IR Experimentation

Jaime Arguello
INLS 890-190: Experimental IR
jarguell@email.unc.edu

January 27, 2015

Outline

Test-collection evaluation (review)

Significance tests

Parameter Tuning

Cross-validation

Evaluation

- 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 A better than system B?
- In your term projects: Is system A with 'special sauce' better than system A without 'special sauce'?

Test Collection Evaluation

components

- Collection: a corpus of retrievable items or 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 decease, which involuntarily reduce population, are not relevant.

(TREC 2005 HARD Track)

Test Collection Evaluation

metrics

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

Comparing Systems
P@10

Query	System A	System B
1	0.20	0.50
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
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

- 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: the measure used to compare the performance of the two systems (e.g., the difference between their average P@10)
- 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

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
- If the p-value is small (p<0.05), we can reject the null hypothesis
- That is, 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 query, 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

Query	System A	System B
1	0.20	0.50
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
Average	0.41	0.48
	Difference	0.07

Fisher's Randomization Test

Query	System A	System B
1	**0.50**	**0.20**
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	**0.90**	**0.80**
7	0.30	0.10
8	0.10	0.20
9	**0.50**	**0.00**
10	0.90	0.80
Average	0.5	0.39
	Difference	-0.11

at least 0.07?

iteration = 1 counter = 0

Fisher’s Randomization Test

Query	System A	System B
1	0.20	0.50
2	0.30	0.30
3	**0.10**	**0.10**
4	0.40	0.40
5	**1.00**	**1.00**
6	0.80	0.90
7	**0.10**	**0.30**
8	**0.20**	**0.10**
9	0.00	0.50
10	**0.08**	**0.90**
Average	0.318	0.5
	Difference	0.182

at least 0.07?

iteration = 2 counter = 1

Fisher's Randomization Test

Query	System A	System B
1	**0.50**	**0.20**
2	0.30	0.30
3	**0.10**	**0.10**
4	**0.40**	**0.40**
5	1.00	1.00
6	**0.90**	**0.80**
7	0.30	0.10
8	0.10	0.20
9	**0.50**	**0.00**
10	0.90	0.80
Average	0.5	0.39
	Difference	-0.11

at least 0.07?

iteration = 100,000 counter = 25,678

Fisher's Randomization Test

procedure

- **Inputs:** counter = 0, N = 100,000
- Repeat N times:

  Step 1: for each query, 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

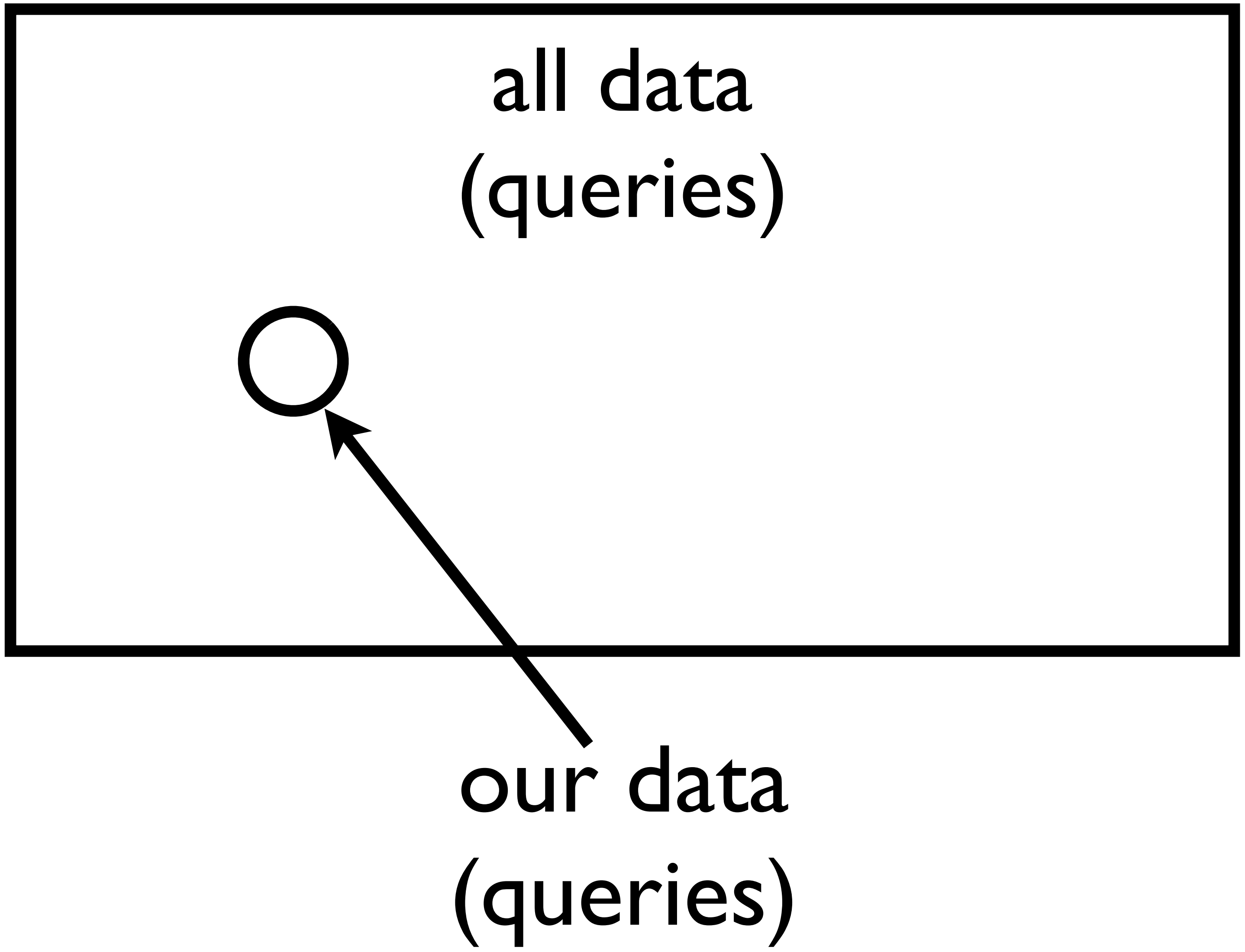
procedure

- 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.
- We cannot claim that System B is better than System A.

Bootstrap-Shift Test

motivation

- Our sample is a representative sample of all data



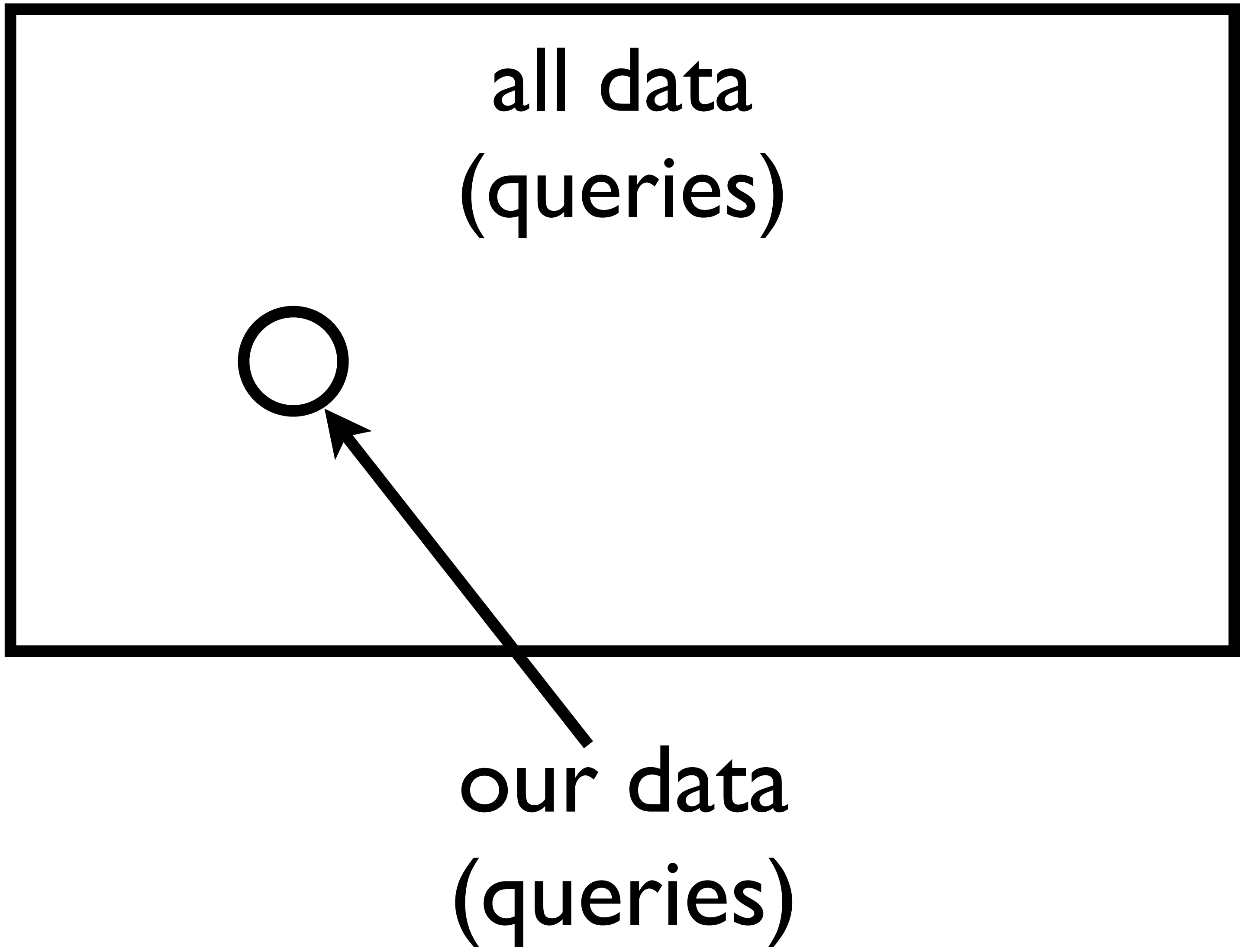
all data
(queries)

our data
(queries)

