Advances in IR Evaluation

Ben Carterette Evangelos Kanoulas Emine Yilmaz

Course Outline

- Intro to evaluation
  - Evaluation methods, test collections, measures, comparable evaluation
- Low cost evaluation
- Advanced user models
  - Web search models, novelty & diversity, sessions
- Reliability
  - Significance tests, reusability
- Other evaluation setups

Low-Cost Evaluation (4)

- Estimating *measures* with less judgments
  - Aslam et al. SIGIR06, Yilmaz and Aslam CIKM06, Yilmaz et al SIGIR09

- Estimating systems *ranking* with less judgments
  - Carterette et al. SIGIR06, Moffat et al. SIGIR07

Goals for a Test Collection

- Different goals suggest different approaches:
  - Find the relevant documents:
    - Pooling
    - Move-to-Front pooling, Hedge
    - Interactive Searching and Judging
  - Estimate the value of an evaluation measure:
    - infAP, xinfAP, statAP
  - Compare two or more systems by some measure:
    - MTC (Minimal Test Collections)

MTC (Minimal Test Collections)

- MTC is an adaptive, episodic algorithm for deciding which documents to judge
- Its goals:
  - Accurately *compare* two or more systems
  - Make a minimum number of judgments
  - Use existing judgments to help choose
- *Not* goals of MTC:
  - Select documents most likely to be relevant
  - Find all (or even most) of the relevant documents
  - Accurate estimates of evaluation measures

MTC's Two Parts

- MTC comprises two separate parts:
  1. An algorithm for selecting documents to judge
  2. A way to evaluate when many judgments are missing
- If you "believe in" one but not the other, you may pick and choose
  - The judgments the algorithm produces can be fed into other evaluation approaches
  - The evaluation approach can be used with judgments from any other method
- They are linked in the algorithm's stopping condition

MTC Selection Algorithm Outline

- Start with the simplest case: compare two systems by some measure on one topic
- Outline of MTC algorithm:
  - Derive document weights from an algebraic expression of the difference in the measure
  - Order documents by weight and judge the highest-weighted
  - Use the judgment to update the weights
  - Continue until a stopping condition is reached

Detailed Example: Precision

- Say we want to compare two systems by precision at rank k
- First, define the difference in precision:
  - $\Delta prec@k = prec_1@k - prec_2@k$
- Goal: determine sign of Δprec@k
- Define prec1@k, prec2@k in terms of relevance:

$$prec_1@k = \frac{1}{k}\sum_{i=1}^{k} rel_{1,i}, \quad prec_2@k = \frac{1}{k}\sum_{i=1}^{k} rel_{2,i}$$

  - If we knew the values of $rel_{1,i}$, $rel_{2,i}$, we would know the sign of Δprec@k

Refining Δprec@k

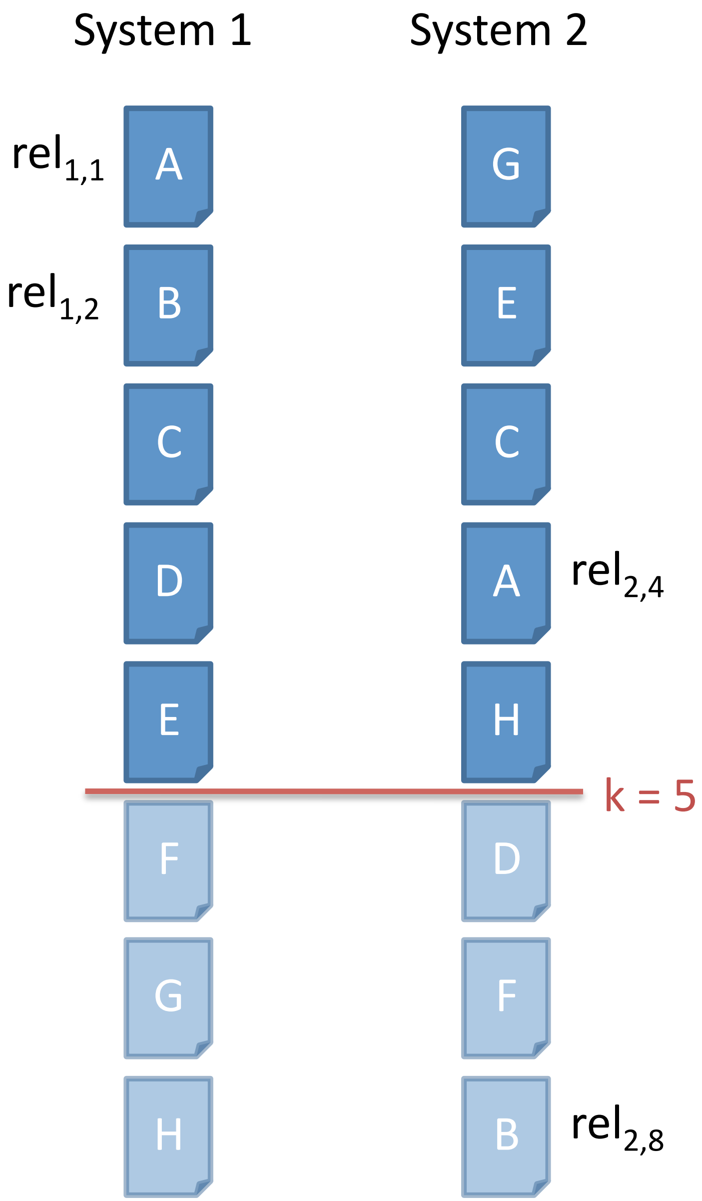
- $rel_{1,i}$, $rel_{2,i}$ could represent the same document
  - System 1 places Doc A at rank 1; system 2 places Doc A at rank 4
    - ⇒ $rel_{1,1} \equiv rel_{2,4}$
- No sense in using two different variables to refer to it
  - Number documents independently of their ranking
  - Let x_i indicate the relevance of document number i
  - Let $rank_j(i)$ indicate the rank document i appears at in system j
- Now we can write Δprec@k as:

$$\Delta prec@k = \frac{1}{k}\sum_{i=1}^{n} x_i I(rank_1(i) \leq k) - \frac{1}{k}\sum_{i=1}^{n} x_i I(rank_2(i) \leq k)$$

$$= \frac{1}{k}\sum_{i=1}^{n} x_i \left(I(rank_1(i) \leq k) - I(rank_2(i) \leq k)\right)$$

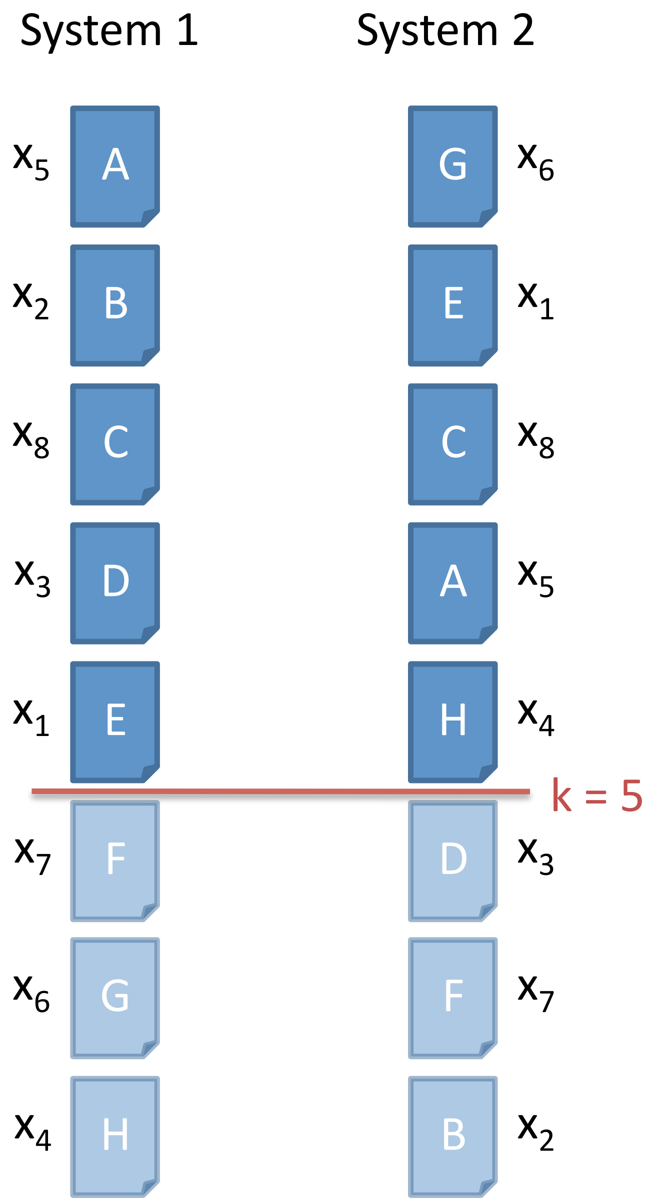
  - $I(rank_j(i) \leq k)$ is 1 if document i is ranked above k and 0 otherwise

Precision at rank 5
with local document numbering



Precision at rank 5
with global document numbering

Document numbers are independent
of rank… use rank(i) to map back to rank



$$\Delta prec@5 = \frac{1}{5}\sum_{i=1}^{n} x_i \left(I(rank_1(i) \leq 5) - I(rank_2(i) \leq 5)\right)$$

Goal of MTC

- Decide which subset of $x^n = \{x_1, x_2, \ldots, x_n\}$ to "reveal" (have judged) to prove sign of Δprec@k is -1, 0, or 1

$$\Delta prec@k = \frac{1}{k} \sum_{i=1}^{n} x_i \left(I(rank_1(i) \leq k) - I(rank_2(i) \leq k) \right)$$

- Notice:
  - Judging a document ranked below k by both systems tells us nothing
    - $I(rank_1(i) \leq k) - I(rank_2(i) \leq k) = 0 - 0 = 0$
  - Judging a document ranked above k by both systems tells us nothing
    - $I(rank_1(i) \leq k) - I(rank_2(i) \leq k) = 1 - 1 = 0$
  - The only interesting documents are those ranked above k by one system but not the other
- Define "interestingness" weight
  - $w_i = I(rank_1(i) \leq k) - I(rank_2(i) \leq k)$

