lambda = 0.6$

P@10 = 0.50

Advantages and Disadvantages?

# Single Train/Test Split

- Advantage
  - ▶ the data used to find the optimal parameter value is not the same data used to test!
  - ▶ we are testing generalization performance.
- Disadvantage
  - ▶ we are putting all our eggs in one basket!
  - ▶ out of pure coincidence, the training set may have regularities that don't generalize to the test set

# Parameter Tuning

- Option 3: cross-validation
  1. divide the set of 50 queries into  $N$  sets of  $50/N$  queries
  2. use the union of  $N-1$  sets to find the best parameter values
  3. measure performance (using the best parameters) on the held-out set
  4. do steps 2-3  $N$  times
  5. average performance across the  $N$  held-out sets
- This is called  $N$ -fold cross-validation (usually,  $N=10$ )

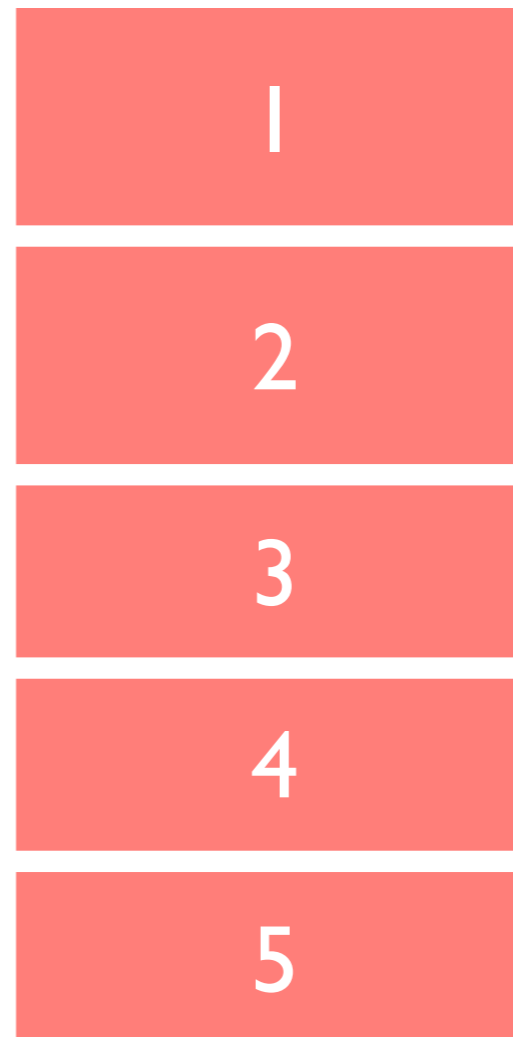
# Cross-Validation



DATASET  
(50 queries)

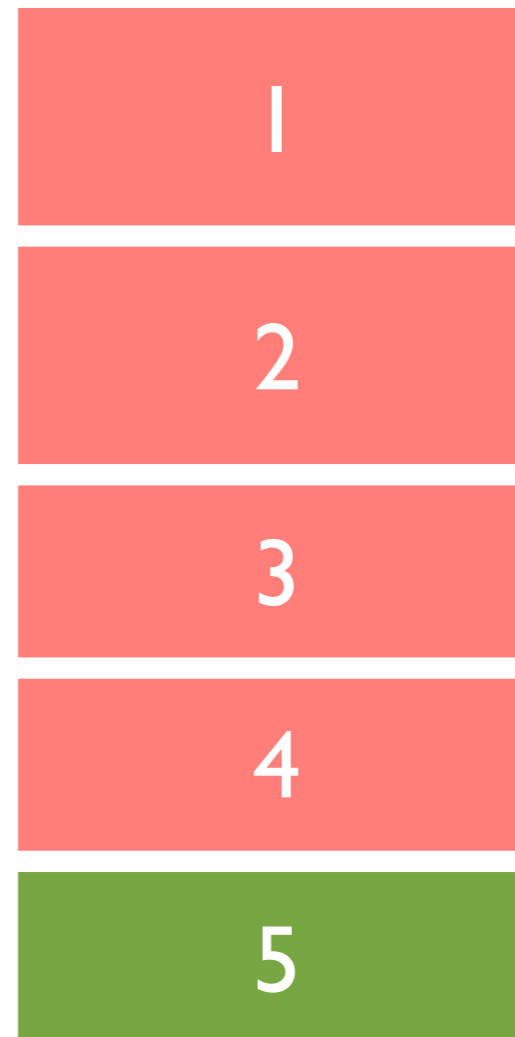
# Cross-Validation

- Split the data into  $N = 5$  folds of 10 queries each



# Cross-Validation

- For each fold, find the parameter value that maximizes average performance on the union of  $N - 1$  folds and test this parameter value on the held-out fold.