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 queries (with replacement) from our set of 10 queries (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

Query	System A	System B
1	0.20	0.50
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
Average	0.41	0.48
	Difference	0.07

Bootstrap-Shift Test

Query	System A	System B	**sample**
1	0.20	0.50	**0**
2	0.30	0.30	**1**
3	0.10	0.10	**2**
4	0.40	0.40	**2**
5	1.00	1.00	**0**
6	0.80	0.90	**1**
7	0.30	0.10	**1**
8	0.10	0.20	**1**
9	0.00	0.50	**2**
10	0.90	0.80	**0**

iteration = 1

Bootstrap-Shift Test

Query	System A	System B
2	0.30	0.30
3	0.10	0.10
3	0.10	0.10
4	0.40	0.40
4	0.40	0.40
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
9	0.00	0.50
Average	0.25	0.35
	Difference	**0.1**

T = {**0.10**}

iteration = 1

Bootstrap-Shift Test

Query	System A	System B	**sample**
1	0.20	0.50	**0**
2	0.30	0.30	**0**
3	0.10	0.10	**3**
4	0.40	0.40	**2**
5	1.00	1.00	**0**
6	0.80	0.90	**1**
7	0.30	0.10	**1**
8	0.10	0.20	**1**
9	0.00	0.50	**1**
10	0.90	0.80	**1**

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

iteration = 2

Bootstrap-Shift Test

Query	System A	System B
3	0.10	0.10
3	0.10	0.10
3	0.10	0.10
4	0.40	0.40
4	0.40	0.40
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
Average	0.32	0.36
	Difference	**0.04**

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

iteration = 2

Bootstrap-Shift Test

Query	System A	System B
1	0.20	0.50
1	0.20	0.50
4	0.40	0.40
4	0.40	0.40
4	0.40	0.40
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
8	0.10	0.20
10	0.90	0.80
Average	0.38	0.44
	Difference	**0.06**

T = {**0.10**,
0.04,
……,
0.06}

iteration = 100,000

Bootstrap-Shift Test

procedure

- **Inputs:** Array T = {}, N = 100,000
- Repeat N times:

  Step 1: sample 10 queriesqueries (with replacement) from our set of 10 queries (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

Average = **0.12**

Step 4

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

Bootstrap-Shift Test

procedure

- **Inputs:** Array T = {}, N = 100,000
- Repeat N times:

  Step 1: sample 10 queries (with replacement) from our set of 10 queries (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) = **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

Average = **0.12**

Step 4

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

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 tests make no assumptions, are robust, and easy to understand

Outline

Test-collection evaluation (review)

Significance tests

Parameter Tuning

Cross-validation

Comparing Systems

parameter tuning

Query	System A	System B
1	0.20	0.50
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
Average	0.41	0.48
	Difference	0.07

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
- The goal is to set these parameter values such that we maximize performance
- We need to do this for both systems, not just the one we want to win!
- Can you think of some example parameters?

Parameter Tuning

- Query-likelihood model with linear interpolation

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

- Parameter λ avoids zero probabilities when a document is missing a query-term
- How should we determine the value of λ ?

Parameter Tuning

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

Parameter Tuning

λ =	Average P@10 =
0.0	0.25
0.1	0.27
0.2	0.29
0.3	0.35
0.4	0.45
0.5	0.50
0.6	**0.55**
0.7	0.47
0.8	0.35
0.9	0.20
1.0	0.00

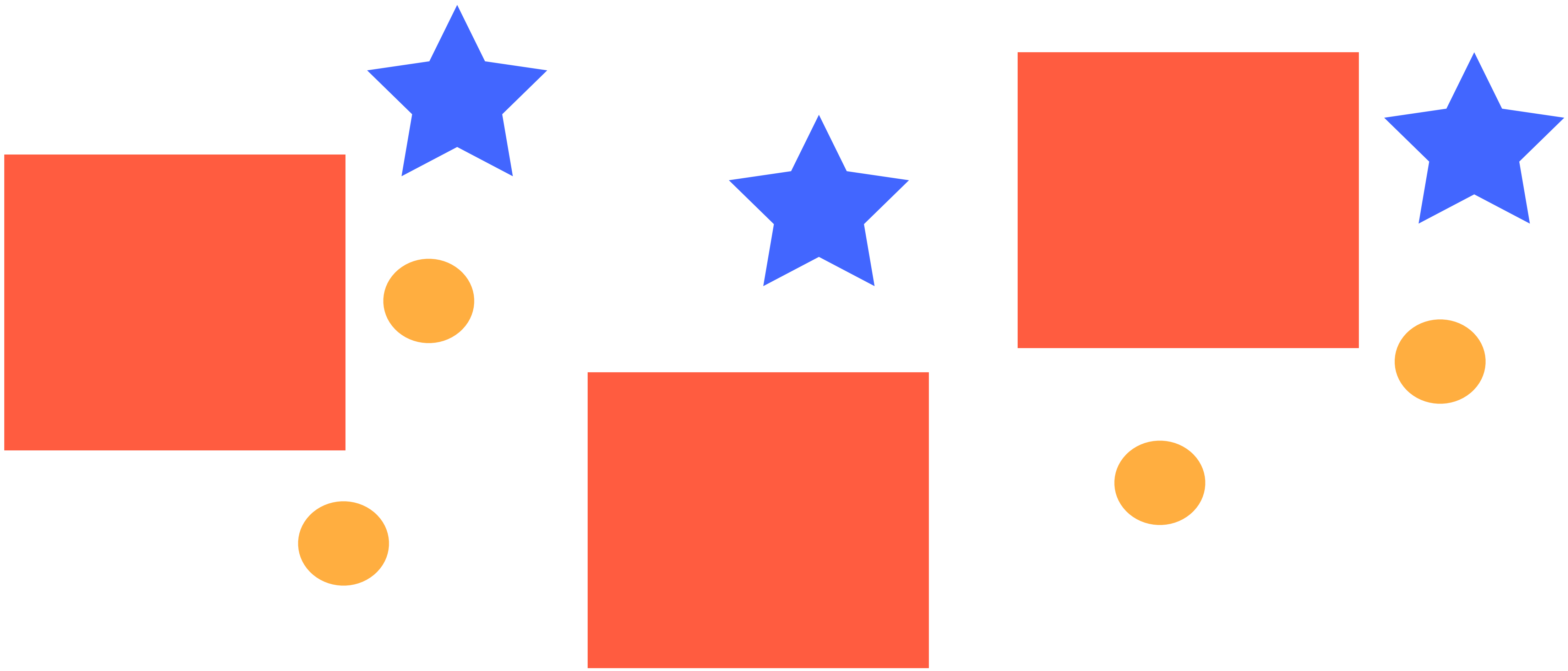
50
TEST
QUERIES

Why is this a bad idea?