Calculating document weights

$w_1 = I(rank_1(1) \leq 5) - I(rank_2(1) \leq 5)$
$= 1 - 1$
$= 0$

$w_2 = 1$

$w_3 = 1$

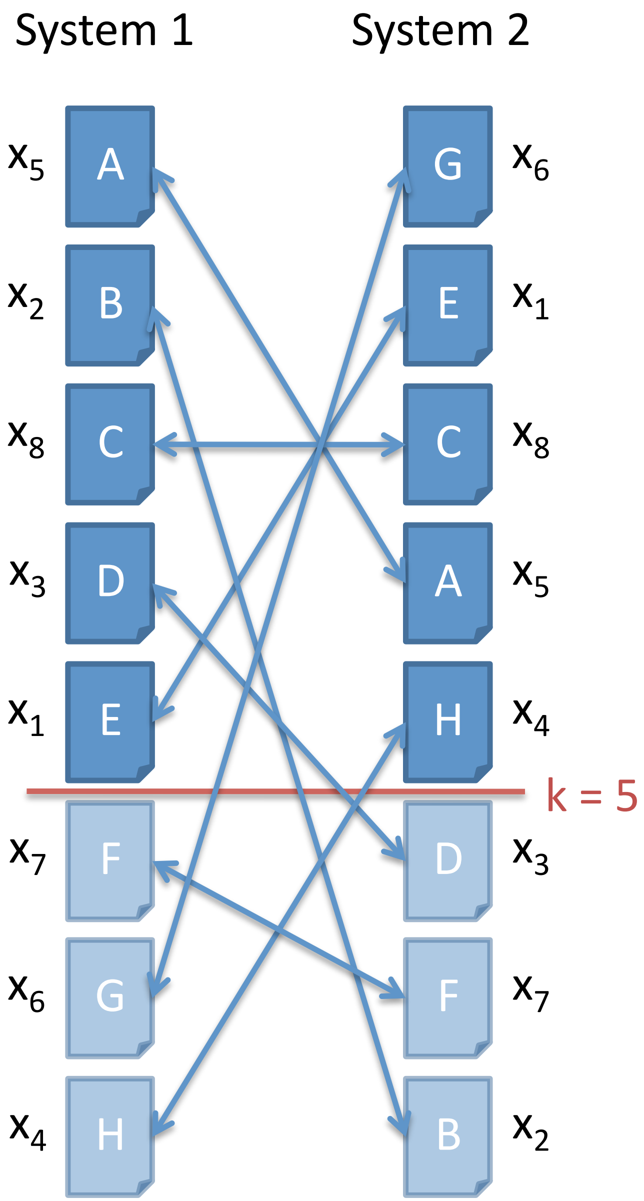
$w_4 = -1$

$w_5 = 0$

$w_6 = -1$

$w_7 = 0$

$w_8 = 0$



Only four documents are useful to judge…

Selecting Documents

- But do we need to judge *all* of the interesting documents?
- After each judgment, ask the following:
  - What is the maximum possible value of Δprec@k?
  - What is the minimum possible value of Δprec@k?
- Check these values:
  - If the maximum possible is less than zero, then we have proved that sign(Δprec@k) = -1; no more judging is necessary
  - If the minimum possible is greater than zero, we have proved that sign(Δprec@k) = 1; no more judging is necessary
  - Otherwise we must keep judging
- In other words, *bound* Δprec@k
  - Calculate lower and upper bounds by making different assumptions about the relevance of the unjudged documents

Bounding Δprecision@5

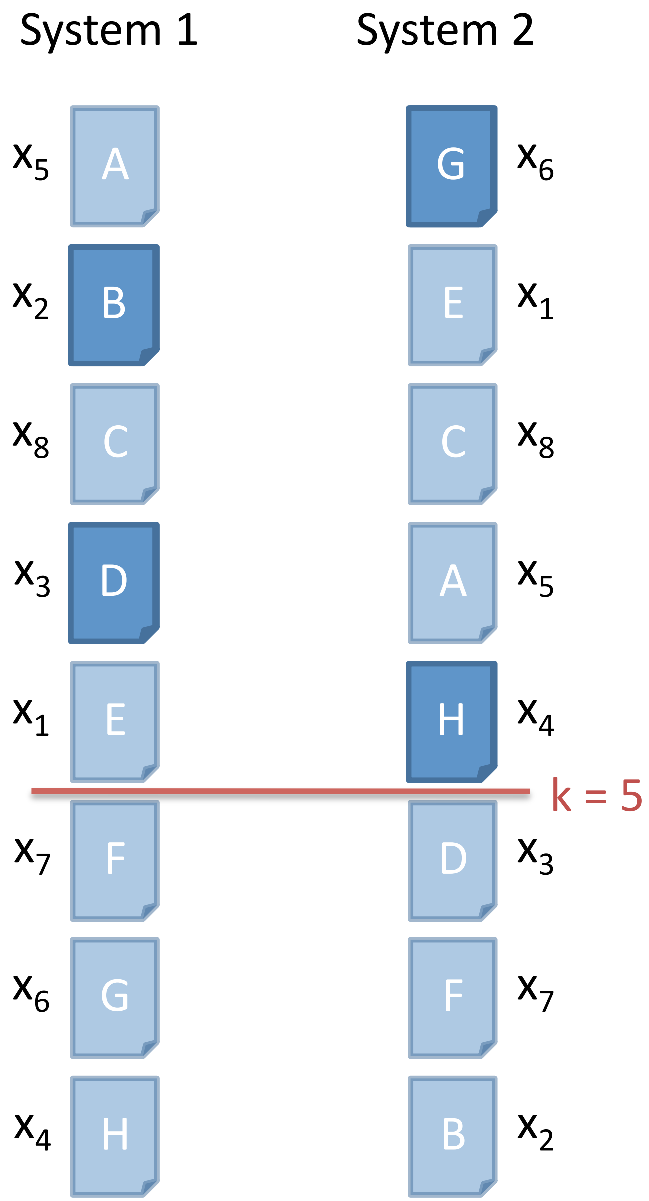
Starting from no relevance judgments:

Upper bound: B, D relevant
G, H not relevant

Lower bound: B, D not relevant
G, H relevant

-0.4 ≤ Δprec@ ≤ 0.4

We cannot conclude anything.



Bounding Δprecision@5

Suppose B and D are judged relevant. Then:

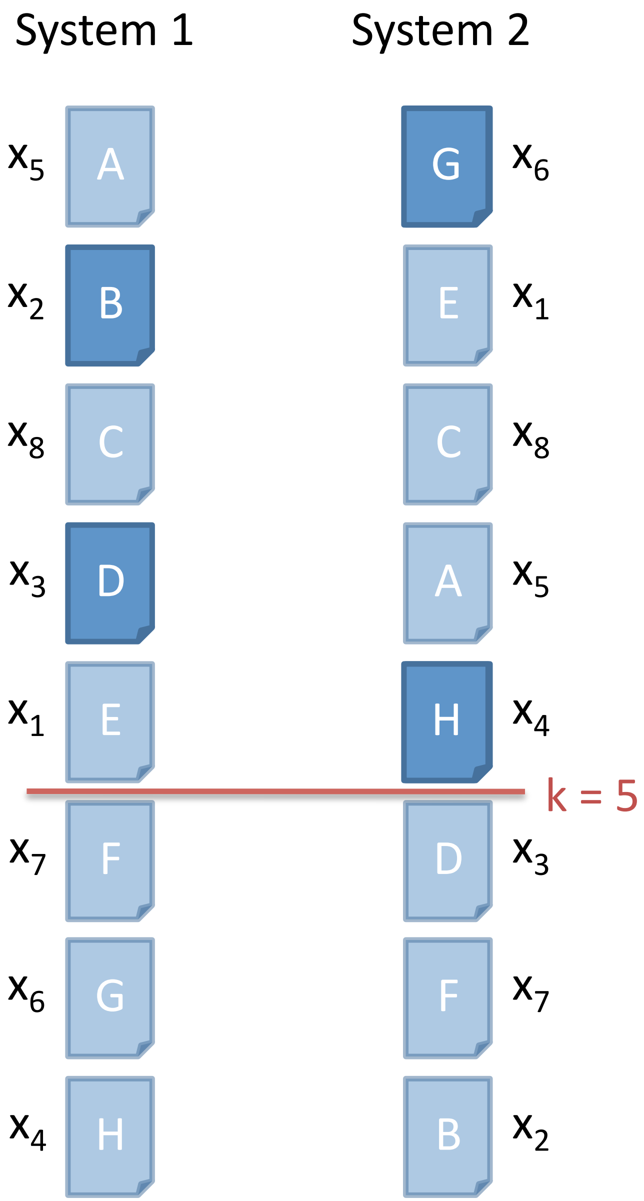
Upper bound: no more relevant
G, H not relevant

Lower bound: no more not relevant
G, H relevant

$0.0 \leq \Delta prec@ \leq 0.4$

We conclude that system 2 cannot be better than system 1.

We don't know whether system 1 is better than system 2.



Bounding Δprecision@5

Suppose B and D are judged *non*relevant.
Then:

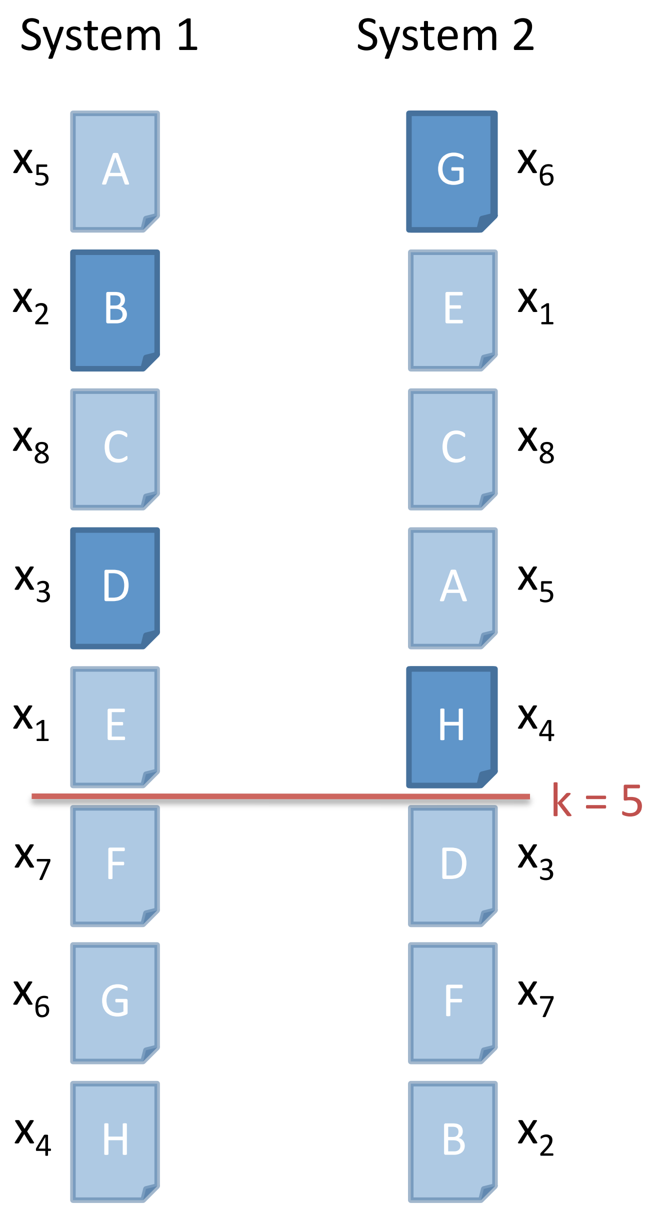
Upper bound: no more relevant
G, H not relevant

Lower bound: no more not relevant
G, H relevant

-0.4 ≤ Δprec@ ≤ 0.0

We conclude that system 1 cannot be better than system 2.

