Rank Learning with the Committee Perceptron

Jonathan L. Elsas, Vitor R. Carvalho, Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University

Rank Learning with the Committee Perceptron

Jonathan L. Elsas, Vitor R. Carvalho, Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University


A Brief History of Features in IR

A Brief History of Features in IR

- In the beginning there was exact match.

A Brief History of Features in IR

- In the beginning there was exact match.
- Models evolved, bringing more features: TF, IDF, document length.

A Brief History of Features in IR

- In the beginning there was exact match.
- Models evolved, bringing more features: TF, IDF, document length.
- Collections evolved, still more features: link structure, anchor text, document structure.

A Brief History of Features in IR

- In the beginning there was exact match.
- Models evolved, bringing more features: TF, IDF, document length.
- Collections evolved, still more features: link structure, anchor text, document structure.
- Today:
  social annotations, click-through data, ...

Example Features

Example Features

$$f_1(Q, d) = \sum_{t_i \in Q} tf_{t_i;d}$$

Raw Query Term Freq.

Example Features

$$f_1(Q,d) = \sum_{t_i \in Q} tf_{t_i;d}$$

Raw Query Term Freq.

$$f_2(Q,d) = \prod_{t_i \in Q} \frac{tf_{t_i;d} + \mu P(t_i|C)}{dl_d + \mu}$$

Language Modeling
Query Likelihood

Example Features

$$f_1(Q,d) = \sum_{t_i \in Q} tf_{t_i;d}$$

Raw Query Term Freq.

$$f_2(Q,d) = \prod_{t_i \in Q} \frac{tf_{t_i;d} + \mu P(t_i|C)}{dl_d + \mu}$$

Language Modeling
Query Likelihood

$$f_3(Q,d) = \prod_{t_i \in Q} \frac{tf_{t_i;d_{title}} + \mu P(t_i|C)}{dl_{dtitle} + \mu}$$

Query Likelihood
title only

Example Features

$$f_1(Q,d) = \sum_{t_i \in Q} tf_{t_i;d}$$

Raw Query Term Freq.

$$f_2(Q,d) = \prod_{t_i \in Q} \frac{tf_{t_i;d} + \mu P(t_i|C)}{dl_d + \mu}$$

Language Modeling Query Likelihood

$$f_3(Q,d) = \prod_{t_i \in Q} \frac{tf_{t_i;d_{title}} + \mu P(t_i|C)}{dl_{dtitle} + \mu}$$

Query Likelihood title only

$$f_4(Q,d) = \mathrm{PageRank}(d)$$

Query-independent Score

How do we use all these features?

How do we use all these features?

- Many features used in real-world web search engines

How do we use all these features?

- Many features used in real-world web search engines
- Ideally, adapt feature weights across tasks & users

How do we use all these features?

- Many features used in real-world web search engines
- Ideally, adapt feature weights across tasks & users
- The solution: learning from previous queries + relevance judgements

Learning to Rank (LETOR)

Recent explosion of research:

- RankSVM — Joachims, 2002
- RankBoost — Freund & Schapire, 2003
- RankNet — Burges et. al., 2005
- ListNet (Cao et. al., 2007), AdaRank (Xu & Li, 2007), LambdaRank (Burges et. al., 2006), and many more

Pairwise Preference Learning

- Training data consists of *pairs* of documents

$$\{(q, d_i, d_j) \quad | \quad d_i \succ_q d_j\}$$

- Learn a *preference function (s)* over pairs

$$d_i \succ d_j \iff s(d_i) > s(d_j)$$

- *Ranking* reduces to *classification* over pairs of documents:

+ $\boxed{s(d_i) - s(d_j) > 0}$

− $\boxed{s(d_i) - s(d_j) \leq 0}$

Pairwise Preference Learning

Goal: minimize number of mis-ranked document pairs

$$L_s = \sum_{(d_i, d_j)} \mathbb{I}[(s(d_i) - s(d_j)) \leq 0]$$

Pairwise Preference Learning

- Most classification algorithms can be adapted to this task
- Generalizes to any graded relevance levels, or any (full/partial) ordering of training data
- Evidence that pairwise preference assessment is easier for assessors

(Carterette et. al., ECIR 2008)

Pairwise Preference Learning

Minimizing the number of mis-ranked pairs places a lower-bound on many common retrieval performance measures