Parameter Tuning

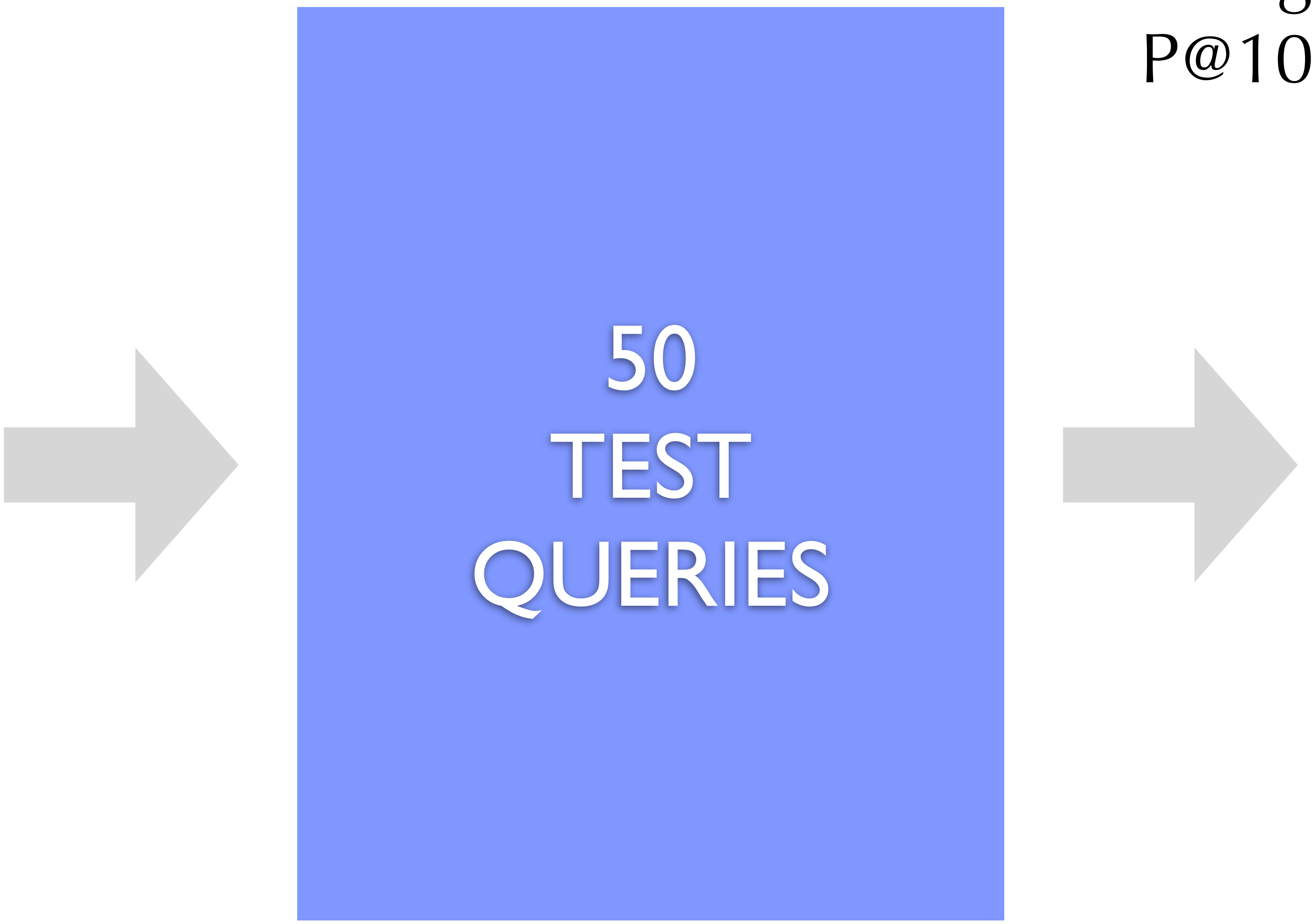
toy example

- Objective: distinguish between stars, squares, and circles



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

Parameter Tuning



50
TEST
QUERIES

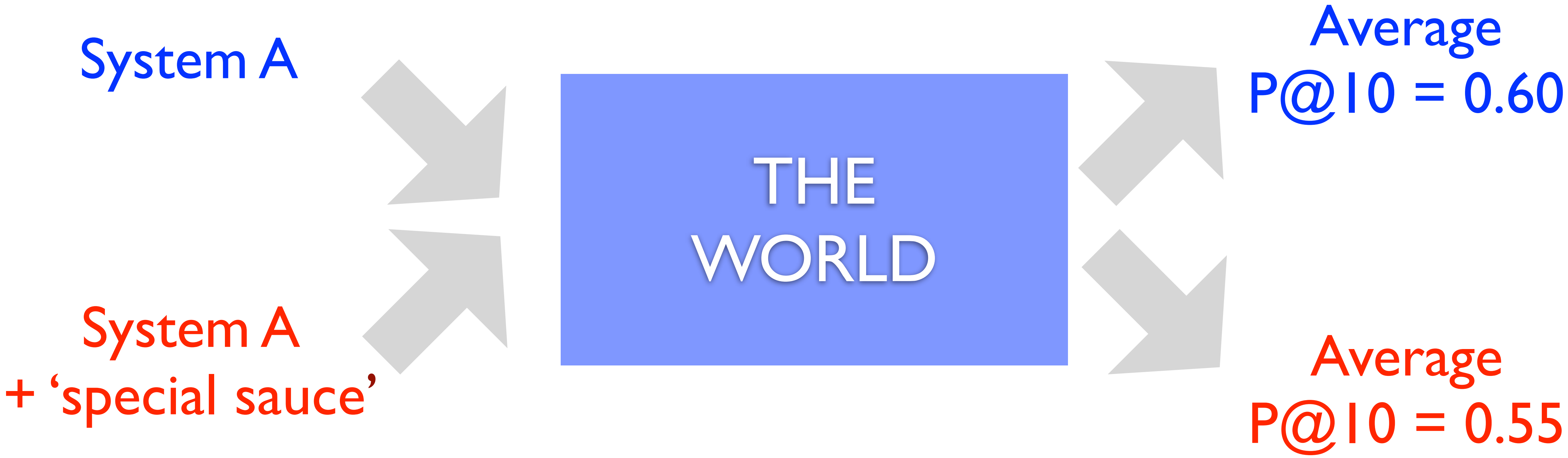
λ =	Average P@10 =
0.0	0.25
0.1	0.27
0.2	0.29
0.3	0.35
0.4	0.45
0.5	0.50
0.6	**0.55**
0.7	0.47
0.8	0.35
0.9	0.20
1.0	0.00

Why is this a bad idea?

Parameter Tuning

- The goal is to set parameter values such that we maximize performance
- What is the performance that we are really interested in?
- We care about 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 are meaningful for the overall population!

Parameter Tuning



System A

System A
+ ‘special sauce’

THE
WORLD

Average
P@10 = 0.60

Average
P@10 = 0.55

Parameter Tuning

- Option 2:

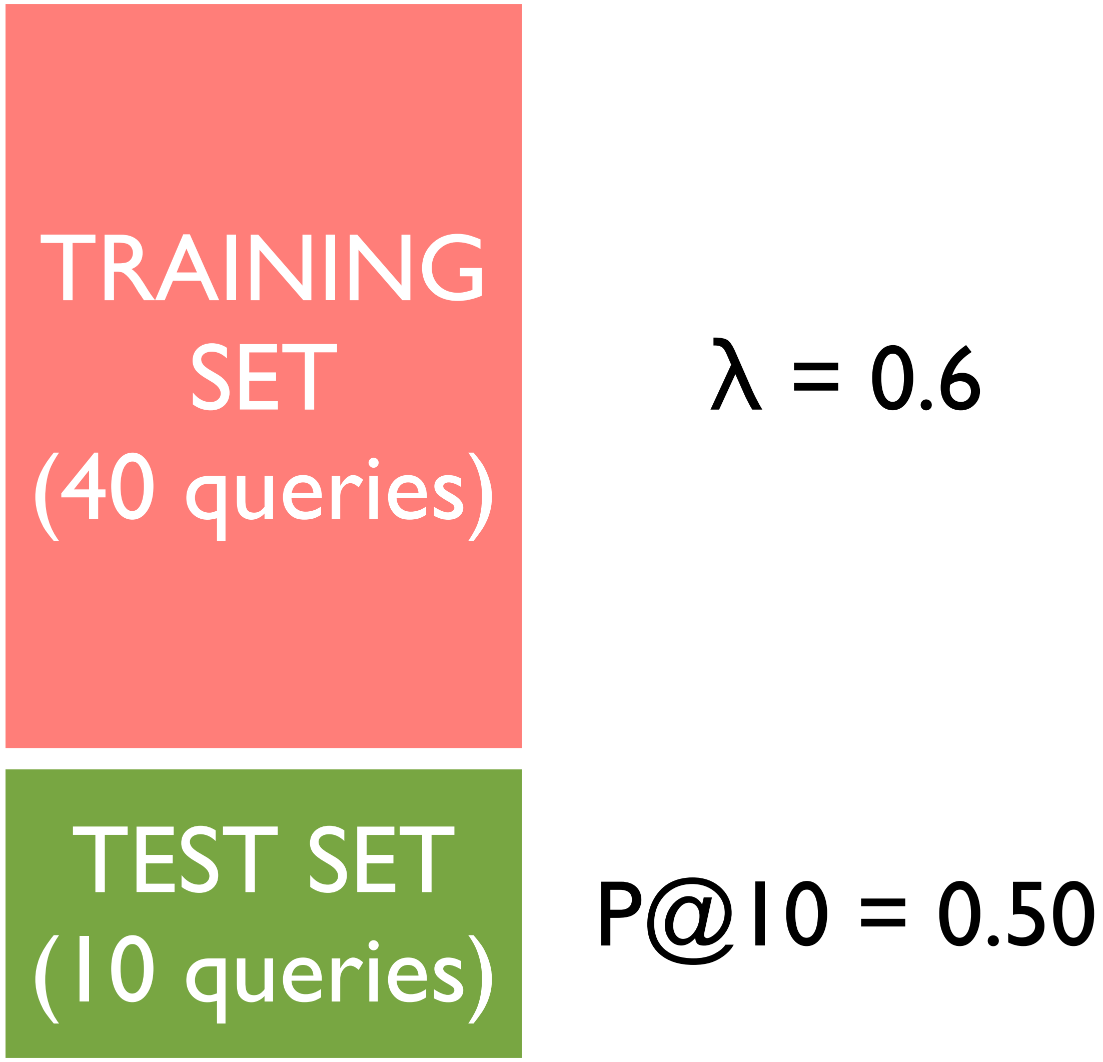
1. divide the set of 50 queries into two sets:
    - training set: a set of queries used to find the best parameter values (e.g., 40 queries)
    - test set: a held-out set used to evaluate model performance (e.g., 10 queries)
2. train: find the parameter value that maximize performance on the training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