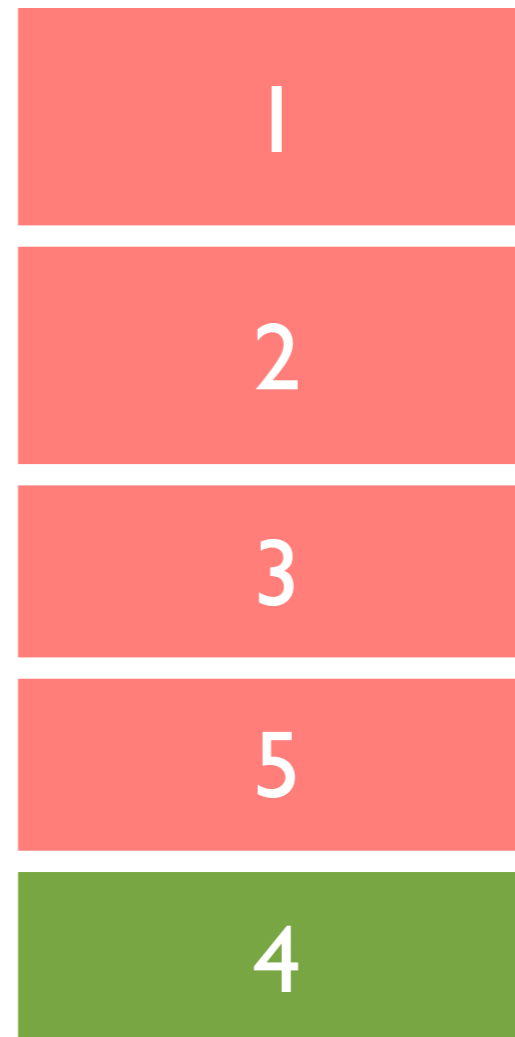


$\lambda = 0.6$

$P@10 = 0.50$

# Cross-Validation

- For each fold, find the parameter value that maximizes average performance on the union of  $N - 1$  folds and test this parameter value on the held-out fold.