MAP, P@k, R-Precision, MRR

Pairwise Preference Learning: Linear Setting

- Documents represented by a vector of feature values:

$$\mathbf{d}_{i,q} = (f_0(d_i, q), f_1(d_i, q), \ldots, f_m(d_i, q))$$

- With a linear scoring function:

$$s(\mathbf{d}_{i;q}; \mathbf{w}) = \langle \mathbf{d}_{i;q}, \mathbf{w} \rangle$$

- Loss function becomes:

$$L_s = \sum_{(d_i, d_j)} [\langle \mathbf{d}_{i;q} - \mathbf{d}_{j;q}, \mathbf{w} \rangle]_+$$

Perceptron Algorithm

(Rosenblatt, 1958)

- Online algorithm, instance at a time
- Update current hypothesis (***w***) whenever a classification (or ranking) mistake is made.
- Provable mistake bounds & convergence
- **Scalable to large data sets**

Perceptron Algorithm

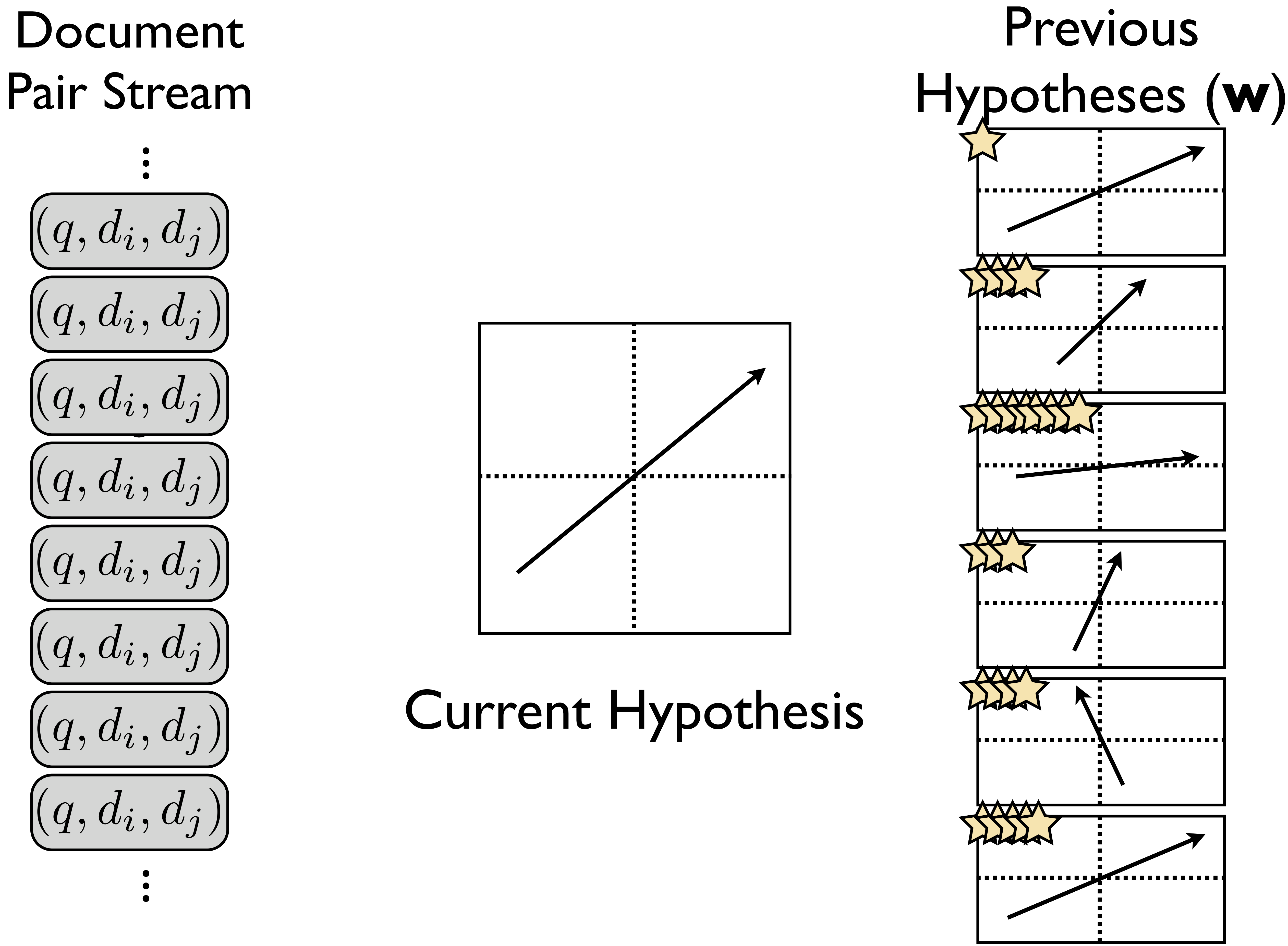
- Recall, pairwise preference loss function:

$$L_s = \sum_{(d_i, d_j)} [\langle \mathbf{d}_{i;q} - \mathbf{d}_{j;q}, \mathbf{w} \rangle]_+$$