TRAINING SET (40 queries)

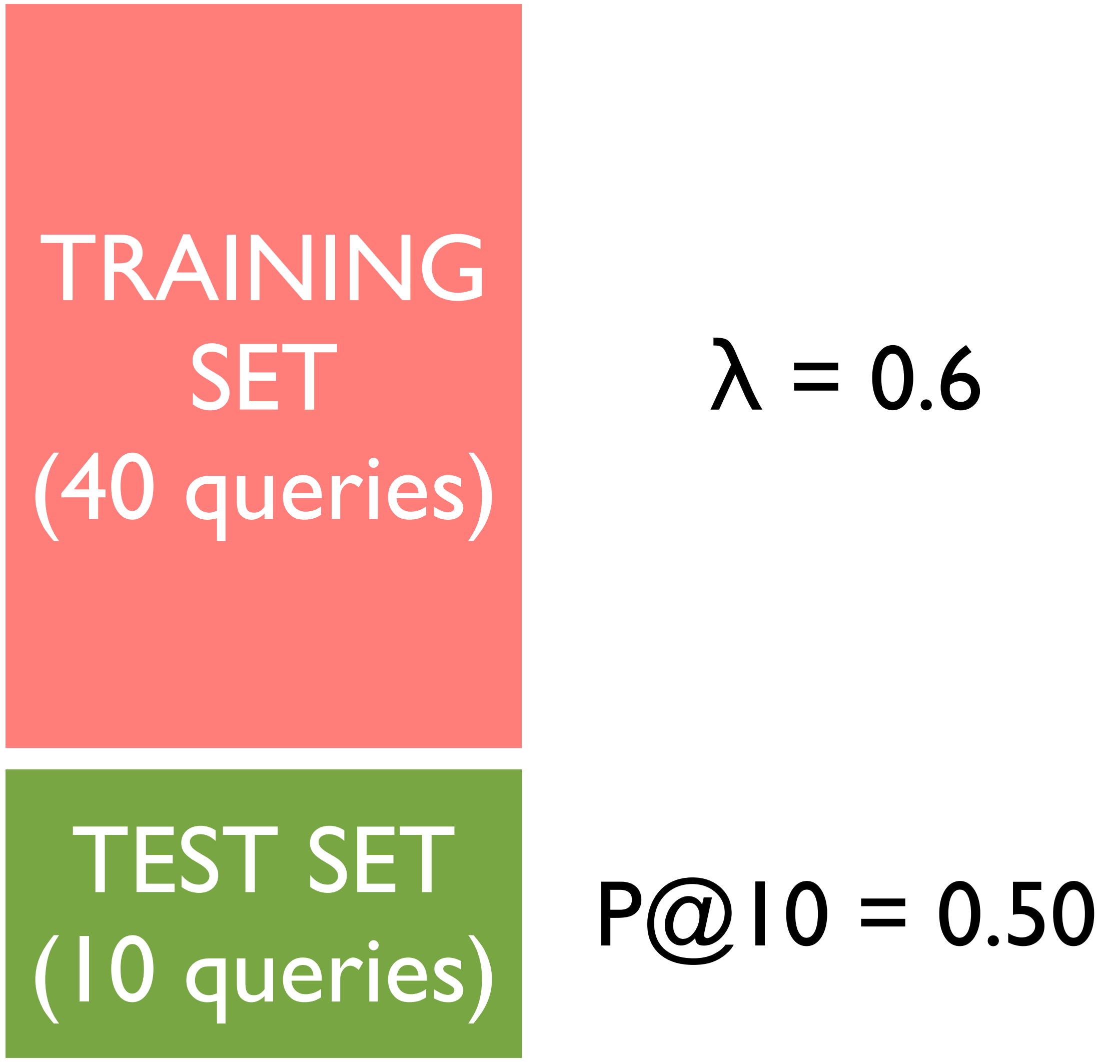
$\lambda = 0.6$

TEST SET (10 queries)

P@10 = 0.50

Parameter Tuning

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



TRAINING SET (40 queries)

λ = 0.6

TEST SET (10 queries)

P@10 = 0.50

Advantages and Disadvantages?

Single Train/Test Split

- Advantage
  - the data used to find the optimal parameter values 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
  - we may be punishing ourselves unnecessarily

Parameter Tuning

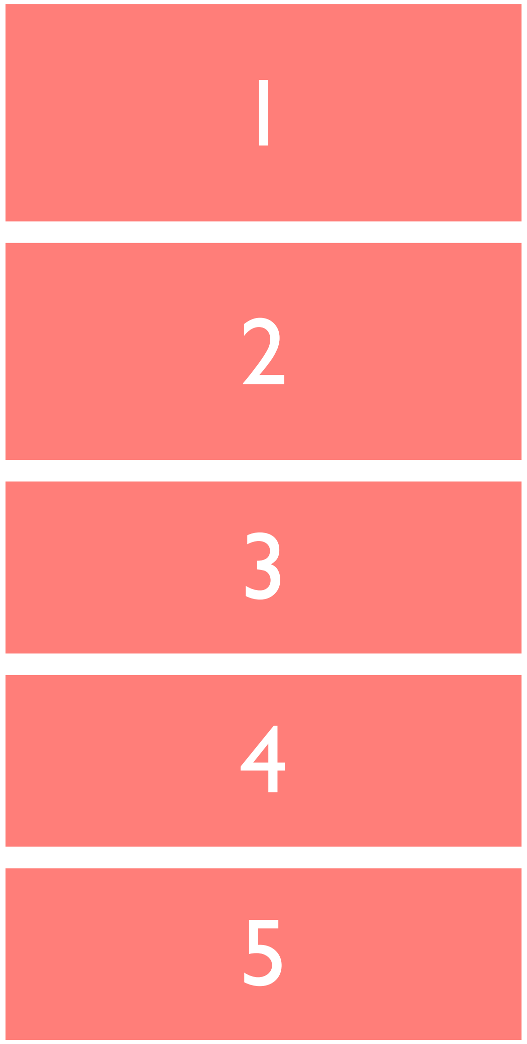
- Option 3: cross-validation
  1. divide the set of 50 queries into N sets of 50/N queries
  2. use the union of N-1 sets to find the best parameter values
  3. measure performance (using the best parameters) on the held-out set
  4. do steps 2-3 N times
  5. average performance across the N held-out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