We don't know whether system 2 is better than system 1.



<u>Whether documents judged relevant or not relevant, effect on bounds is the same.</u>

Bounding Δprecision@k

- The bounds can be expressed with simple formulas:

Contribution of unjudged system 1-only documents

$$\lceil \Delta prec@k \rceil = \frac{1}{k}\left(\boxed{\sum_{i|i \text{ judged}} w_i x_i} + \boxed{\#(\text{unjudged and } w_i > 0)} \right)$$

$$\lfloor \Delta prec@k \rfloor = \frac{1}{k}\left(\boxed{\sum_{i|i \text{ judged}} w_i x_i} - \boxed{\#(\text{unjudged and } w_i < 0)} \right)$$

Contribution of unjudged system 2-only documents

Contribution of judged documents

The Algorithm (MTC for prec@k)

- for each doc i from 1 to n,
  - set $w_i = I(rank_1(i) \le k) - I(rank_2(i) \le k)$
- lowerbound = 0; upperbound = 0
- while (lowerbound ≤ 0 and upperbound ≥ 0)
  - Judge an unjudged document with $|w_i| > 0$
    - Alternate between docs with $w_i = 1$, $w_i = -1$
  - Recompute Δprec@k bounds:
    - lowerbound = $\frac{1}{k}\left(\sum_{i|i \text{ judged}} w_i x_i - \#(\text{unjudged and } w_i < 0)\right)$
    - upperbound = $\frac{1}{k}\left(\sum_{i|i \text{ judged}} w_i x_i + \#(\text{unjudged and } w_i > 0)\right)$

MTC is Minimal

- Theorem: MTC requires the minimal number of judgments to determine the sign of Δprec@k
  - More precisely: among all algorithms with no prior information about relevance, MTC requires no more judgments on average than any of them
    - Algorithms that learn something about the distribution of relevant documents (such as MTF) could do better
    - MTC could do worse on some cases while still doing better on average

MTC is Minimal: Proof Sketch

- First define two probabilities:
  - p_1 is the probability that a document unique to system 1 is judged relevant
    - i.e. the probability that a doc with $w_i > 0$ is relevant
  - p_2 is defined likewise for system 2
- If $p_1 > p_2$ then system 1 is better than system 2
  - And vice versa

MTC is Minimal: Proof Sketch

- Suppose w.l.o.g. that $p_1 > p_2$
- Suppose MTC stops after m judgments
  - At this point the lower bound is greater than zero
  - Because of alternation, m/2 of the judged documents are from system 1, m/2 from system 2
- We can place non-MTC algorithms in one of two bins:
  - Those that might judge documents with $w_i = 0$ (the majority)
  - Those that select among the same set as MTC but do not alternate ("MTC-like")

MTC is Minimal: Proof Sketch

- Suppose an alternative approach also selects m documents to judge
- If even one of those has $w_i = 0$, then the lower bound of Δprec@k cannot be greater than zero
  - At least one more judgment will be required to complete the proof
- This encompasses all non-MTC-like approaches

MTC is Minimal: Proof Sketch

- For MTC-like approaches, the argument is more difficult
- The idea is as follows:
  - Since an MTC-like approach only judges documents with $w_i \neq 0$, the only difference is that it does not alternate between $w_i > 0$ and $w_i < 0$
  - This means it prefers documents unique to system 1 or documents unique to system 2
  - Because of this preference, it may be able to prove one bound faster, but it won't be able to prove the other bound faster
  - Therefore it cannot do better than MTC

MTC for DCG@k

- DCG has become a popular measure due to its use of a user model and graded judgments
  - Gain function $g(x_i)$ maps judgments to gain values
  - Discount function d(rank(i)) discounts gains by rank
  - DCG is a family of measures with particular cases defined by specific g() and d()
- As we did with precision, define DCG in terms of relevance variables x_i and their ranks rank(i):

$$DCG@k = \sum_{i=1}^{n} \frac{g(x_i)}{d(rank(i))} I(rank(i) \leq k)$$

MTC for DCG@k

- Now we can define the difference ΔDCG@k:

$$\Delta DCG@k = \sum_{i=1}^{n} \frac{g(x_i)}{d(rank_1(i))} I(rank_1(i) \leq k) - \frac{g(x_i)}{d(rank_2(i))} I(rank_2(i) \leq k)$$
$$= \sum_{i=1}^{n} g(x_i) \left(\frac{I(rank_1(i) \leq k)}{d(rank_1(i))} - \frac{I(rank_2(i) \leq k)}{d(rank_2(i))} \right)$$

- ... and the document weights:

$$w_i = \frac{I(rank_1(i) \leq k)}{d(rank_1(i))} - \frac{I(rank_2(i) \leq k)}{d(rank_2(i))}$$

- This is similar to precision, but now the ranks matter as well as whether it was retrieved

DCG at rank 5

w_1 = 1/log2(5+1) – 1/log2(2+1)
= -0.244

w_2 = 1/log2(2+1) – 0/log2(8+1)
= 0.631

w_3 = 1/log2(4+1) – 0/log2(5+1)
= 0.431

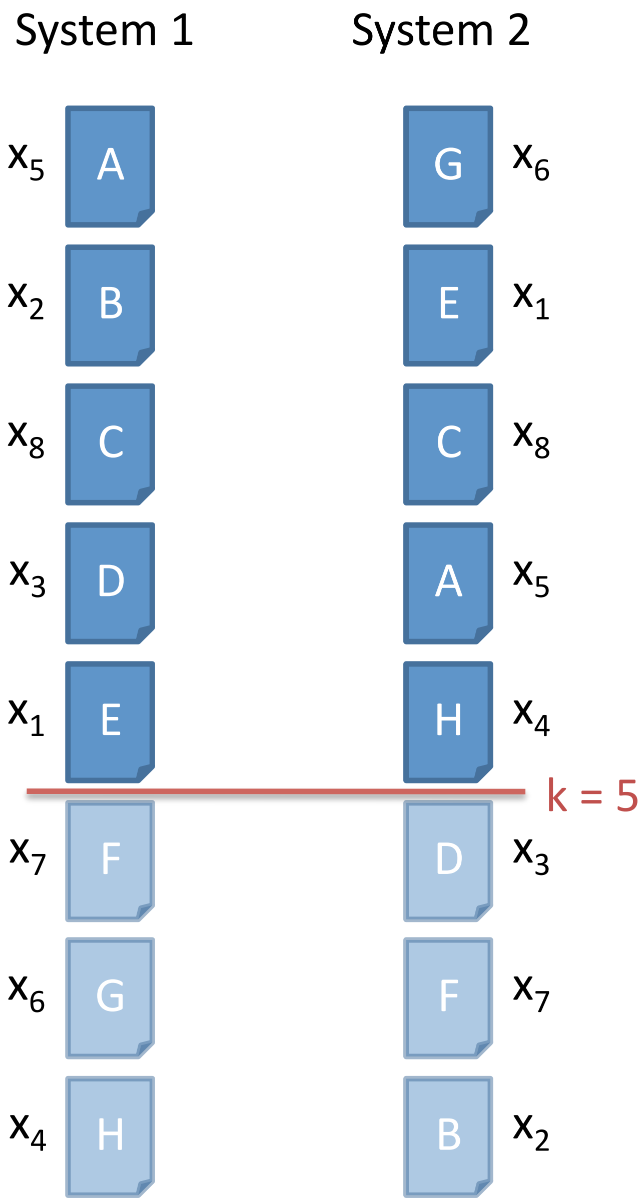
w_4 = -0.387

w_5 = 0.569

w_6 = -1.000

w_7 = 0.000

w_8 = 0.000



$$\Delta DCG@5 = \sum_{i=1}^{n} (2^{x_i} - 1) \left(\frac{I(rank_1(i) \leq 5)}{\log_2(rank_1(i) + 1)} - \frac{I(rank_2(i) \leq 5)}{\log_2(rank_2(i) + 1)} \right)$$

MTC for DCG@k

- Finally, bounds on ΔDCG@k are:

$$\lceil \Delta DCG@k \rceil = \sum_{i|i \text{ judged}} w_i g(x_i) + \sum_{i|i \text{ unjudged and } w_i > 0} w_i \max \text{gain}$$

$$\lfloor \Delta DCG@k \rfloor = \sum_{i|i \text{ judged}} w_i g(x_i) + \sum_{i|i \text{ unjudged and } w_i < 0} w_i \max \text{gain}$$