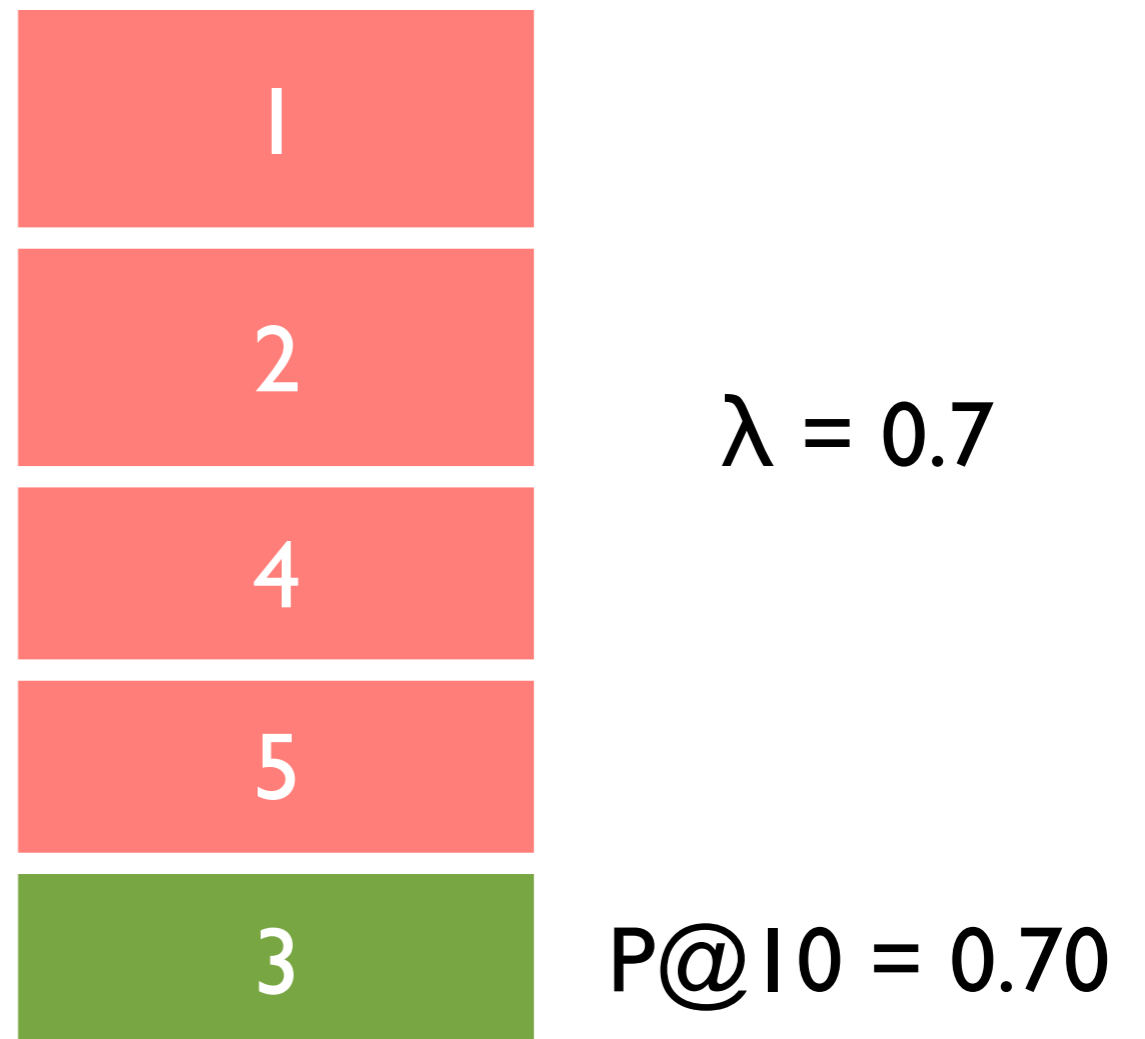
$\lambda = 0.5$

$P@10 = 0.55$



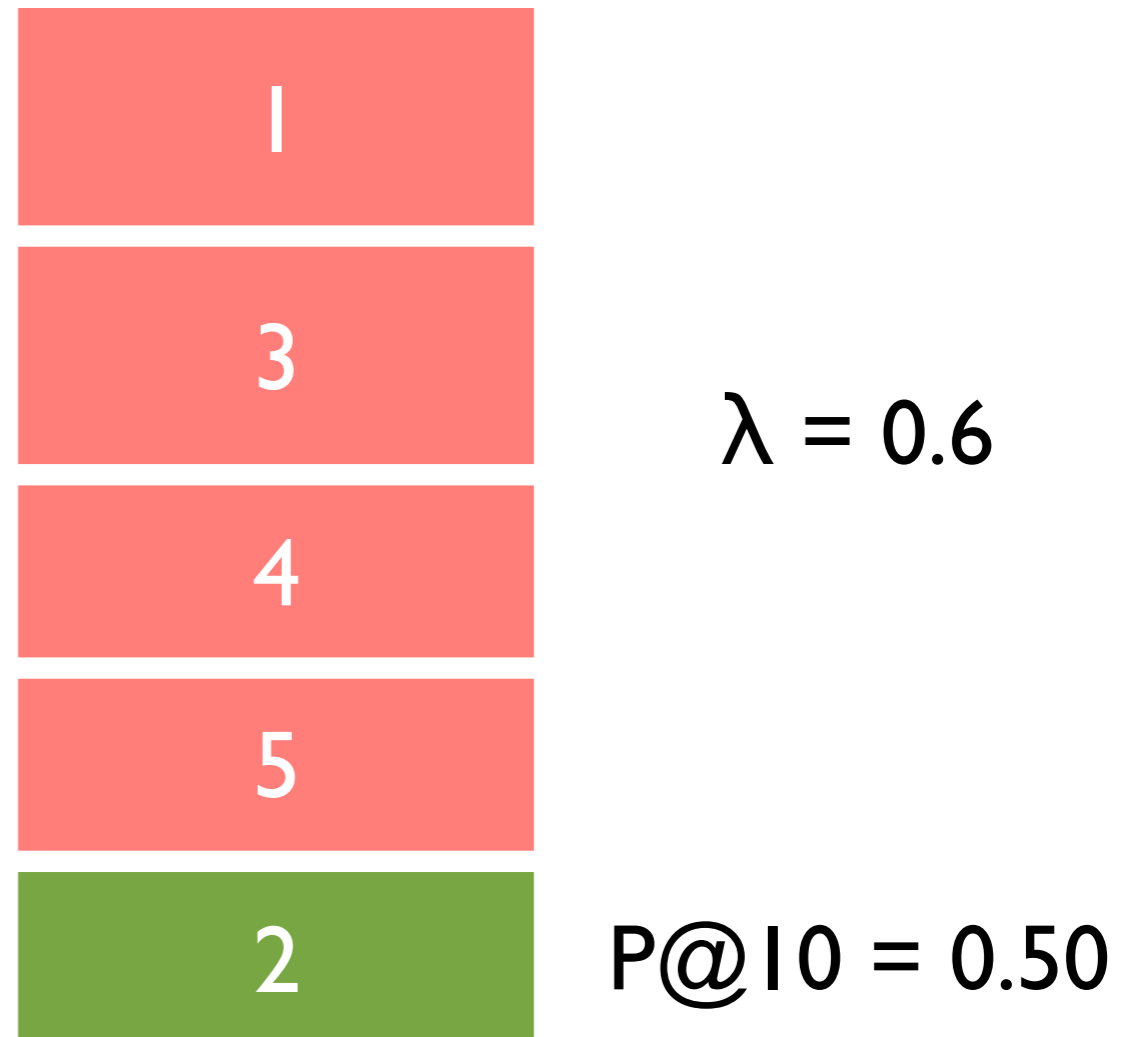
# Cross-Validation

- For each fold, find the parameter value that maximizes average performance on the union of  $N - 1$  folds and test this parameter value on the held-out fold.



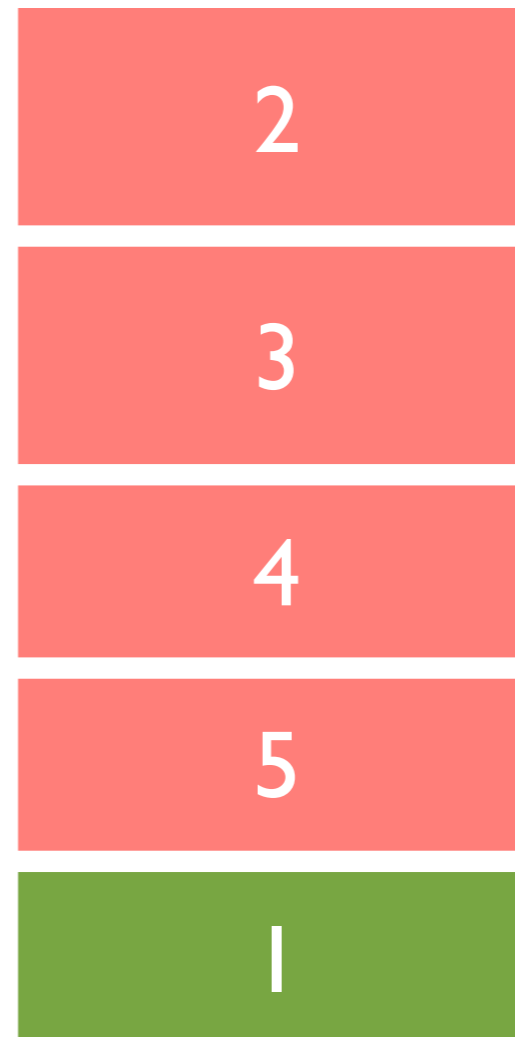
# Cross-Validation

- For each fold, find the parameter value that maximizes average performance on the union of  $N - 1$  folds and test this parameter value on the held-out fold.