1

2

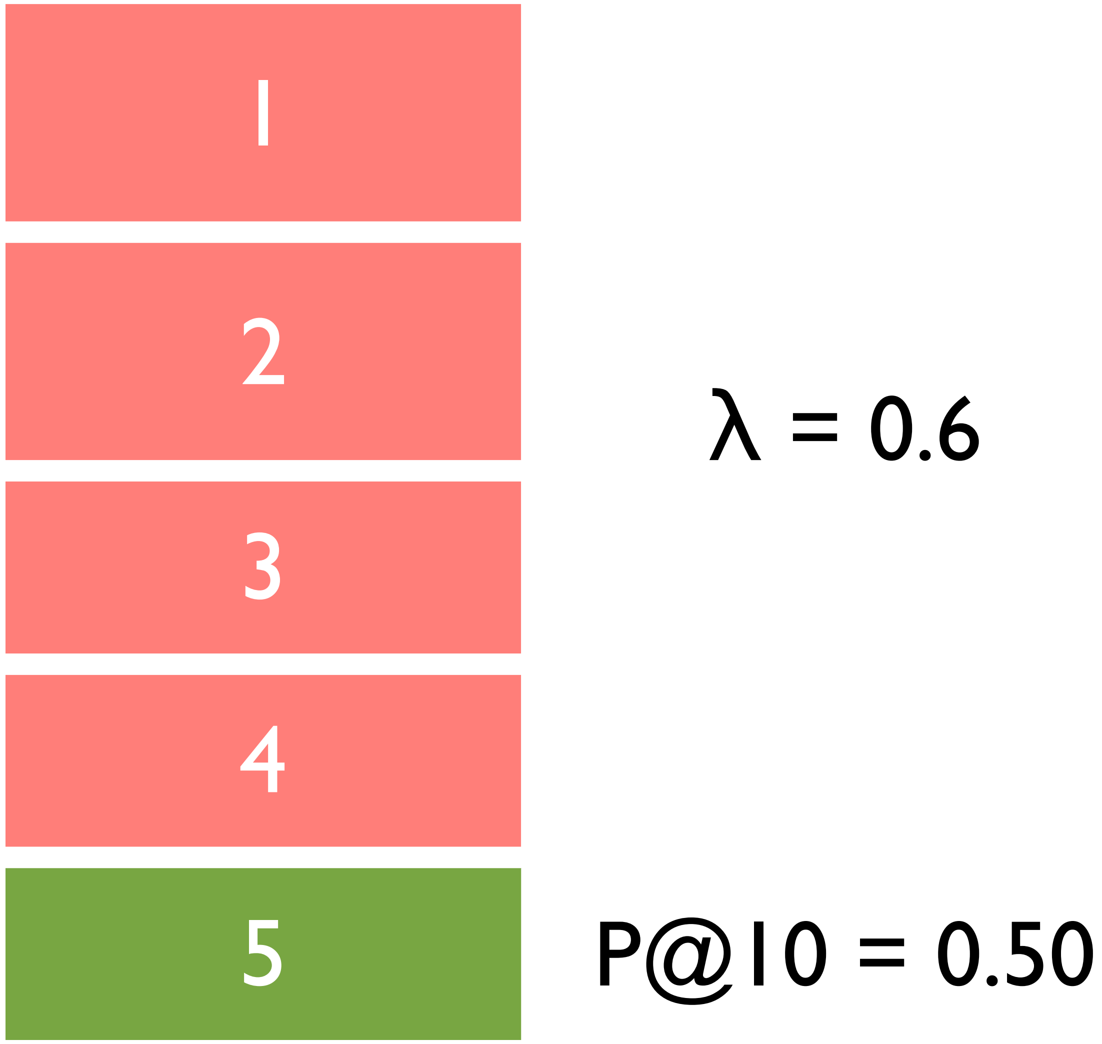
3

4

5

Cross-Validation

- For each fold, find the parameter value that maximize performance on the union of N - 1 folds and test (using this parameter value) on the held-out fold.



1

2

3

4

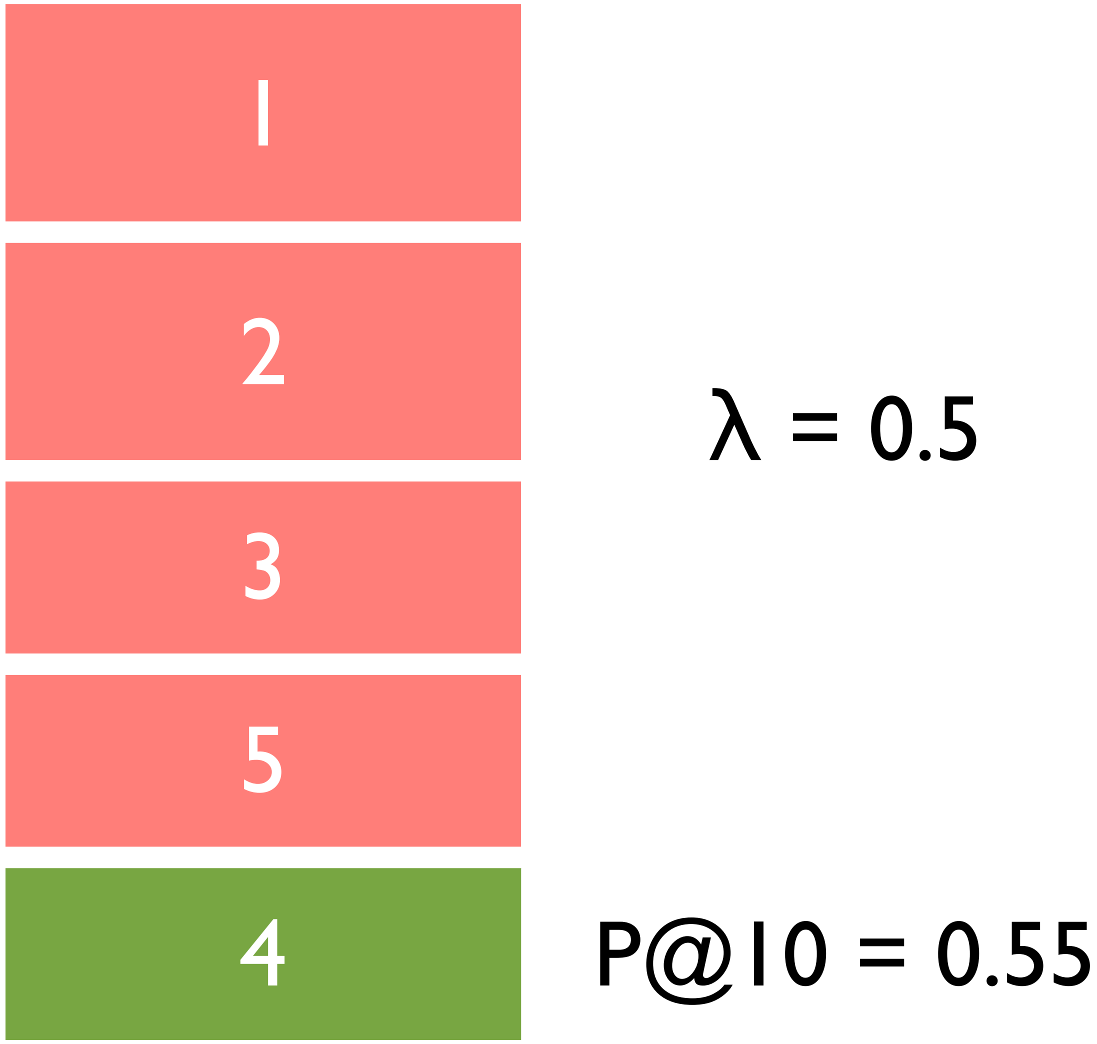
5

$\lambda = 0.6$

P@10 = 0.50

Cross-Validation

- For each fold, find the parameter value that maximize performance on the union of N - 1 folds and test (using this parameter value) on the held-out fold.



1

2

3

5

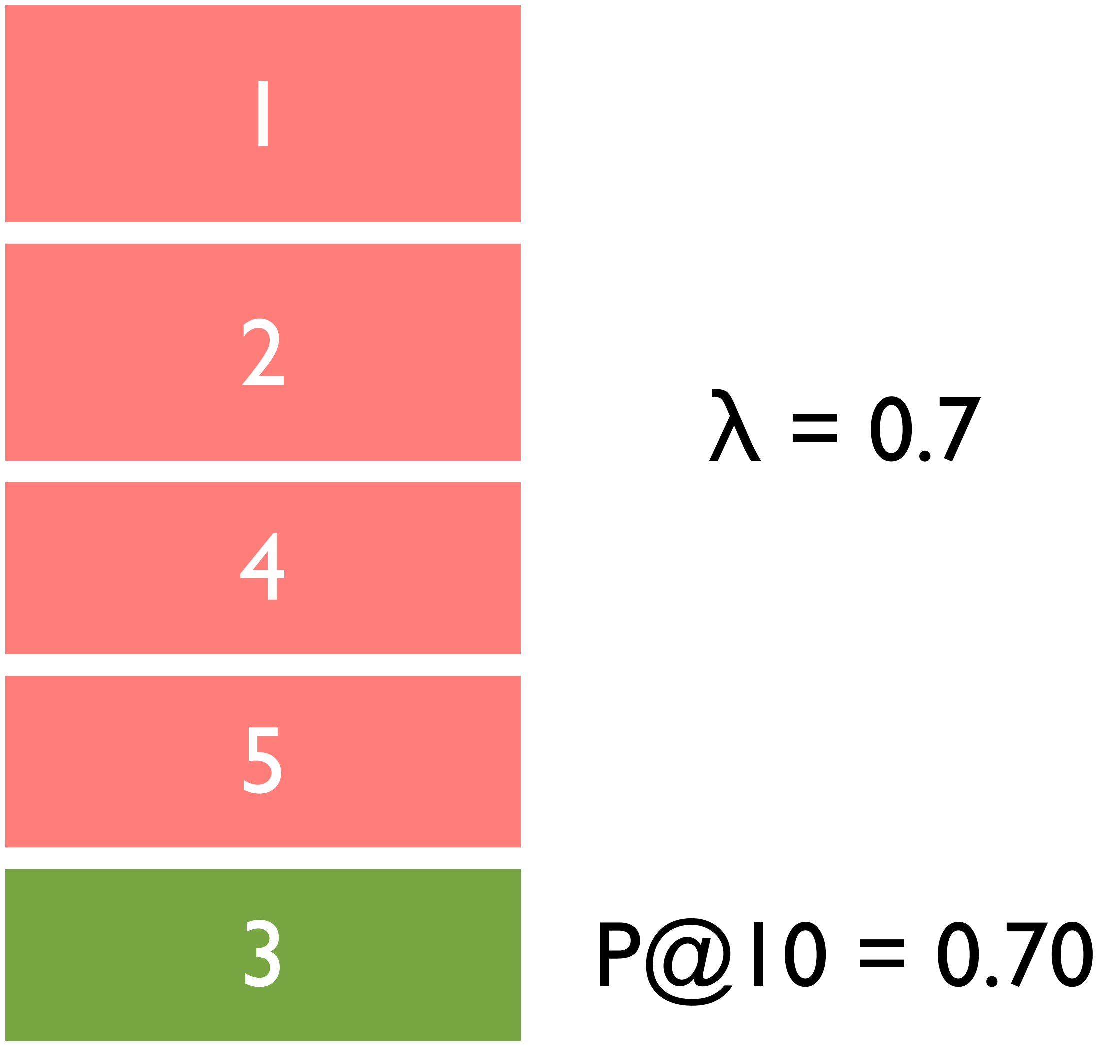
4

λ = 0.5

P@10 = 0.55

Cross-Validation

- For each fold, find the parameter value that maximize performance on the union of N - 1 folds and test (using this parameter value) on the held-out fold.