DCG at rank 5

$w_1 = -0.244$

$w_2 = 0.631$

$w_3 = 0.431$

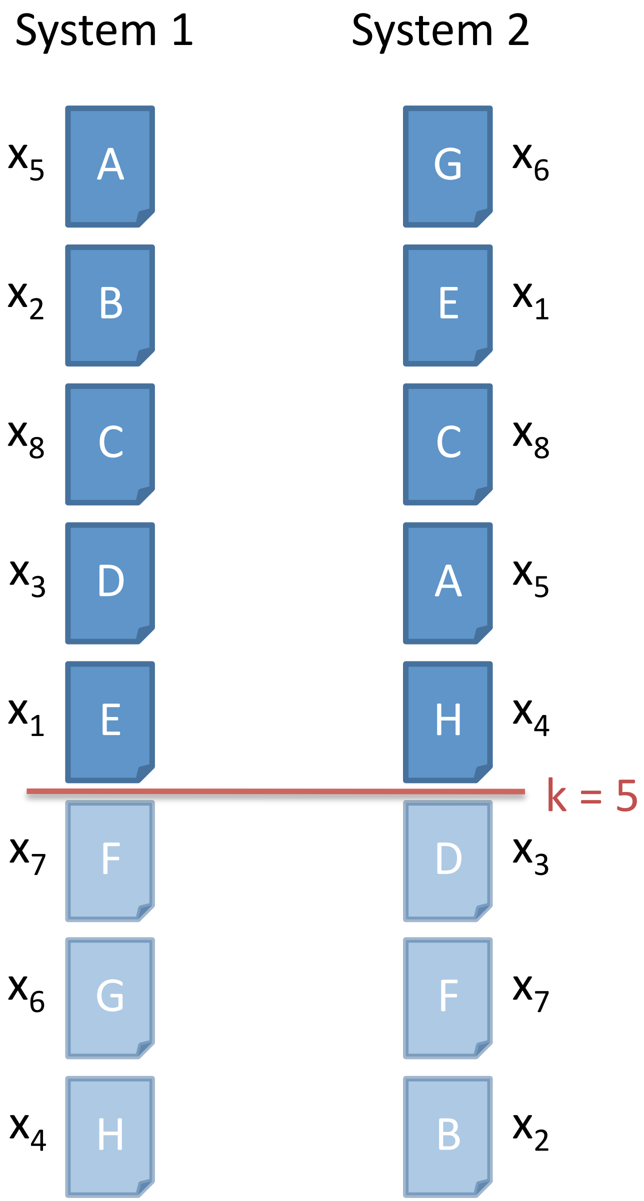
$w_4 = -0.387$

$w_5 = 0.569$

$w_6 = -1.000$

$w_7 = 0.000$

$w_8 = 0.000$



$-0.631 \leq \Delta DCG@5 \leq 1.631$

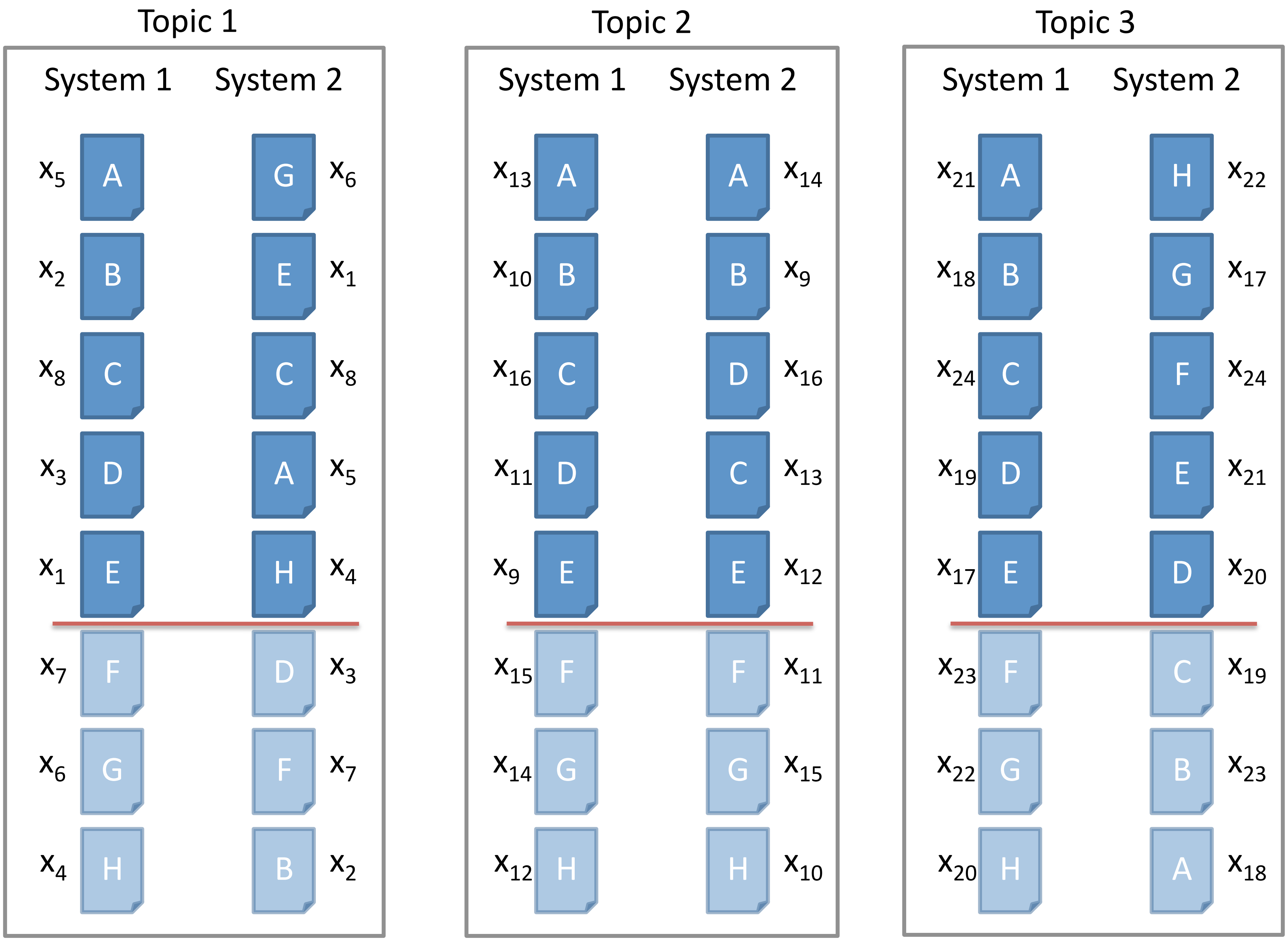
$$\Delta DCG@5 = \sum_{i=1}^{n} (2^{x_i} - 1) \left(\frac{I(rank_1(i) \leq 5)}{\log_2(rank_1(i) + 1)} - \frac{I(rank_2(i) \leq 5)}{\log_2(rank_2(i) + 1)} \right)$$

Multiple Topics

- We usually evaluate over more than just one topic
- There are two ways to use an MTC algorithm:
  1. Apply it separately to each topic

     Gives a set of signs of measure differences, e.g. 50 values of sign (ΔDCG)
  2. Apply it to all topics simultaneously

     Gives the sign of the mean difference, e.g. the value of sign(ΔDCG) averaged over 50 topics
- The second is better:
  - That's the quantity we're directly interested in
  - It allows the algorithm to find the *topics* that are interesting as well as the documents



Top 6 highest-weighted docs: G (topic 1), A (topic 3), H (topic 3), B (topic 1), B (topic 3), G (topic 3)

Recall Measures

- Note that precision and DCG do not require knowing how many relevant docs there are
  - That is the real challenge for most low-cost methods
- Can MTC work for recall, NDCG, AP, and other such measures?
  - For individual queries, yes: the denominators don't affect the difference
  - For a set of queries…?

MTC for Recall

- Again, define recall@k in terms of x_i and rank(i):

$$rec@k = \frac{1}{\sum_{i=1}^{n} x_i} \sum_{i=1}^{n} x_i I(rank(i) \leq k)$$

  – The denominator is the total number of relevant documents

- Similarly, a difference in recall:

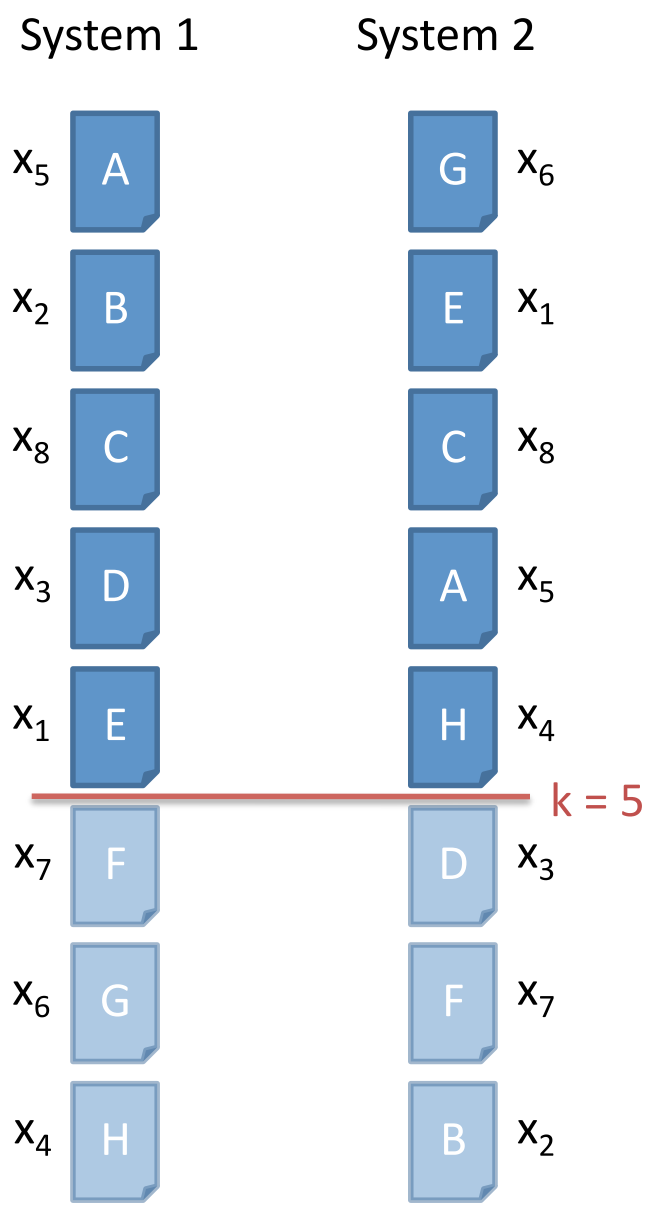
$$\Delta rec@k = \frac{1}{\sum_{i=1}^{n} x_i} \left(\sum_{i=1}^{n} x_i (I(rank_1(i) \leq k) - I(rank_2(i) \leq k) \right)$$

- To define weights, ask: what happens to our understanding of recall when we judge a document?

With no documents judged,
what are the max/min values of Δrec@5?

B, D relevant; G, H nonrelevant
⇒ Δrec@5 = 1.0

B, D nonrelevant; G, H relevant
⇒ Δrec@5 = -1.0



Suspose document B is judged relevant

i.e. $x_2 = 1$

What are the max/min values of Δrec@5?