# Cross-Validation

- For each fold, find the parameter value that maximizes average performance on the union of  $N - 1$  folds and test this parameter value on the held-out fold.



$\lambda = 0.4$

$P@10 = 0.80$

# Cross-Validation

- Average the performance across held-out folds

1	$P@10 = 0.80$
2	$P@10 = 0.50$
3	$P@10 = 0.70$
4	$P@10 = 0.55$
5	$P@10 = 0.50$
<b>Average</b>	<b><math>P@10 = 0.61</math></b>

# Cross-Validation

- Average the performance across held-out folds

1	$P@10 = 0.80$
2	$P@10 = 0.50$
3	$P@10 = 0.70$
4	$P@10 = 0.55$
5	$P@10 = 0.50$
<b>Average</b>	<b><math>P@10 = 0.61</math></b>

Advantages and Disadvantages?

# N-Fold Cross-Validation

- Advantage
  - ▶ multiple rounds of generalization performance.
- Disadvantage
  - ▶ ultimately, we'll tune parameters on the set of 50 queries and send our system into the world.
  - ▶ a model trained on 50 queries should perform better than one trained on 40.
  - ▶ thus, we may be underestimating the model's performance!

# Leave-One-Out Cross-Validation



# Leave-One-Out Cross-Validation

- Split the data into  $N = 50$  folds of 1 queries each





# Leave-One-Out Cross-Validation

- For each query, find the parameter value that maximize performance on for the other queries and and test (using this parameter value) on the held-out query.



# Leave-One-Out Cross-Validation

- For each query, find the parameter value that maximize performance on for the other queries and and test (using this parameter value) on the held-out query.



# Leave-One-Out Cross-Validation

- For each query, find the parameter value that maximize performance on for the other queries and and test (using this parameter value) on the held-out query.
- And so on ...
- Finally, average the performance for each held-out query



# Leave-One-Out Cross-Validation

- For each query, find the parameter value that maximize performance on for the other queries and and test (using this parameter value) on the held-out query.
- And so on ...
- Finally, average the performance for each held-out query



Advantages and Disadvantages?

# Leave-One-Out Cross-Validation

- Advantages
  - ▶ multiple rounds of generalization performance.
  - ▶ each training fold is as similar as possible to the one we will ultimately use to tune parameters before sending the system out into the world.
- Disadvantage
  - ▶ our estimate of generalization performance may still be artificially high
  - ▶ why?

# Leave-One-Out Cross-Validation

- Advantages
  - ▶ multiple rounds of generalization performance.
  - ▶ each training fold is as similar as possible to the one we will ultimately use to tune parameters before sending the system out into the world.
- Disadvantage
  - ▶ our estimate of generalization performance may still be artificially high
  - ▶ we are likely to try lots of different things and pick the one with the best “generalization” performance
  - ▶ still indirectly over-training to the dataset (sigh...)

# Significance Tests

Jaime Arguello

INLS 509: Information Retrieval

[jarguell@email.unc.edu](mailto:jarguell@email.unc.edu)

# Outline

Parameter Tuning

Cross-validation

Significance testing



# Comparing Between Systems

- The main goal in experimental IR is to develop retrieval techniques that are better than the state of the art and to understand why they are better
- **Basic question:** Is system **B** better than system **A**?
- **More often:** Is system **A with 'special sauce'** better than system **A without 'special sauce'**?

# Comparing Systems

P@10

- For each system, tune and test the necessary parameters using N-fold cross-validation
- Use the same folds for both systems
- Compare the difference in average performance across held out folds using a significance test

Fold	System A	System B
1	0.2	0.5
2	0.3	0.3
3	0.1	0.1
4	0.4	0.4
5	1	1
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
9	0	0.5
10	0.9	0.8
Average	0.41	0.48
	Difference	0.07

# Significance Tests

## motivation

- Why would it be risky to conclude that **System B** is better **System A** based on P@10?
- Put differently, what is it that we're trying to achieve?

# Significance Tests

motivation



# Significance Tests

## motivation

- **In theory:** the average performance of **System B** is greater than the average performance of **System A** for all possible queries!
- However, we don't have all queries. We have a sample (usually about 50).
- And, this sample may favor one system vs. the other!

# Significance Tests

## definition

- A **significance test** is a statistical tool that allows us to determine whether a difference in performance reflects a true pattern or just random chance

# Significance Tests

## ingredients

- **Test statistic:** a measure used to judge the two systems (e.g., the difference between their average P@10 values)
- **Null hypothesis:** no “true” difference between the two systems
- **P-value:** take the value of the observed test statistic and compute the probability of observing a value that large (or larger) under the null hypothesis

# Comparing Systems

P@10

- For each system, tune and test the

ne  
pa  
fol

- **P-value:**
- What is the probability of observing an improvement of 0.07 (or more) if it is actually true that both systems are equally good?