1

2

4

5

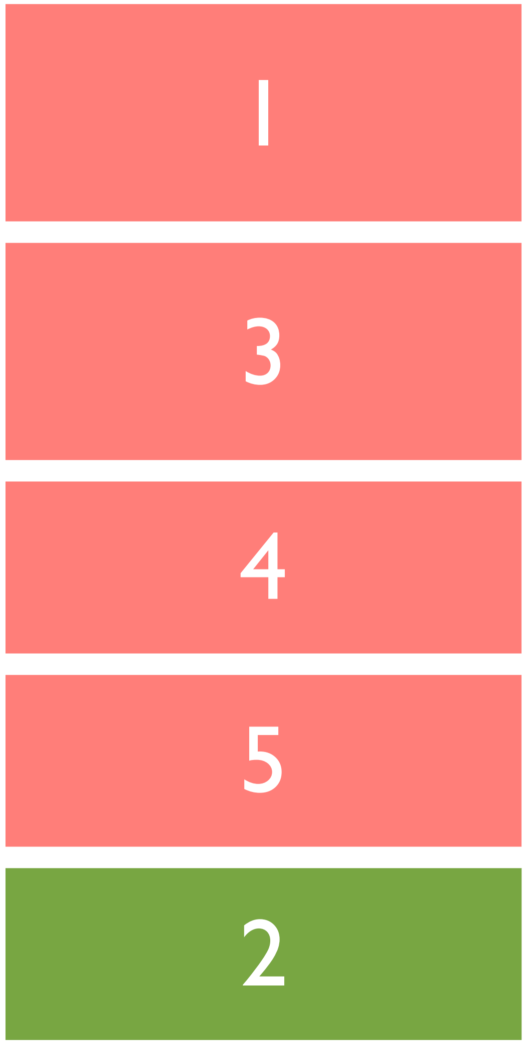
3

λ = 0.7

P@10 = 0.70

Cross-Validation

- For each fold, find the parameter values that maximize performance on the union of N - 1 folds and test (using these parameter values) on the held-out fold.



1

3

4

5

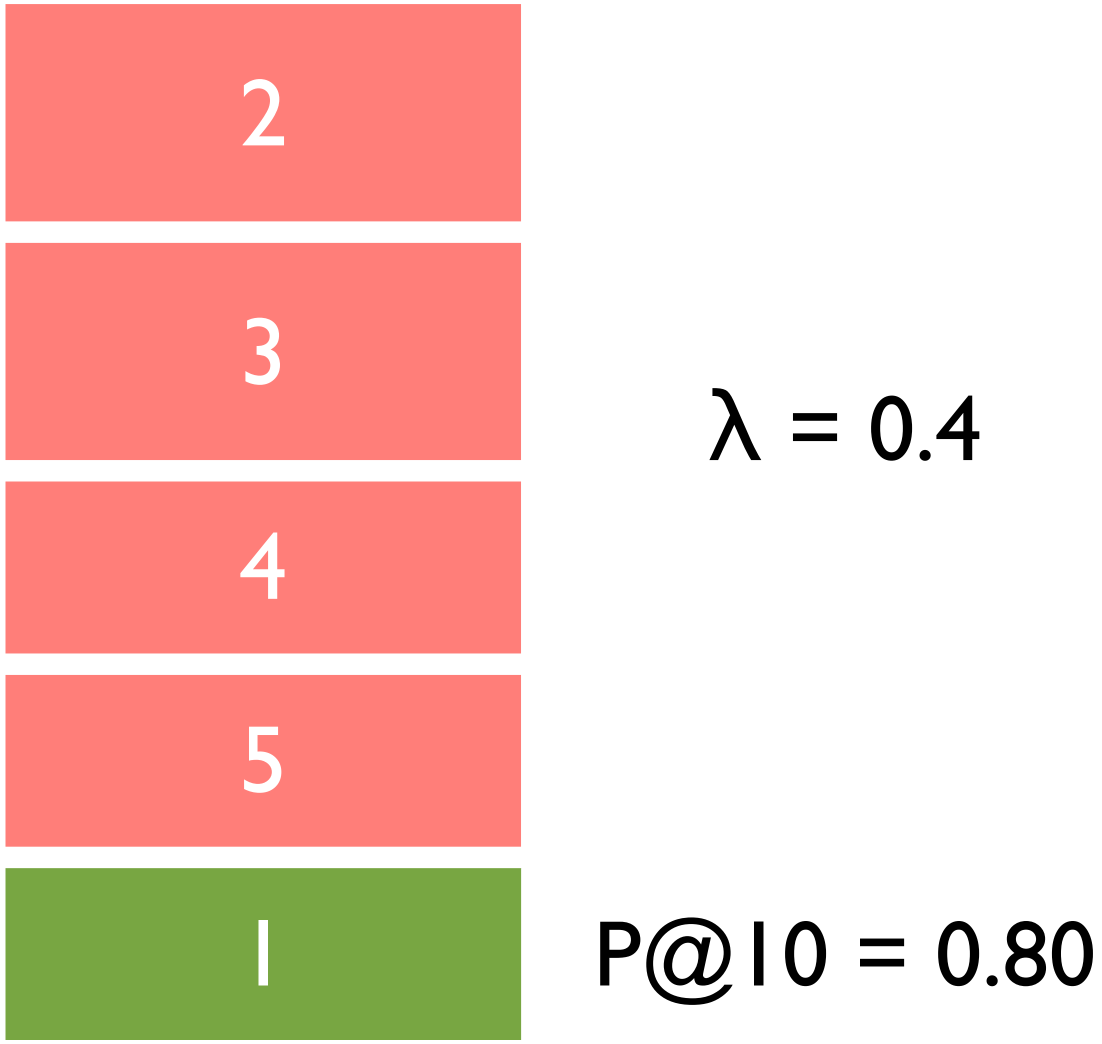
2

λ = 0.6

P@10 = 0.50

Cross-Validation

- For each fold, find the parameter value that maximize performance on the union of N - 1 folds and test (using this parameter value) on the held-out fold.