B relevant

D relevant; G, H nonrelevant

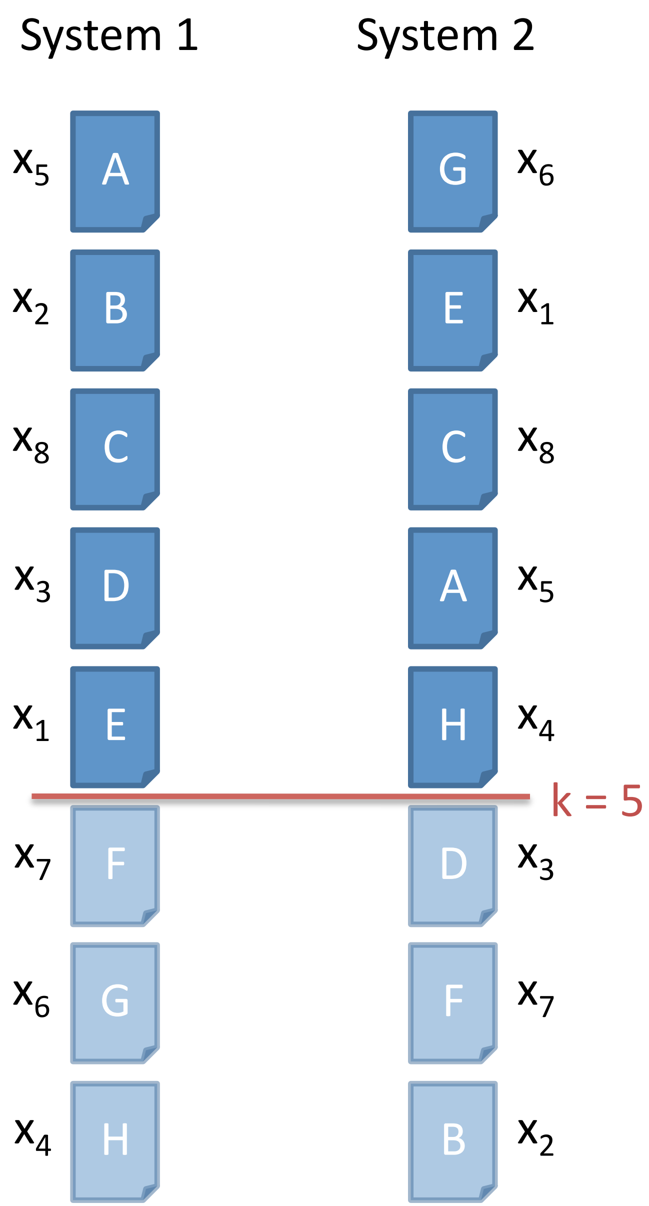
⇒ Δrec@5 = 1.0

B relevant

D nonrelevant; G, H relevant

⇒ Δrec@5 = 1/3 – 2/3 = -0.333

So -0.333 ≤ Δrec@5 ≤ 1.0



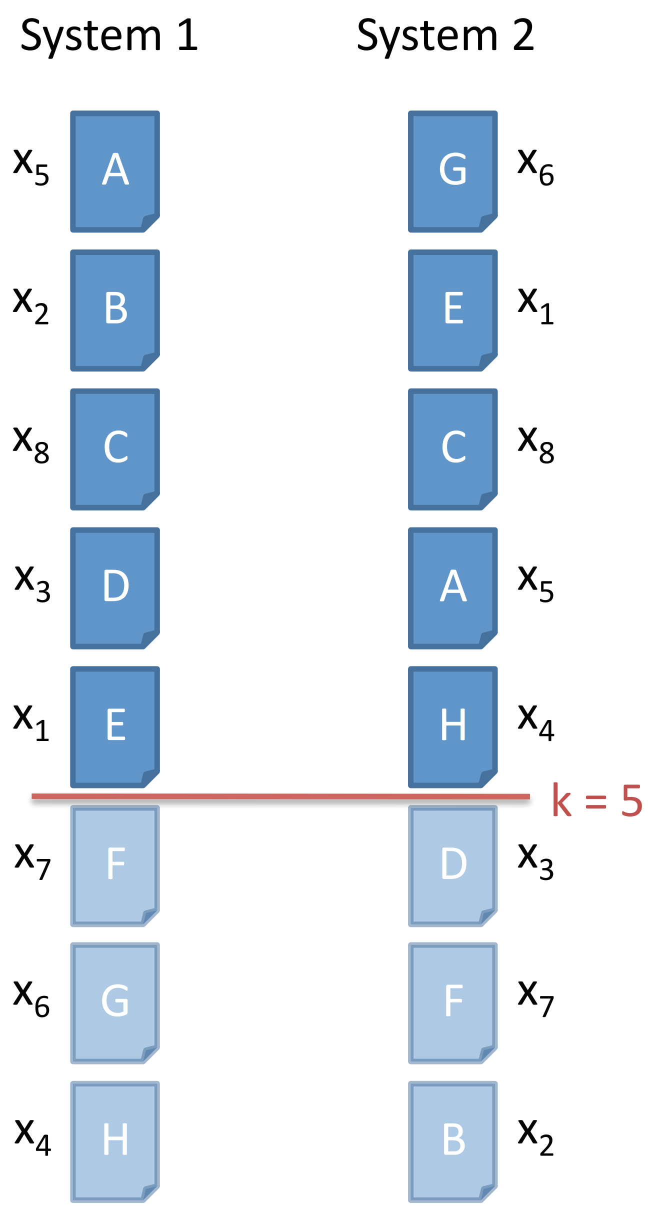
Suppose document B is judged not relevant
i.e. $x_2 = 0$

What are the max/min values of Δrec@5?

B nonrelevant
D relevant; G, H nonrelevant
⇒ Δrec@5 = 1.0

B nonrelevant
D nonrelevant; G, H relevant
⇒ Δrec@5 = 0/2 – 2/2 = -1.0

So -1.0 ≤ Δrec@5 ≤ 1.0



Judging B nonrelevant accomplishes nothing!

MTC for Recall

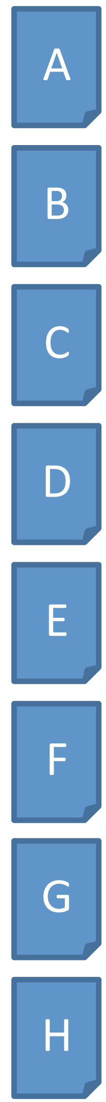
- With precision and DCG, judging a document relevant or not relevant didn't matter
  - Either way, one of the bounds is affected
  - Effect is equal in both cases
- With recall, it does matter
  - A relevant judgment increases the lower bound
  - A nonrelevant judgment does nothing
- Furthermore, each judgment affects the possible effect of future judgments

Finally: MTC for AP

- Average precision presents an additional challenge: relevance judgments interact
  - If the document at rank 1 is relevant, then the contribution of every subsequent relevant document increases
  - If the document at rank 1 is nonrelevant, then the maximum possible contribution of subsequent relevant documents decreases

- Define SP (Sum Precision) as AP*R
  - SP is between 0 and R
- If document A is relevant, its total contribution to SP is as much as 1+1/2+1/3+…
  - Depending on relevance of subsequent docs
- If document A is not relevant, SP cannot be greater than R-1-1/2-1/3-…
- Judgments of nonrelevance can be informative for AP

System 1



MTC for AP

- Define AP in terms of x_i and rank(i) as follows:

$$AP = \frac{1}{\sum_{i=1}^{n} x_i} \sum_{i=1}^{n} x_i \cdot \frac{1}{rank(i)} \sum_{j=1}^{n} x_j I(rank(j) \leq rank(i))$$

- Note that AP sums over all documents
  - Those that were not retrieved should be assumed to appear at rank infinity
- This can be usefully simplified:

$$AP = \frac{1}{\sum_{i=1}^{n} x_i} \sum_{j \leq i} \frac{1}{a_{ij}} x_i x_j, \quad a_{ij} = \min\{rank(i), rank(j)\}$$

MTC for AP

- Now define the difference in AP:

$$\Delta AP = \frac{1}{\sum x_i} \sum_{j \leq i} c_{ij} x_i x_j$$

$$c_{ij} = \frac{1}{\max\{rank_1(i), rank_1(j)\}} - \frac{1}{\max\{rank_2(i), rank_2(j)\}}$$

- For simplicity, ignore the denominator for now

Assume all documents are nonrelevant

What happens if we judge one relevant?

x_1: $SP_1 = 1/5$, $SP_2 = 1/2$

$\Delta SP = -0.300$

x_2: $SP_1 = 1/2$, $SP_1 = 1/8$

$\Delta SP = 0.375$

x_3: $\Delta SP = 0.083$

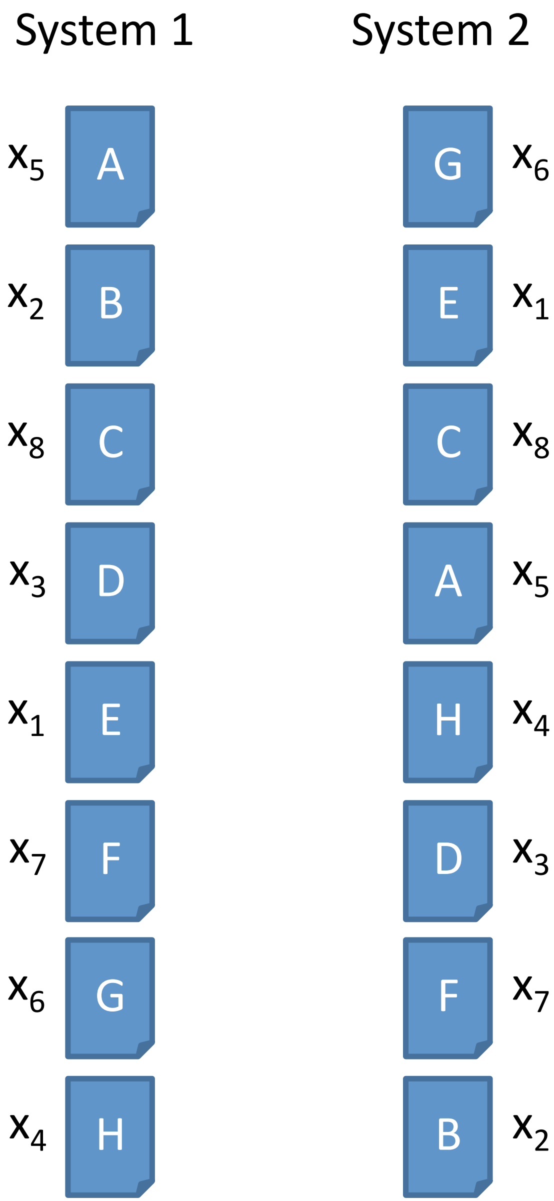
x_4: $\Delta SP = -0.075$

x_5: $\Delta SP = 0.750$

x_6: $\Delta SP = -0.857$

x_7: $\Delta SP = 0.024$

x_8: $\Delta SP = 0.000$



Or assume all documents are relevant
What happens if we judge one nonrelevant?