- Us
- fo
- Co

dif  
pe  
he

a significance test

Fold

1

System A

System B

0.2	0.5
0.3	0.3
0.1	0.1
0.4	0.4
1	1
0.8	0.9
0.3	0.1
0.1	0.2
0	0.5
0.9	0.8
0.41	0.48

Difference

0.07



# Significance Tests

## ingredients

- If the p-value is large, we cannot reject the null hypothesis
- That is, we cannot claim that one system is better than the other
- There is a high probability that the observed test statistic is due to random chance
- If the p-value is small ( $p < 0.05$ ), we can reject the null hypothesis
- That is, we can claim that the observed test-statistic is not due to random chance

# Fisher's Randomization Test




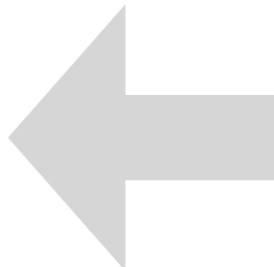
## procedure

- **Inputs:** `counter` = 0, `N` = 100,000
- Repeat `N` times:
  - Step 1:** for each fold, flip a coin and if it lands 'heads', flip the result between System A and B
  - Step 2:** see whether the test statistic is equal to or greater than the one observed and, if so, increment `counter`
- **Output:** `counter` / `N`

# Fisher's Randomization Test

Fold	System A	System B
1	0.2	0.5
2	0.3	0.3
3	0.1	0.1
4	0.4	0.4
5	1	1
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
9	0	0.5
10	0.9	0.8
Average	0.41	0.48
	Difference	0.07

# Fisher's Randomization Test

Fold	System A	System B	
1	<b>0.5</b>	<b>0.2</b>	
2	0.3	0.3	
3	0.1	0.1	
4	0.4	0.4	
5	1	1	
6	<b>0.9</b>	<b>0.8</b>	
7	0.3	0.1	
8	0.1	0.2	
9	<b>0.5</b>	<b>0</b>	
10	0.9	0.8	
Average	0.5	0.39	
	Difference	-0.11	
	<b>iteration = 1</b>	<b>counter = 0</b>	<b>at least 0.07?</b>







# Fisher's Randomization Test

Fold	System A	System B
1	0.2	0.5
2	0.3	0.3
3	<b>0.1</b>	<b>0.1</b>
4	0.4	0.4
5	<b>1</b>	<b>1</b>
6	0.8	0.9
7	<b>0.1</b>	<b>0.3</b>
8	<b>0.2</b>	<b>0.1</b>
9	0	0.5
10	<b>0.08</b>	<b>0.9</b>
Average	0.318	0.5
Difference		0.182

at least 0.07?

iteration = 2      counter = 1

# Fisher's Randomization Test

Fold	System A		System B	
1	<b>0.5</b>		<b>0.2</b>	
2	0.3		0.3	
3	<b>0.1</b>		<b>0.1</b>	
4	<b>0.4</b>		<b>0.4</b>	
5	1		1	
6	<b>0.9</b>		<b>0.8</b>	
7	0.3		0.1	
8	0.1		0.2	
9	<b>0.5</b>		<b>0</b>	
10	0.9		0.8	
Average	0.5		0.39	
	Difference		-0.11	
				at least <b>0.07?</b>

**iteration = 100,000**      **counter = 25,678**

# Fisher's Randomization Test

## procedure

- **Inputs:** `counter` = 0, `N` = 100,000
- Repeat `N` times:
  - Step 1:** for each fold, flip a coin and if it lands 'heads', flip the result between System A and B
  - Step 2:** see whether the test statistic is equal to or greater than the one observed and, if so, increment `counter`
- **Output:** `counter` / `N` = (25,678/100,00) = 0.25678

# Fisher's Randomization Test

## procedure

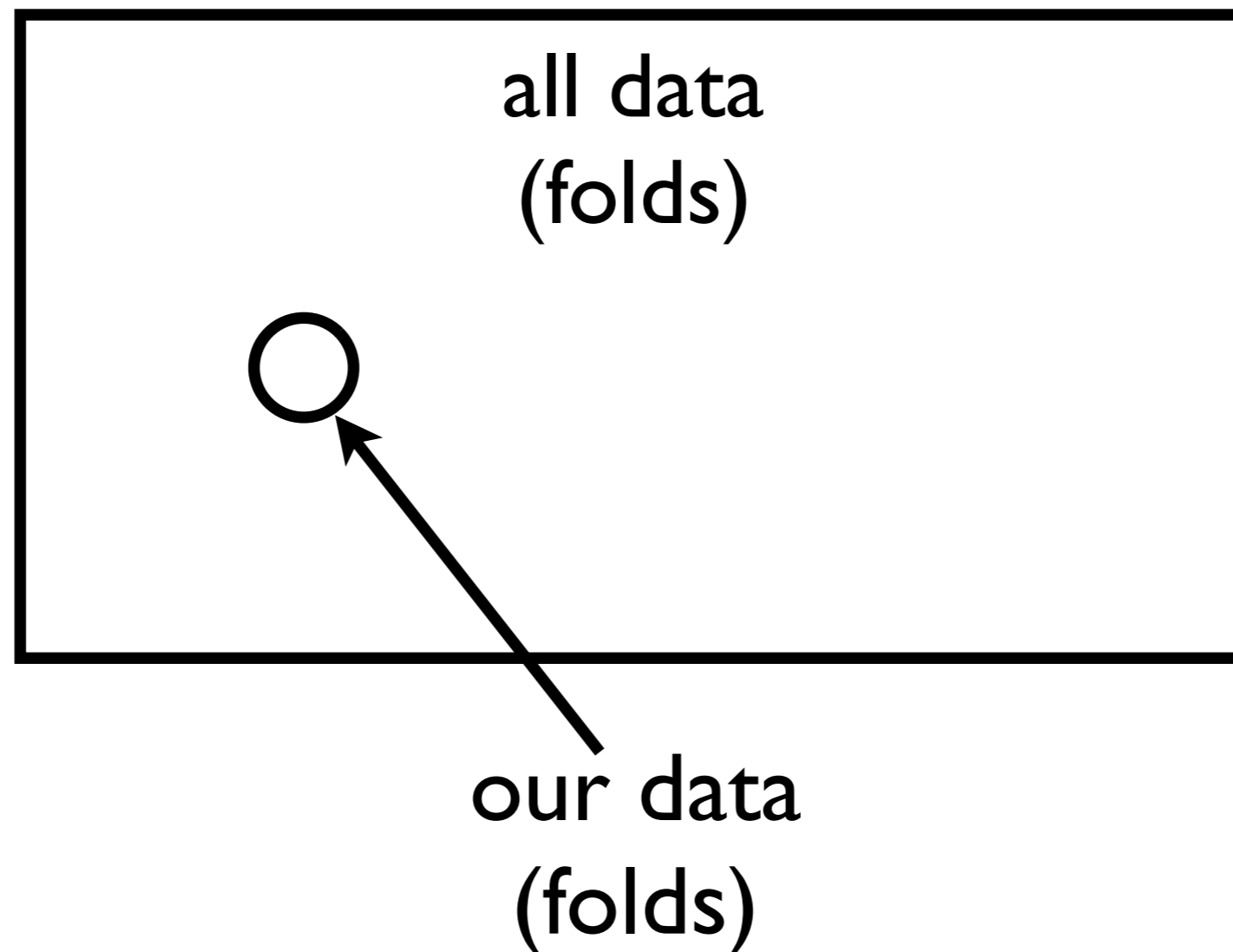
- Under the null hypothesis, the probability of observing a value of the test statistic of 0.07 or greater is about 0.26.
- Because  $p > 0.05$ , we cannot confidently say that the value of the test statistic is not due to random chance.
- A difference between the average P@10 values of 0.07 is not significant