2

3

4

5

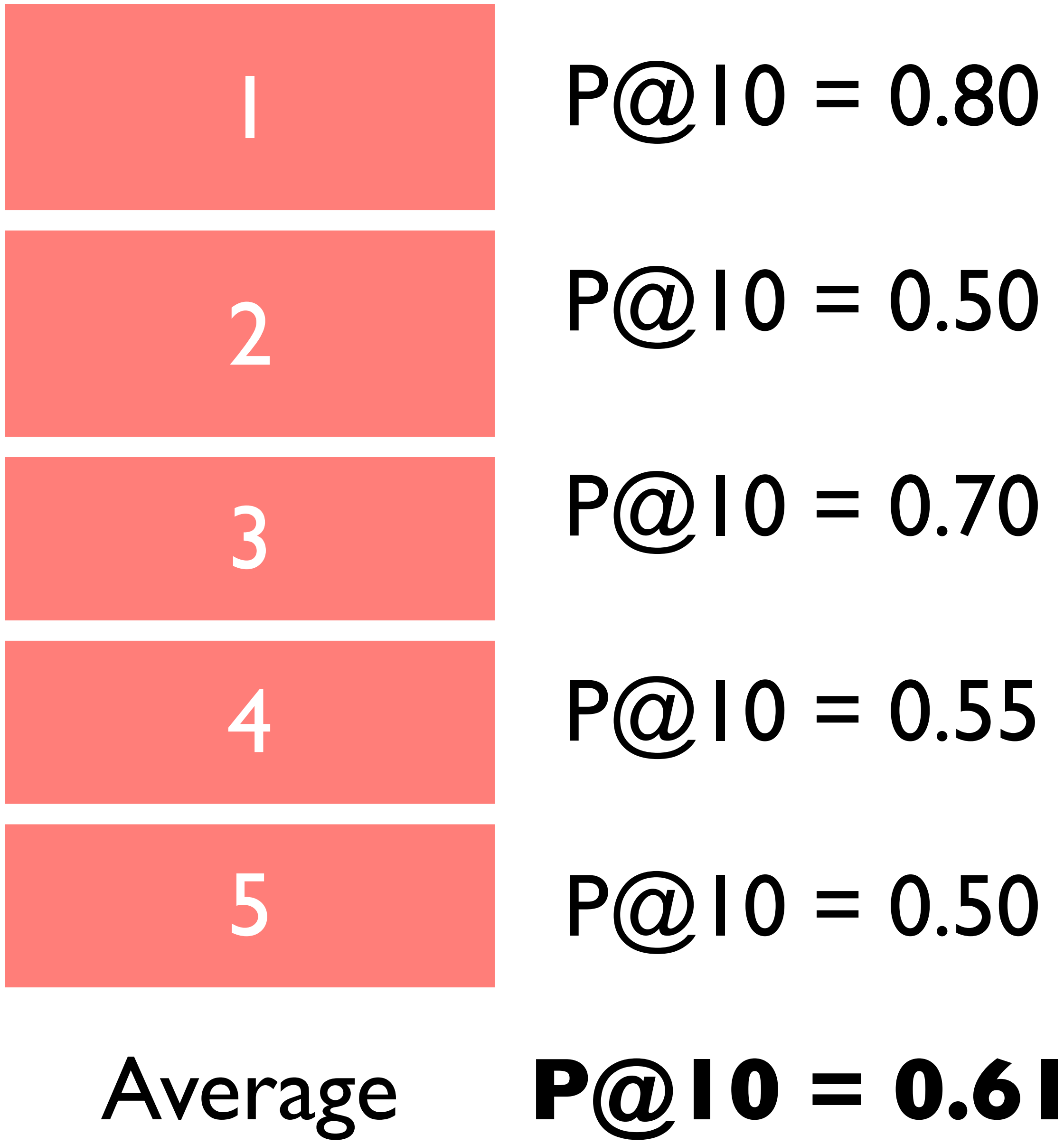
1

λ = 0.4

P@10 = 0.80

Cross-Validation

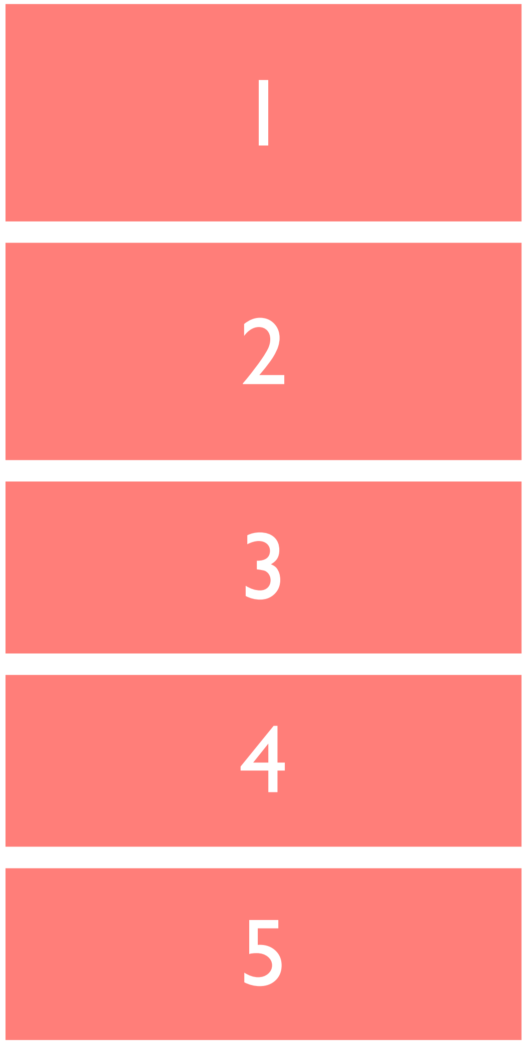
- Average the performance across held-out folds



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
Average	**P@10 = 0.61**

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
Average	**P@10 = 0.61**

Advantages and Disadvantages?

N-Fold Cross-Validation

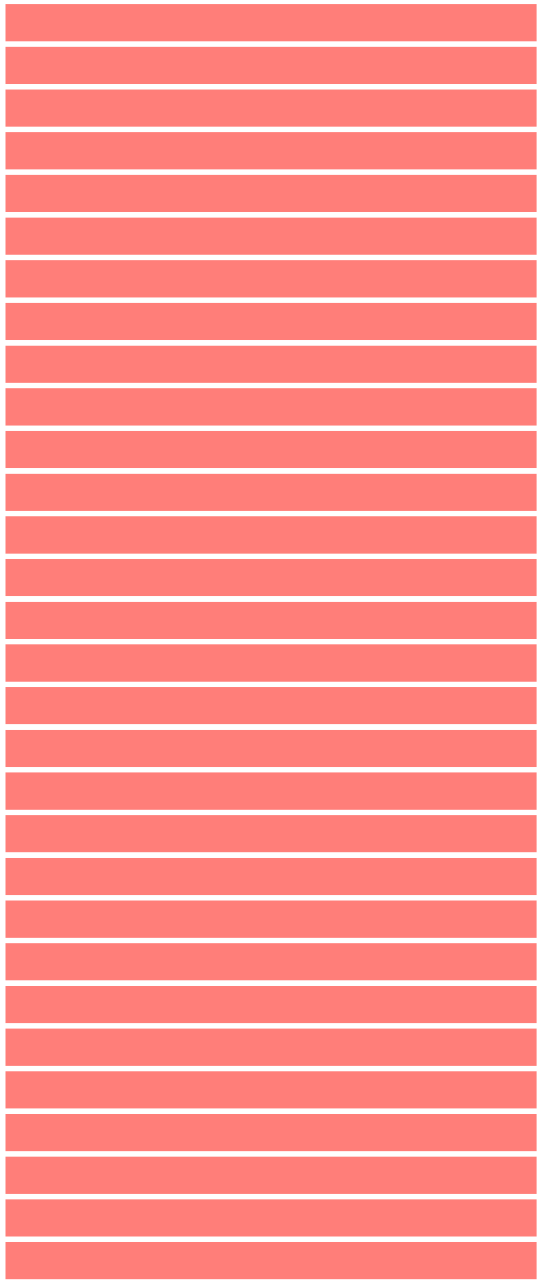
- Advantage
    - multiple rounds of generalization performance.
- Disadvantage
    - ultimately, we'll tune parameters on the set of 50 queries and send our system into the world.
    - a model trained on 50 queries should perform better than one trained on 40.
    - thus, we may be underestimating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

Leave-One-Out Cross-Validation

DATASET
(50 queries)