x_1: $SP_1 = 1+1+1+1+5/6+6/7+7/8$
$SP_2 = 1+2/3+3/4+\ldots+7/8$
$\Delta SP = 0.783$

x_2: $SP_1 = 1+2/3+3/4+\ldots+7/8$
$SP_1 = 1+1+1+1+1+1+1$
$\Delta SP = -1.218$

x_3: $\Delta SP = -0.367$

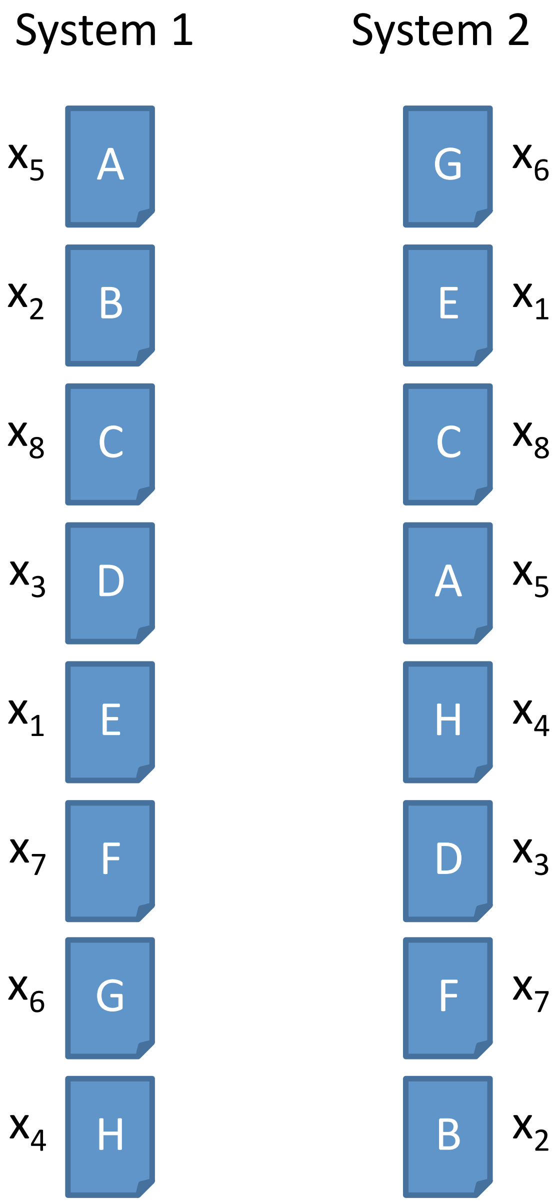
x_4: $\Delta SP = 0.434$

x_5: $\Delta SP = -1.083$

x_6: $\Delta SP = 1.593$

x_7: $\Delta SP = -0.143$

x_8: $\Delta SP = 0.000$



Calculating Document Weights

- Initially each document gets a "relevant weight" and a "nonrelevant weight"
  - Relevant weight = effect on ΔSP if relevant
    $= c_{ii}$
  - Nonrelevant weight = effect on ΔSP if nonrelevant
    $= c_{ii} + c_{1i} + c_{2i} + c_{3i} + \dots + c_{ni}$
- Judge the document with the greatest maximum of rel weight and nonrel weight

G judged nonrelevant ($x_6 = 0$)
Assume all documents are nonrelevant
What happens if we judge one relevant?

x_1: $SP_1 = 1/5$, $SP_2 = 1/2$
$\Delta SP = -0.300$

x_2: $SP_1 = 1/2$, $SP_1 = 1/8$
$\Delta SP = 0.375$

x_3: $\Delta SP = 0.083$

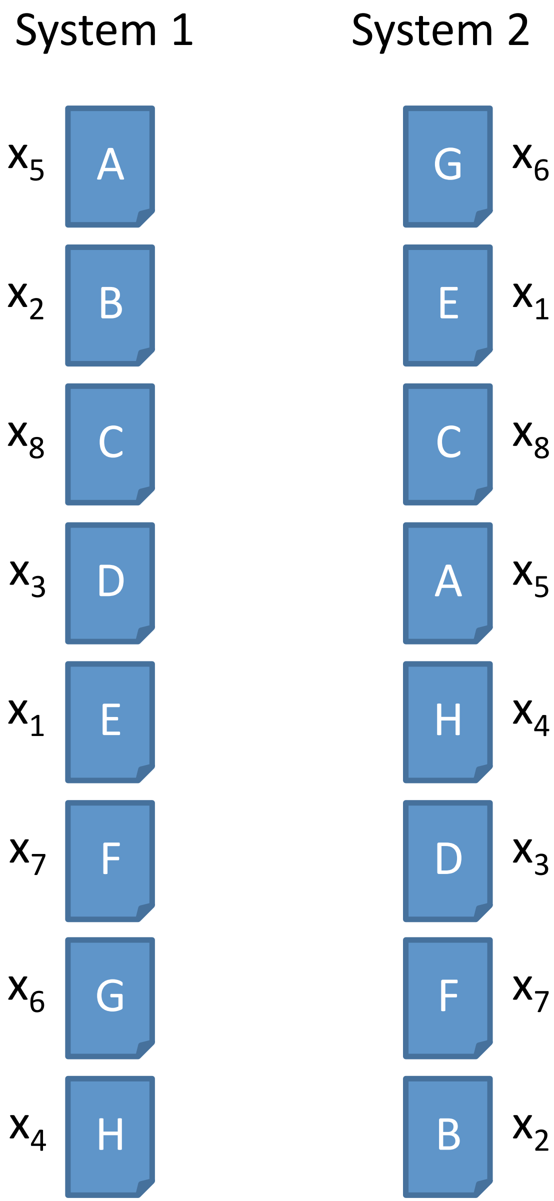
x_4: $\Delta SP = -0.075$

x_5: $\Delta SP = 0.750$

~~x_6: $\Delta SP = -0.857$~~

x_7: $\Delta SP = 0.024$

x_8: $\Delta SP = 0.000$



Or assume all documents are relevant
What happens if we judge one nonrelevant?

x_1: $SP_1 = 1+1+1+1+5/6+6/8$
$SP_2 = 1/3+2/4+\ldots+6/8$
$\Delta SP = 2.019$

x_2: $SP_1 = 1+2/3+3/4+\ldots+6/8$
$SP_1 = 1/2+2/3+\ldots+6/7$
$\Delta SP = 0.393$

x_3: $\Delta SP = 1.202$

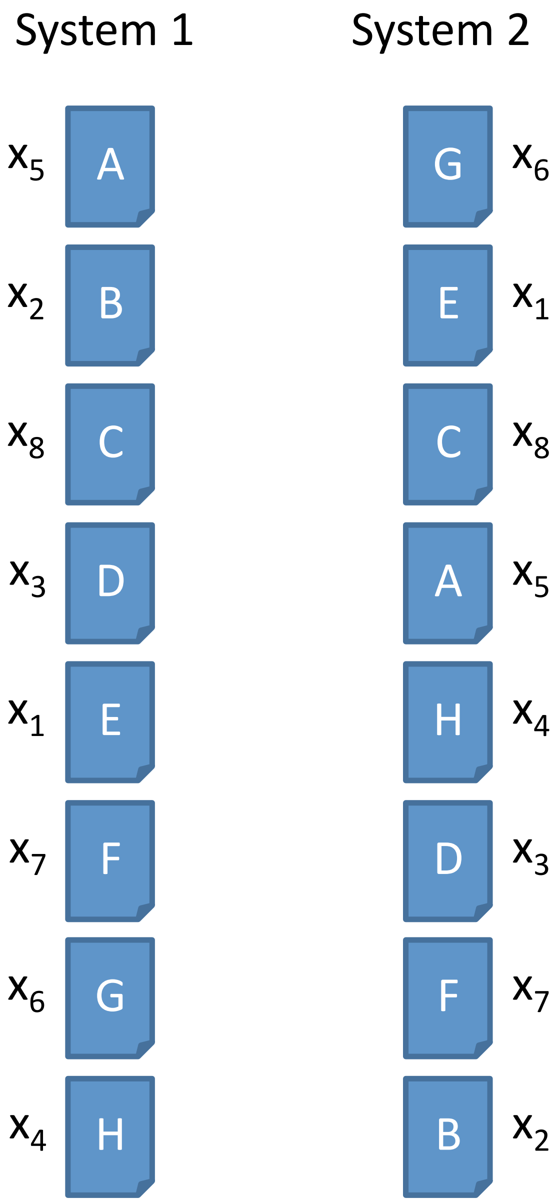
x_4: $\Delta SP = 1.952$

x_5: $\Delta SP = 0.402$

~~x_6: $\Delta SP = 1.593$~~

x_7: $\Delta SP = 1.450$

x_8: $\Delta SP = 1.402$



Updating Document Weights

$$w_i^R = c_{ii} + \sum_{j \mid j \text{ judged}} c_{ij} x_j$$

$$w_i^N = c_{ii} + \sum_{j \mid j \text{ judged}} c_{ij} x_j + \sum_{j \mid j \text{ not judged}} c_{ij}$$

base effect

additional base for nonrel weight

interactions with
judged documents

G judged nonrelevant ($x_6 = 0$)
E judged relevant ($x_1 = 1$)

~~x_1: $\Delta SP = -0.600$~~

x_2: $\Delta SP = 0.150$

x_3: $\Delta SP = -0.183$

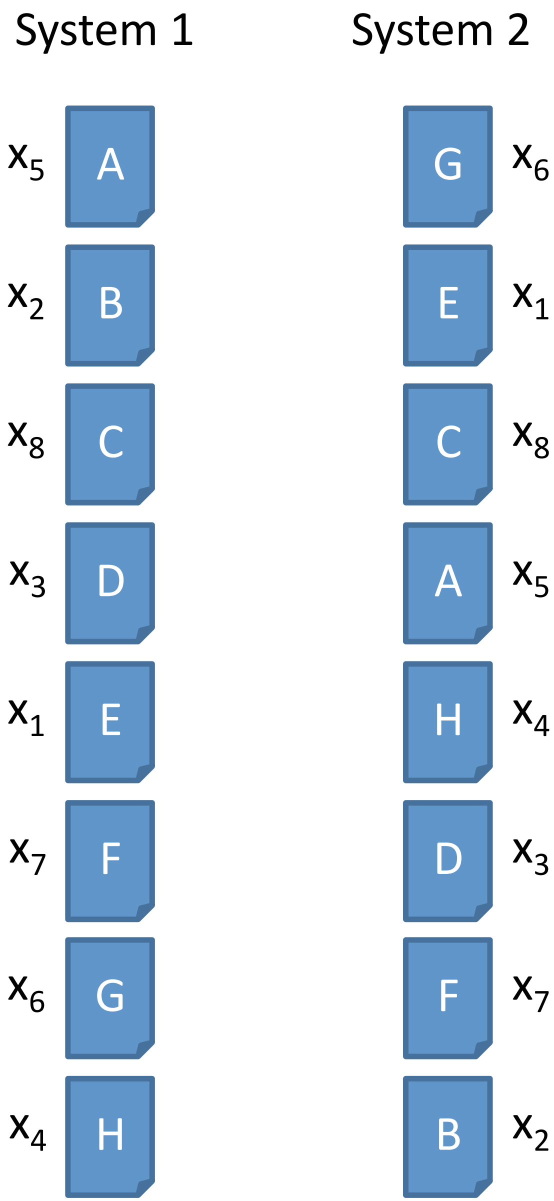
x_4: $\Delta SP = -0.450$

x_5: $\Delta SP = 0.400$

~~x_6: $\Delta SP = -1.514$~~

x_7: $\Delta SP = 0.252$

x_8: $\Delta SP = -0.433$



Stopping Condition

- How do we calculate bounds on ΔSP?
  - A: We don't. They're too hard (NP-Hard).
- But we can still determine whether the stopping condition is satisfied
  - "Look ahead"
  - If the algorithm continued in the best case, would our conclusion change?
- If ΔSP > 0 with current judgments, can it become < 0 after a series of future judgments?