# Bootstrap-Shift Test

## motivation

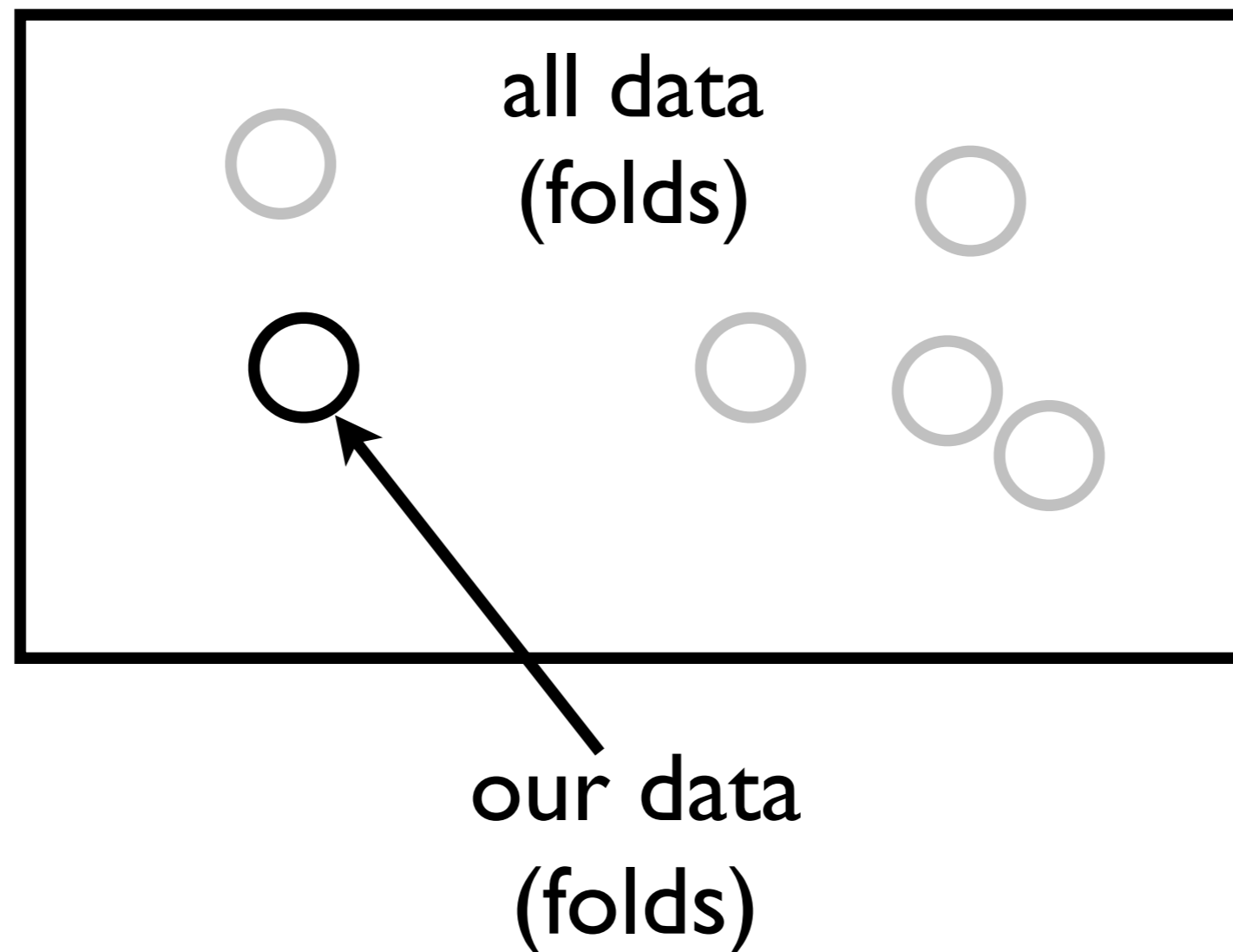
- Our sample is a representative sample of all data



# Bootstrap-Shift Test

## motivation

- Suppose we could sample many other folds.
- Assuming that the null hypothesis is true, what would be the average test statistic value across all those folds?