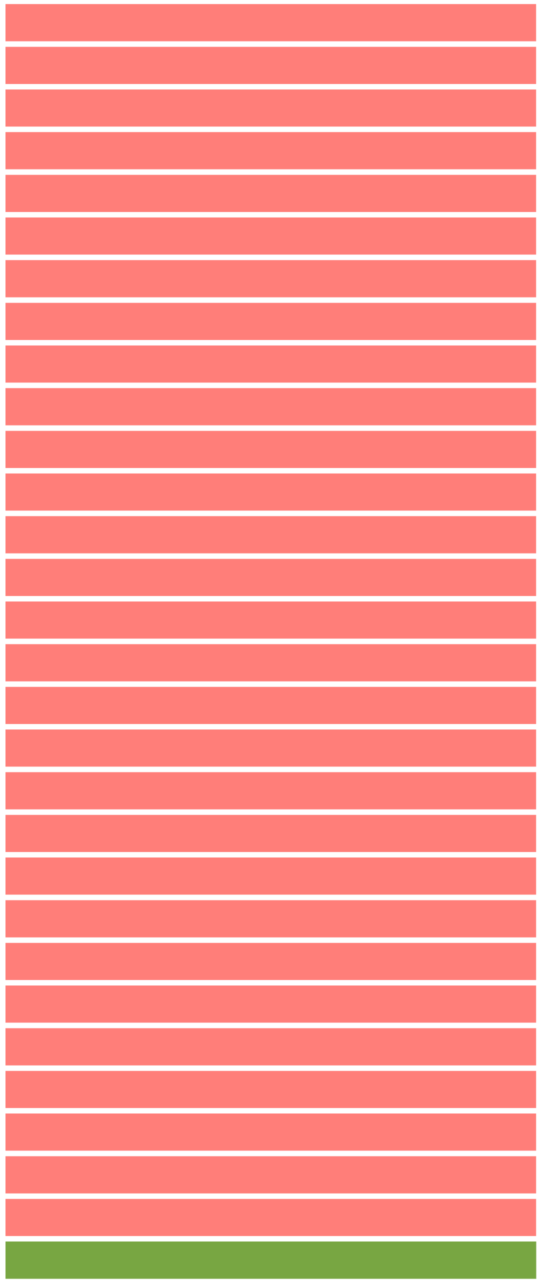
Leave-One-Out Cross-Validation

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



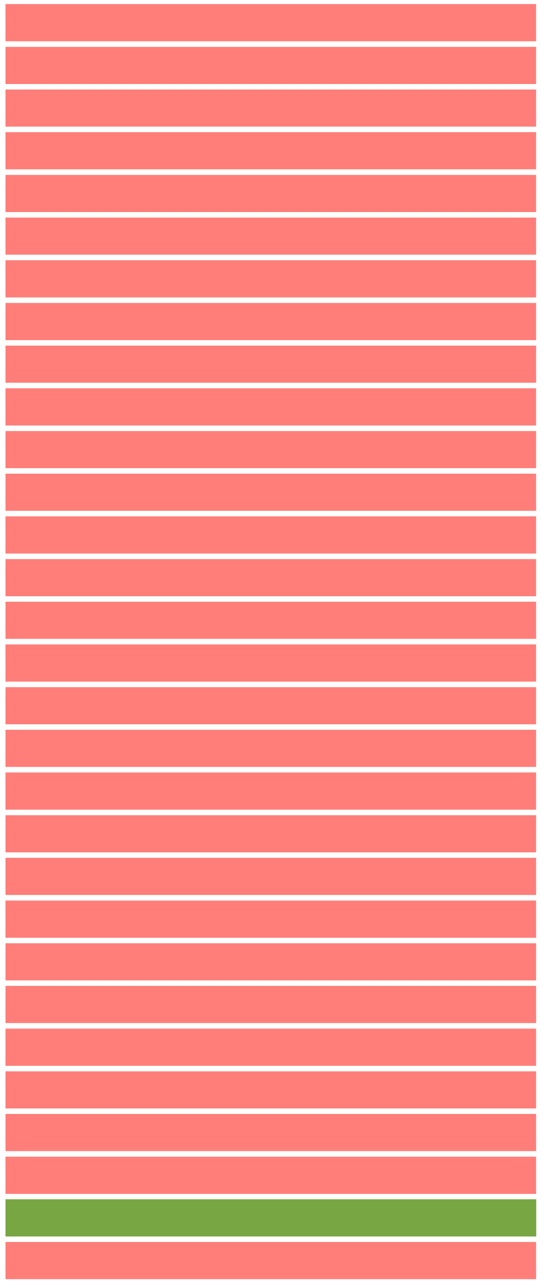
Leave-One-Out Cross-Validation

- 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.



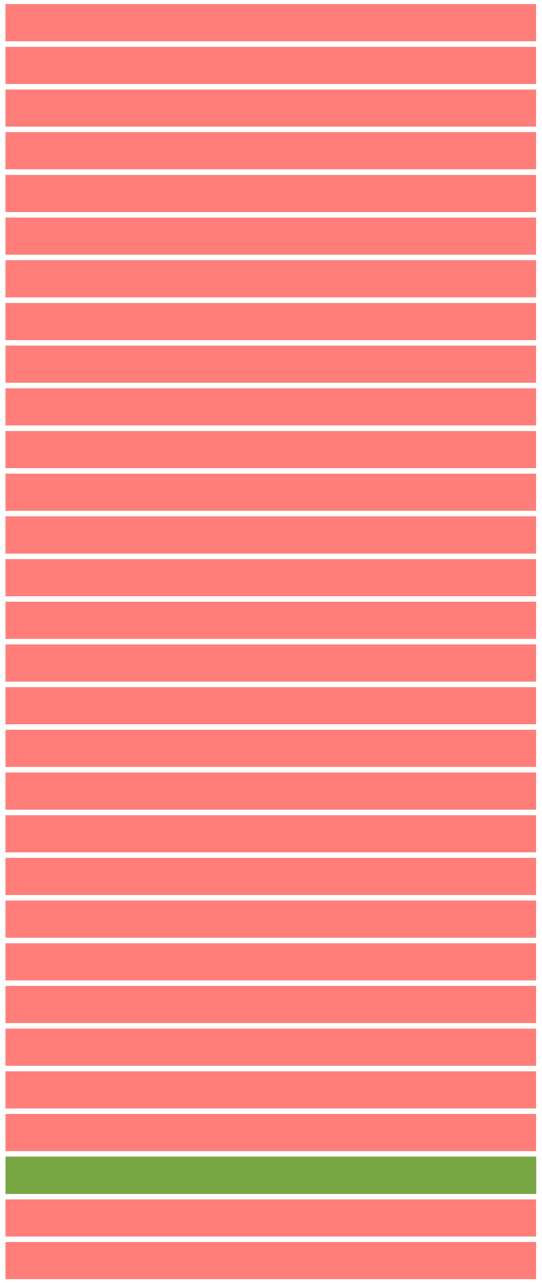
Leave-One-Out Cross-Validation

- 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

- 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.

Putting it all Together

- 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 folds using a significance test

Fold	System A	System B
1	0.20	0.50
2	0.30	0.30
3	0.10	0.10
4	0.40	0.40
5	1.00	1.00
6	0.80	0.90
7	0.30	0.10
8	0.10	0.20
9	0.00	0.50
10	0.90	0.80
Average	0.41	0.48
	Difference	0.07

The Annoying Details

lots of experiments

- A model with three parameters (each with a range between 0.0 and 1.0) has 10 x 10 x 10 = 1000 parameter combinations
- With 10-fold cross-validation, that's (1000 x 10) + (1000 x 10) = 110,000 batch evaluation cycles per system
- You can't do this on your iPad.

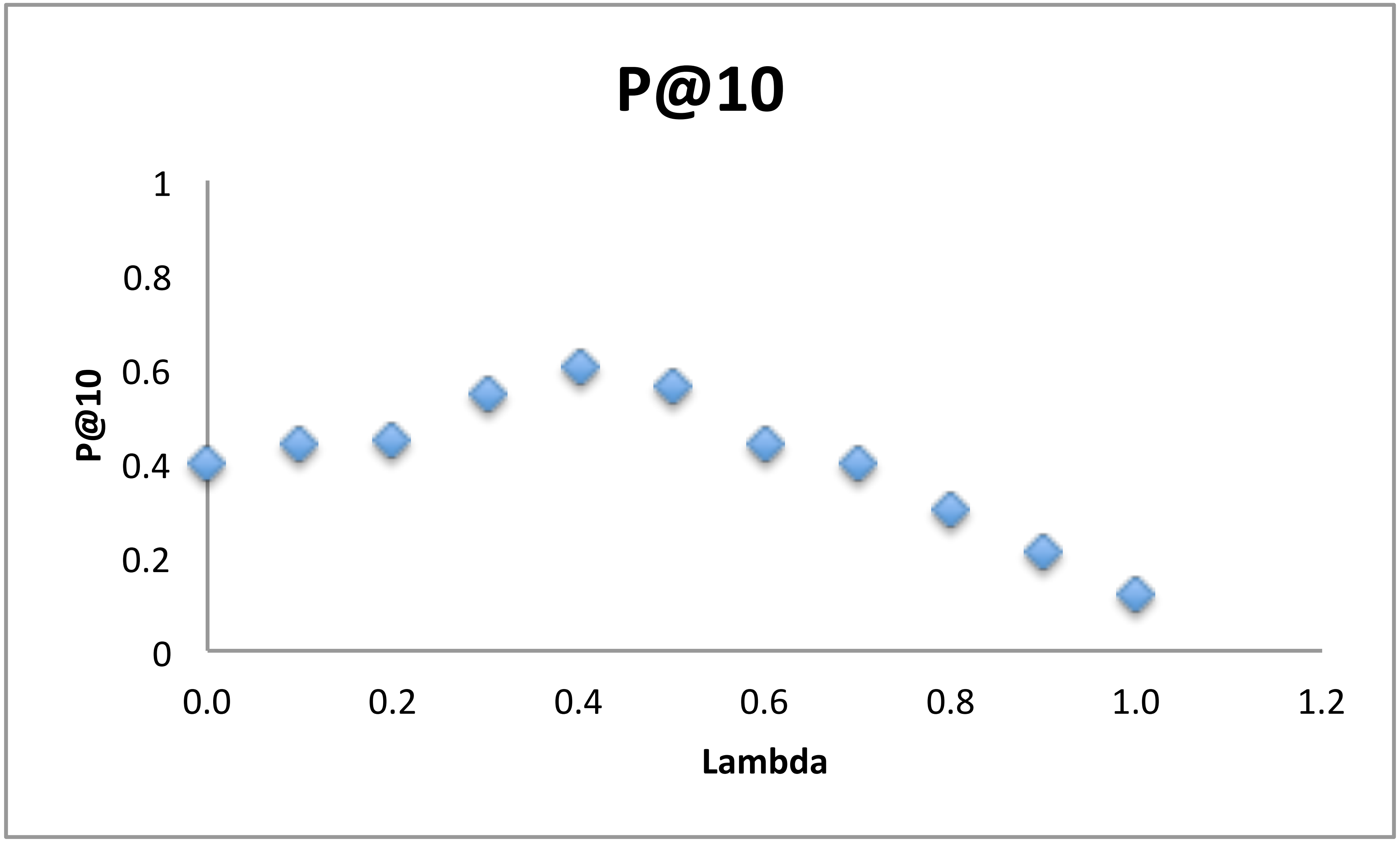
The Annoying Details

resisting temptation

- If your goal is to outperform some baseline system, it can be tempting to not tune the baseline.
- You need to be thorough
- What happens if the optimal parameter value is always the same (across cross-validation folds)?
    - you may need to increase the range
    - you may need to increase the granularity

The Annoying Details

resisting temptation



- Suppose that the optimal value of Lambda for different training folds is consistently 0.40.

- Are we done?

Conclusions

- Good experimentation is based on testing generalization performance
- Statistical significance tests provide a level of confidence that one system is better in general
- Parameter tuning + cross-validation allows to estimate how the model (with free parameters) will perform on new data
- Be skeptic and thorough
- When in doubt, ask your local statistician