So far we know:

G is nonrelevant

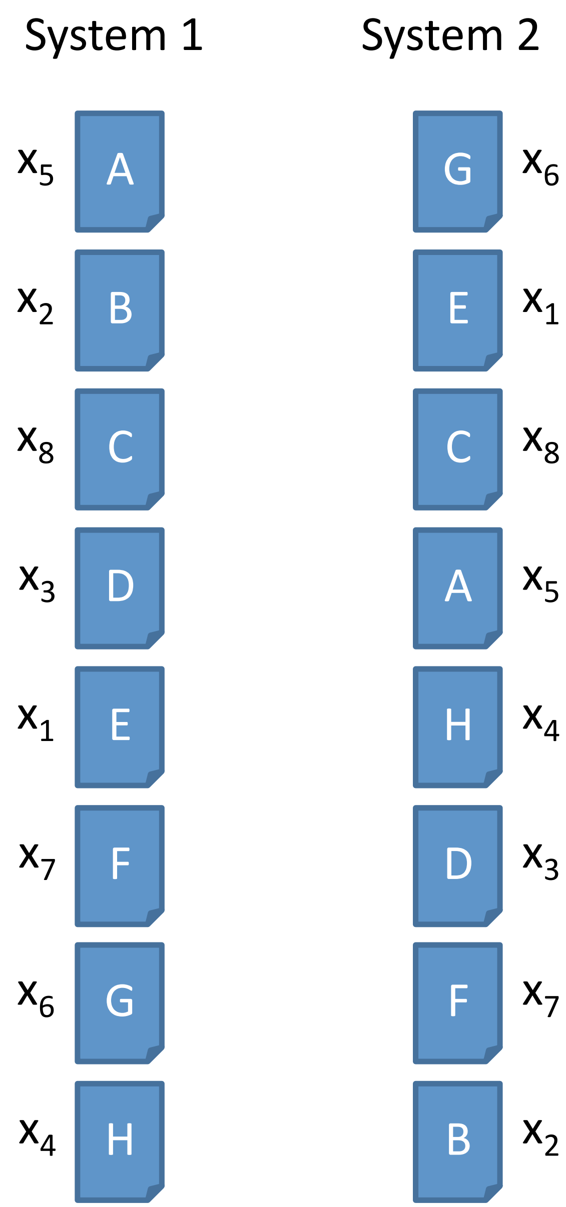
E is relevant

Based on that, ΔSP = -0.3

Is it possible for system 1 to catch up?

YES: if A is judged relevant,

ΔSP will go up to 0.4



MTC for AP: Algorithm

- while (!done)
  - for each unjudged document i,
    - $w_i = \max\{w_i^R, w_i^N\}$ (where w_i^R, w_i^N calculated as above)
  - judge document with max $|w_i|$
  - calculate ΔAP with current judgments
    - if ΔAP > 0, simulate algorithm forward taking documents in order of increasing w_i^R
    - if ΔAP < 0, simulate forward taking documents in order of decreasing w_i^R
  - if sign is the same after simulation, done = true

MTC: Summary So Far

- MTC is a family of algorithms with specific cases for each evaluation measure
- An algorithm is defined by
  - A way to weight documents
  - A way to select which document to judge next
  - A way to update document weights
  - A stopping condition based on bounds
- Some algorithms are easier to understand/implement/prove optimal than others…

Refining the Bounds

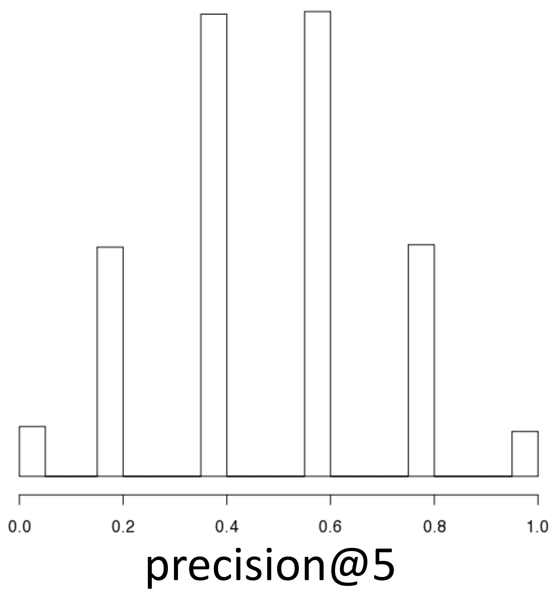
- Lower and upper bounds are a blunt instrument
  - Bounds can be on the wrong side of zero, but only by a small fraction

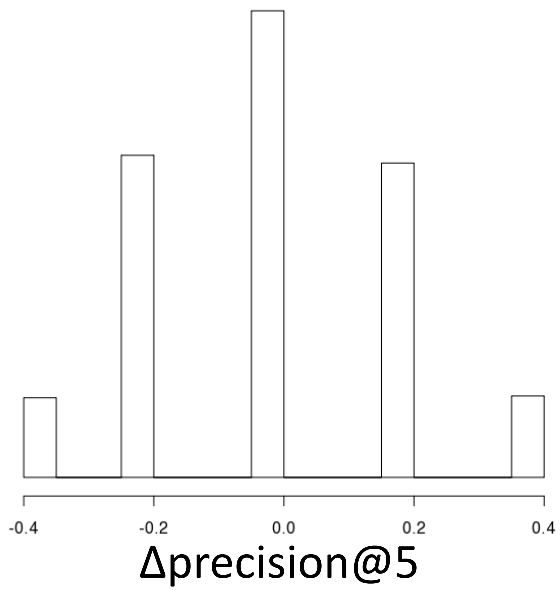
- Define a probability distribution over values between the bounds
  - If the total probability of values greater than 0 is low, stop judging

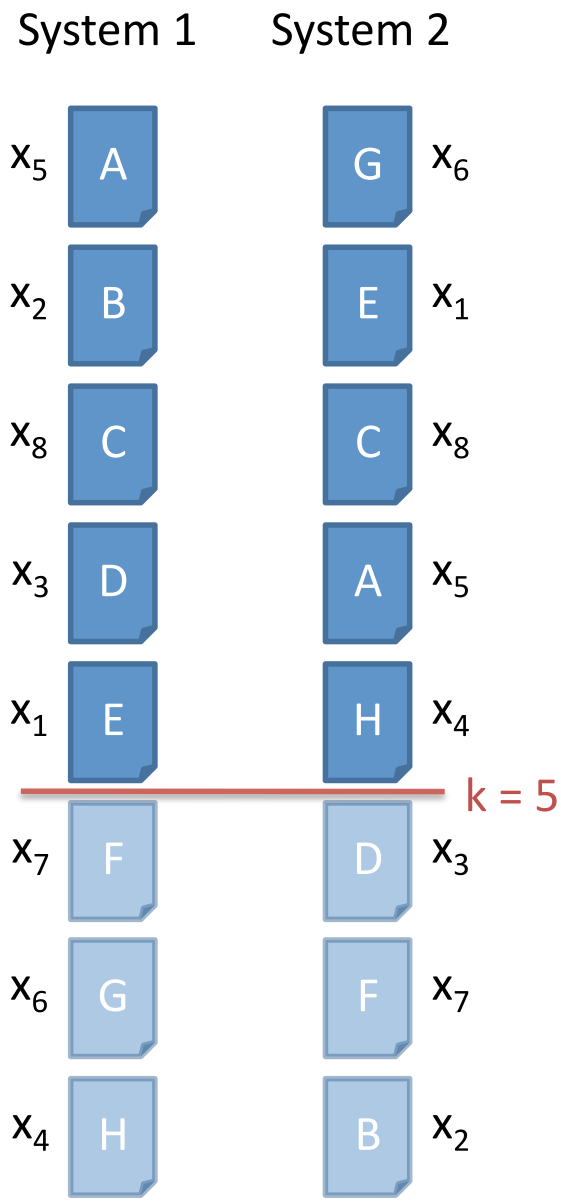
Distributions of Evaluation Measures

- Basic idea:
  - There is a set of m unjudged documents
  - Each one could be relevant or nonrelevant
  - Thus, there are 2^m total possible ways to assign relevance to the unjudged documents
  - Each one of those assignments results in a particular value of the measure
  - We can therefore count the number of ways every possible value of Δprec@k, Δrec@k, ΔAP, etc. can occur

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$prec_1@5$	$prec_2@5$	$\Delta prec@5$
0	0	0	0	0	0	0	0	0.0	0.0	0.0
1	1	1	1	1	1	1	1	1.0	1.0	0.0
1	0	1	0	1	0	1	0	0.6	0.4	0.2
0	1	0	1	0	1	0	1	0.4	0.6	-0.2
1	1	0	0	1	1	0	0	0.6	0.6	0.0