# Comparing Systems

P@10

- For each system, tune and test the new parameter on a held out fold
- Use the performance across held out folds using a significance test

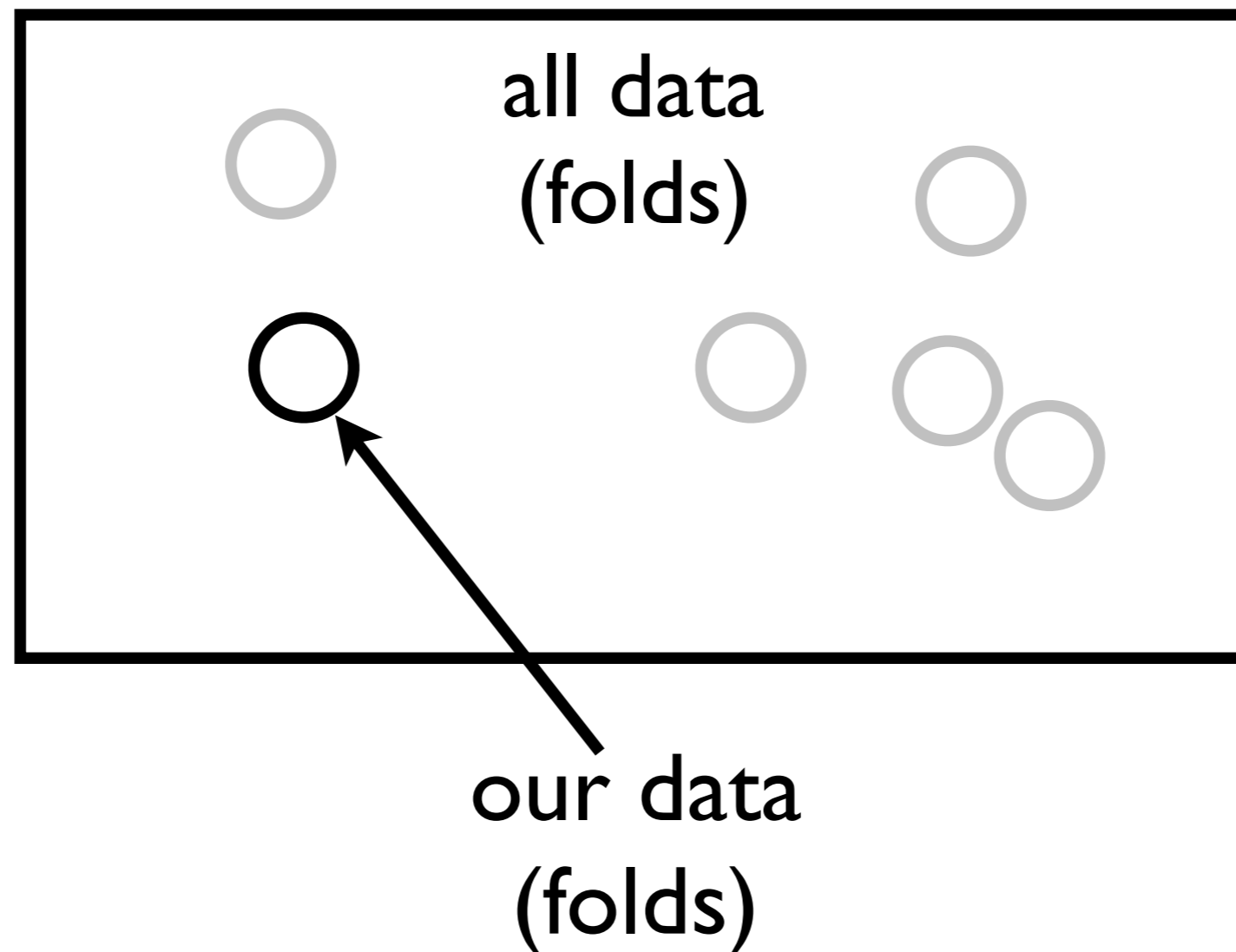
- Suppose we could repeat this experiment with many other data samples.
- Assuming that the null hypothesis is true, what would be the average of this test statistic?

Fold	System A	System B
1	0.2	0.5
	0.3	0.3
	0.1	0.1
	0.4	0.4
1	0.8	0.9
	0.3	0.1
	0.1	0.2
	0	0.5
	0.9	0.8
Average	0.41	0.48
Difference		0.07

# Bootstrap-Shift Test

## motivation

- If we sample (with replacement) from our sample, we can generate a new representative sample of all data



# Bootstrap-Shift Test procedure

- **Inputs:** Array  $T = \{\}$ ,  $N = 100,000$
- Repeat  $N$  times:
  - Step 1:** sample 10 folds (with replacement) from our set of 10 folds (called a subsample)
  - Step 2:** compute test statistic associated with new sample and add to  $T$
- **Step 3:** compute average of numbers in  $T$
- **Step 4:** reduce every number in  $T$  by average
- **Output:** % of numbers in  $T$  greater than or equal to the observed test statistic

# Bootstrap-Shift Test

Fold	System A	System B
1	0.2	0.5
2	0.3	0.3
3	0.1	0.1
4	0.4	0.4
5	1	1
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
9	0	0.5
10	0.9	0.8
Average	0.41	0.48
	Difference	0.07

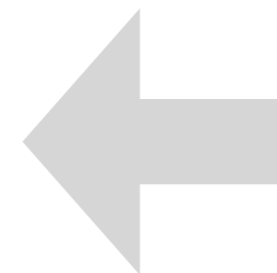
# Bootstrap-Shift Test

Fold	System A	System B	sample
1	0.2	0.5	<b>0</b>
2	0.3	0.3	<b>1</b>
3	0.1	0.1	<b>2</b>
4	0.4	0.4	<b>2</b>
5	1	1	<b>0</b>
6	0.8	0.9	<b>1</b>
7	0.3	0.1	<b>1</b>
8	0.1	0.2	<b>1</b>
9	0	0.5	<b>2</b>
10	0.9	0.8	<b>0</b>

iteration = |

# Bootstrap-Shift Test

Fold	System A	System B
2	0.3	0.3
3	0.1	0.1
3	0.1	0.1
4	0.4	0.4
4	0.4	0.4
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
9	0	0.5
9	0	0.5
Average	0.25	0.35
	Difference	<b>0.1</b>



$$T = \{0.10\}$$

iteration = 1



# Bootstrap-Shift Test

Fold	System A	System B	sample
1	0.2	0.5	<b>0</b>
2	0.3	0.3	<b>0</b>
3	0.1	0.1	<b>3</b>
4	0.4	0.4	<b>2</b>
5	1	1	<b>0</b>
6	0.8	0.9	<b>1</b>
7	0.3	0.1	<b>1</b>
8	0.1	0.2	<b>1</b>
9	0	0.5	<b>1</b>
10	0.9	0.8	<b>1</b>

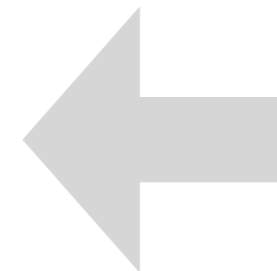
**T** = {**0.10**}

iteration = 2

# Bootstrap-Shift Test

Fold	System A	System B
3	0.1	0.1
3	0.1	0.1
3	0.1	0.1
4	0.4	0.4
4	0.4	0.4
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
9	0	0.5
10	0.9	0.8
Average	0.32	0.36
	Difference	<b>0.04</b>

iteration = 2



$$T = \{ \mathbf{0.10}, \mathbf{0.04} \}$$

# Bootstrap-Shift Test

Fold	System A	System B
1	0.2	0.5
1	0.2	0.5
4	0.4	0.4
4	0.4	0.4
4	0.4	0.4
6	0.8	0.9
7	0.3	0.1
8	0.1	0.2
8	0.1	0.2
10	0.9	0.8
Average	0.38	0.44

Difference **00000000**

**iteration = 100,000**

**T = {0.10,**  
**0.04,**  
**.....,**  
**0.06}**

# Bootstrap-Shift Test procedure

- **Inputs:** Array  $T = \{\}$ ,  $N = 100,000$
- Repeat  $N$  times:
  - Step 1:** sample 10 folds (with replacement) from our set of 10 folds (called a subsample)
  - Step 2:** compute test statistic associated with new sample and add to  $T$
- **Step 3:** compute average of numbers in  $T$
- **Step 4:** reduce every number in  $T$  by average
- **Output:** % of numbers in  $T$  greater than or equal to the observed test statistic

# Bootstrap-Shift Test procedure

- For the purpose of this example, let's assume  $N = 10$ .

$T = \{0.10,$   
 $0.04,$   
 $0.21,$   
 $0.20,$   
 $0.13,$   
 $0.09,$   
 $0.22,$   
 $0.07,$   
 $0.03,$   
 $0.11\}$

Step 3



Step 4

$T' = \{-0.02,$   
 $-0.08,$   
 $0.09,$   
 $0.08,$   
 $0.01,$   
 $-0.03,$   
 $0.10,$   
 $-0.05,$   
 $-0.09,$   
 $-0.01\}$

Average = 0.12

# Bootstrap-Shift Test procedure

- **Inputs:** Array  $T = \{\}$ ,  $N = 100,000$
- Repeat  $N$  times:
  - Step 1:** sample 10 folds (with replacement) from our set of 10 folds (called a subsample)
  - Step 2:** compute test statistic associated with new sample and add to  $T$
- **Step 3:** compute average of numbers in  $T$
- **Step 4:** reduce every number in  $T$  by average
- **Output:** % of numbers in  $T'$  greater than or equal to the observed test statistic

# Bootstrap-Shift Test procedure

- **Output:**  $(3/10) = \mathbf{0.30}$

$T = \{$   
  **0.10,**  
  **0.04,**  
  **0.21,**  
  **0.20,**  
  **0.13,**  
  **0.09,**  
  **0.22,**  
  **0.07,**  
  **0.03,**  
  **0.11}**

**Step 3**



**Step 4**

$T' = \{$   
  **-0.02,**  
  **-0.08,**  
  **0.09,**  
  **0.08,**  
  **0.01,**  
  **-0.03,**  
  **0.10,**  
  **-0.05,**  
  **-0.09,**  
  **-0.01}**

**Average = 0.12**

# Significance Tests

## summary

- Significance tests help us determine whether the outcome of an experiment signals a “true” trend
- The null hypothesis is that the observed outcome is due to random chance (sample bias, error, etc.)
- There are many types of tests
- **Parametric tests:** assume a particular distribution for the test statistic under the null hypothesis
- **Non-parametric tests:** make no assumptions about the test statistic distribution under the null hypothesis
- The **randomization** and **bootstrap-shift** tests make no assumptions, are robust, and easy to understand