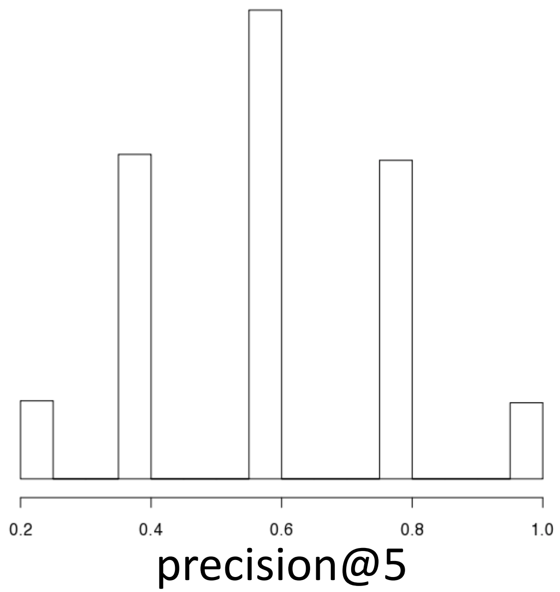


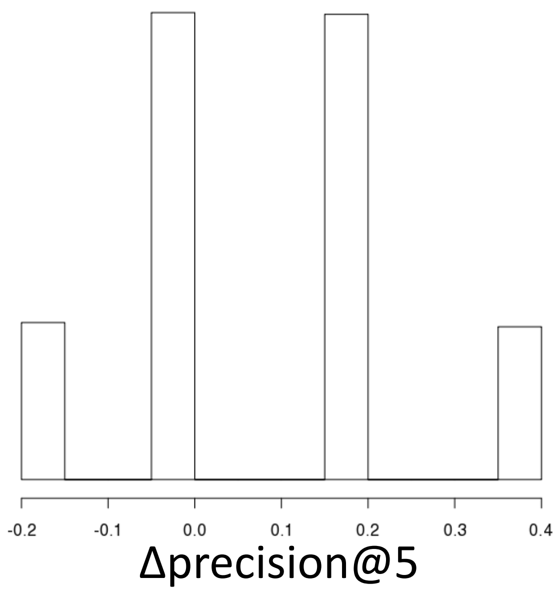


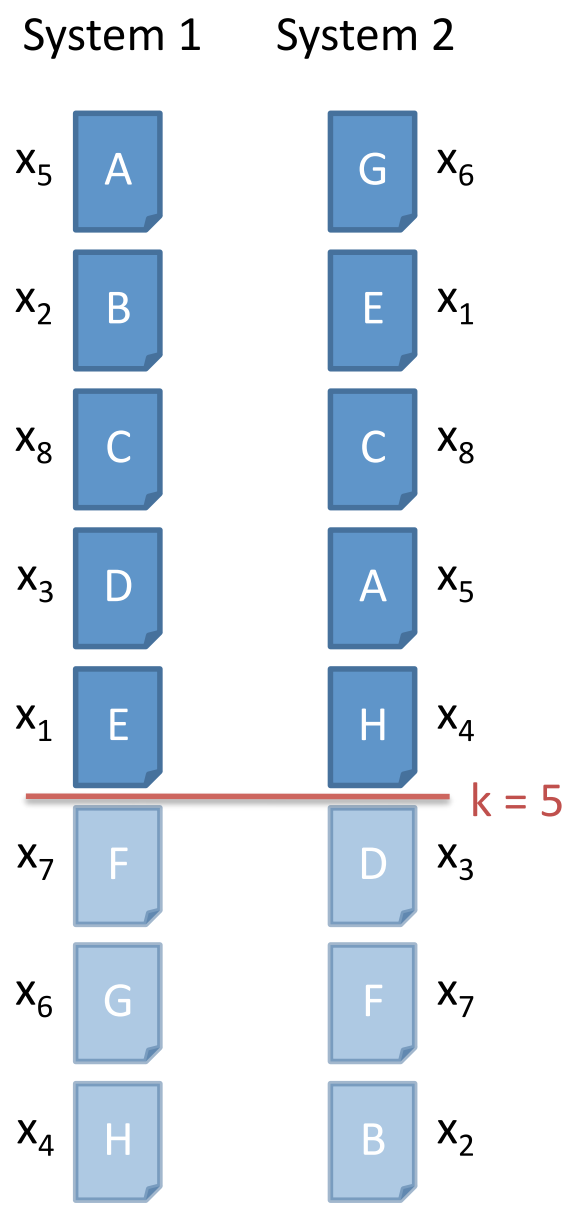
Distributions of Evaluation Measures

- Forming a distribution:
  - Assume each of the 2^m assignments of relevance is equally likely
    - uniform distribution over possible assignments of relevance
  - Result: values of Δprec@k have a binomial distribution
- As documents are judged, the distribution's center shifts, but it remains binomial

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$prec_1@5$	$prec_2@5$	$\Delta prec@5$
0	0	0	0	0	0	0	0	0.0	0.0	0.0
1	1	1	1	1	1	1	1	1.0	1.0	0.0
1	0	1	0	1	0	1	0	0.6	0.4	0.2
0	1	0	1	0	1	0	1	0.4	0.6	-0.2
1	1	0	0	1	1	0	0	0.6	0.6	0.0







Normal Approximations

- The binomial distribution can be approximated by a normal distribution
  - Pretty close approximation even for small k
- It turns out that distributions of ΔDCG and ΔAP can also be roughly approximated by normal distributions
  - Proofs possible using combinatoric arguments and limit theory
  - Proofs don't require uniform distribution of relevance assignments

Using Distributions in MTC

- Since measures are normally distributed, it is very easy to compute the probability that one system will be better than another
  - i.e. given a set of judgments J, we can easily compute P(Δmeasure > 0 | J)
- This in turn lets us know whether it's worth making more judgments
  - Instead of computing bounds, compute a probability
  - If the probability is low, judging can stop

Results

- So how well does MTC actually do?
- Experiment: randomly select a pair of systems, compare them using MTC
  - Validate against "true" results using TREC qrels

	MTC for AP		pooling	
collection	**judgments**	**% correct**	**judgments**	**% correct**
TREC-3	367.77	91%	622.04	96%
TREC-4	411.11	97%	559.44	100%
TREC-5	408.29	91%	813.76	100%
TREC-6	354.19	91%	1198.36	96%
TREC-7	302.59	89%	892.37	93%
TREC-8	297.44	91%	731.48	100%

Inferring Relevance

- A uniform distribution over relevance assignments is not a good assumption
  - Documents that were not retrieved are as likely to be relevant as documents at rank 1?
- Better estimates of the relevance of individual documents would improve performance

Inferring Relevance

- We want an estimate of the probability that each document is relevant
  - i.e. $p_i = P(x_i = 1)$
- Our goal will be to use existing relevance judgments to train a model of relevance
- What we can do that IR systems cannot:
  - Use the judgments for a particular topic as training data, then predict judgments on documents for the same topic

Inferring Relevance

- First assumption we're going to make:
  - Documents are independently relevant, i.e. $P(x_1, x_2, \ldots, x_n) = P(x_1)P(x_2)\ldots P(x_n)$
  - This is a basic assumption of ad hoc IR and many other IR tasks
- Second assumption:
  - The log of the odds of relevance of a document is a linear combination of feature values
  - This is for simplicity: linear models are easier to fit

Inferring Relevance

- The model is:

$$\log \frac{p_i}{1 - p_i} = \beta_0 + \sum_{j=1}^{F} \beta_j f_{ij}$$

  - where f_{ij} is the value of a feature calculated from document i and β_j is a coefficient
- Note that this is just a logistic regression model, appropriate for binary judgments
  - Graded judgments would require an ordinal model

Features for Inferring Relevance

- Features can be anything appropriate for predicting relevance
- Some we have tried:
  - Document similarity features
  - System performance features
  - Click features
- The following slides will discuss each in slightly more detail

Document Similarities

- Using document similarities as features is inspired by van Rijsbergen's Cluster Hypothesis:
  - *Closely associated documents tend to be relevant to the same requests*
- Take a shallow pool of documents to be "features"
- Feature values for document i are its similarities to every document in that pool

System Performance

- Use features derived from the systems being evaluated, such as:
  - Number of (known) relevant documents retrieved
  - Ranks at which relevant documents appear
  - Precisions at ranks of relevant documents
- Inspiration is the "metasearch hypothesis" (cf. Joon Ho Lee):
  - *Systems tend to retrieve the same relevant documents but different nonrelevant documents*

Clicks

- If available, the number of clicks on a document may be indicative of its relevance
- Some complications:
  - Presentation bias: higher ranks are preferred even if less relevant
  - Interactions: relevance of document at rank i can affect clicks at rank j

MTC Evaluation

- As we said earlier, MTC evaluation is separate from its document selection
  - We could use MTC judgments with the usual assumption: that unjudged docs are not relevant
    - Since MTC is not trying to find all the relevant documents, this is probably not appropriate, though
  - We could use bpref or Q-measures that explicitly account for whether a document is judged or not
- MTC evaluation instead uses the idea of forming a distribution over possible values of the evaluation measure

MTC Evaluation

- The idea is the same as with the stopping condition:
  - We used distribution of ΔAP to calculate P(ΔAP > 0)
  - Now we will just look at the distribution of P(AP)
- But a distribution is not an evaluation measure
- For a single-number summary, calculate the expectation of the distribution

MTC Evaluation

- Since we're assuming documents are independently relevant, expectations are easy

$$E[prec@k] = \frac{1}{k}\sum_{i=1}^{n} p_i I(rank(i) \leq k)$$

$$E[R] = \sum_{i=1}^{n} p_i$$

$$E[rec@k] \approx \frac{1}{E[R]}\sum_{i=1}^{n} p_i I(rank(i) \leq k)$$

$$E[AP] \approx \frac{1}{E[R]}\sum_{i=1}^{n} c_{ii} p_i + \sum_{i<j} c_{ij} p_i p_j$$

MTC Evaluation

- What we can show:
  - Although E[AP] is an approximation, the error is on the order of 2^{-n} in the size of the collection
  - Variance of AP is also computable in $O(n^3)$ time
- What we cannot show:
  - That E[AP] is a good estimate of the actual value of AP
    - In practice it is not: our relevance models tend to overestimate relevance, leading to low values of E[AP]

MTC Evaluation: Example

run	topic	eR	eAP	eRprec	eP5	eP10
udelIndDRPR	1	3518.66	0.0177	0.0569	0.0433	0.1681
udelIndDRSP	1	3518.66	0.0830	0.1129	1.0000	0.9857
udelIndDMRM	1	3518.66	0.0792	0.1101	1.0000	0.9857

summary results:

run	eMAP	eRprec	eP5	eP10
udelIndDRPR	0.030971	0.090344	0.265973	0.295068
udelIndDMRM	0.046869	0.103990	0.231451	0.323774
udelIndDRSP	0.047082	0.104238	0.277171	0.356119

UDel results from TREC 2009 Web track (ad hoc task)

MTC Evaluation: Example

```
summary results:
run                eMAP    eRprec       eP5      eP10
udelIndDRPR 0.030971 0.090344 0.265973 0.295068
udelIndDMRM 0.046869 0.103990 0.231451 0.323774
udelIndDRSP 0.047082 0.104238 0.277171 0.356119
```

```
pairwise comparisons
            udelIndDMRM     udelIndDRSP
udelIndDRPR     -0.0159         -0.0161
                 0.0000          0.0000
                 1.0000          1.0000
udelIndDMRM                     -0.0002
                                 0.0000
                                 0.5551
```

UDel results from TREC 2009 Web track (ad hoc task)

MTC in Practice

- Practical considerations include:
  - Selecting documents when more than two systems are involved
    - Simple solution: judge the document with maximum weight across all pairs—computable in linear time
  - Deciding which documents to predict relevance
    - Usually infeasible to do all of them, instead restrict to pool of retrieved documents
  - "Unbiasing" expected evaluation measures
    - Possibly using priors to keep relevance models from overestimating—work in progress

MTC Summary

- MTC is a family of algorithms for selecting documents to judge
  - The probabilistic stopping condition of those algorithms also produces an evaluation measure
- The best way to use MTC is to compare systems
- The best way to interpret it is with the probability that one system is better than another
  - i.e. $P(\Delta AP > 0)$
  - This is the quantity that tells you whether you can have confidence that the judgments are sufficient