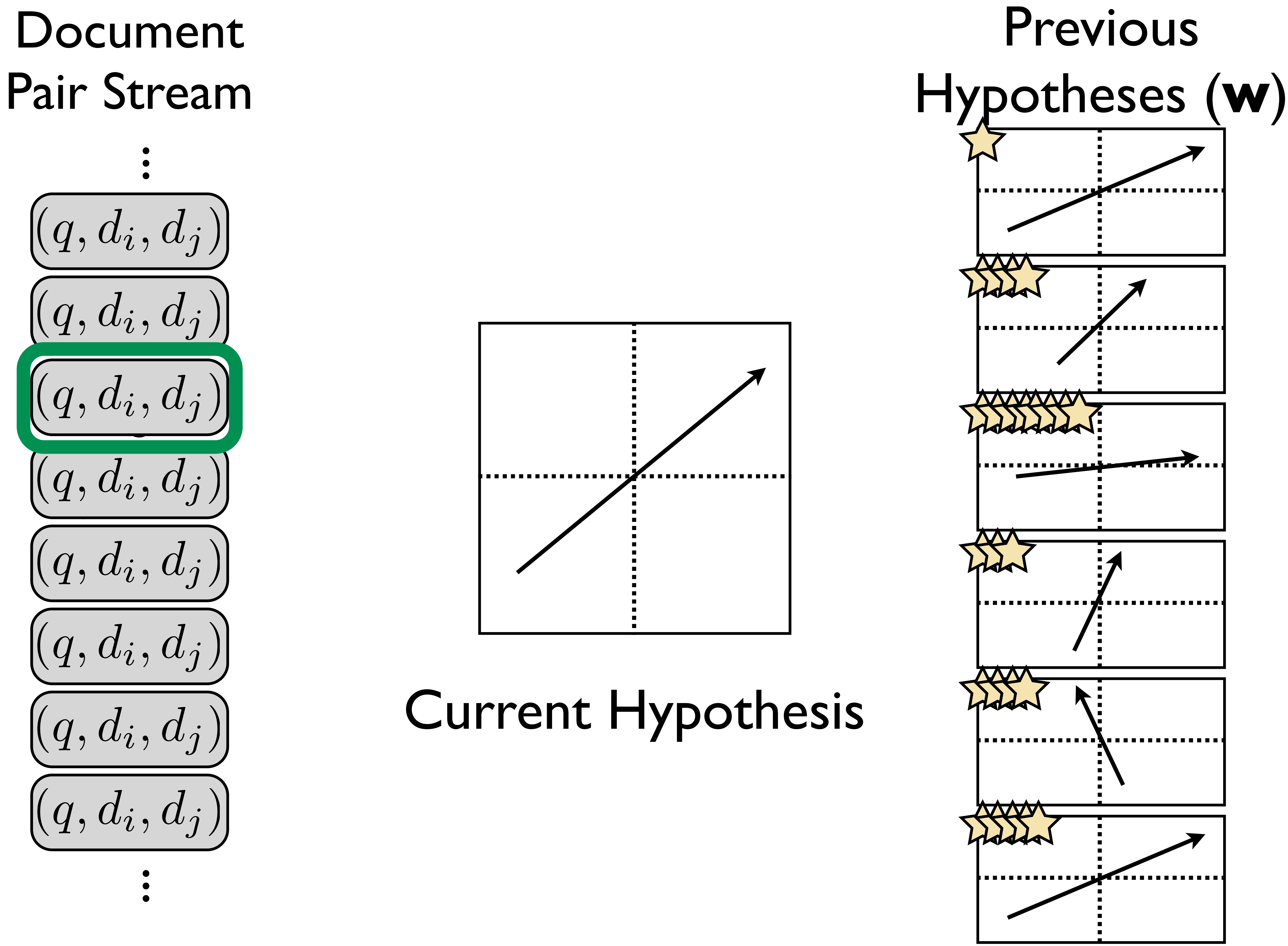
- Simple iterative update rule for document ranking:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \eta_q(\mathbf{d}_{i;q} - \mathbf{d}_{j;q})$$

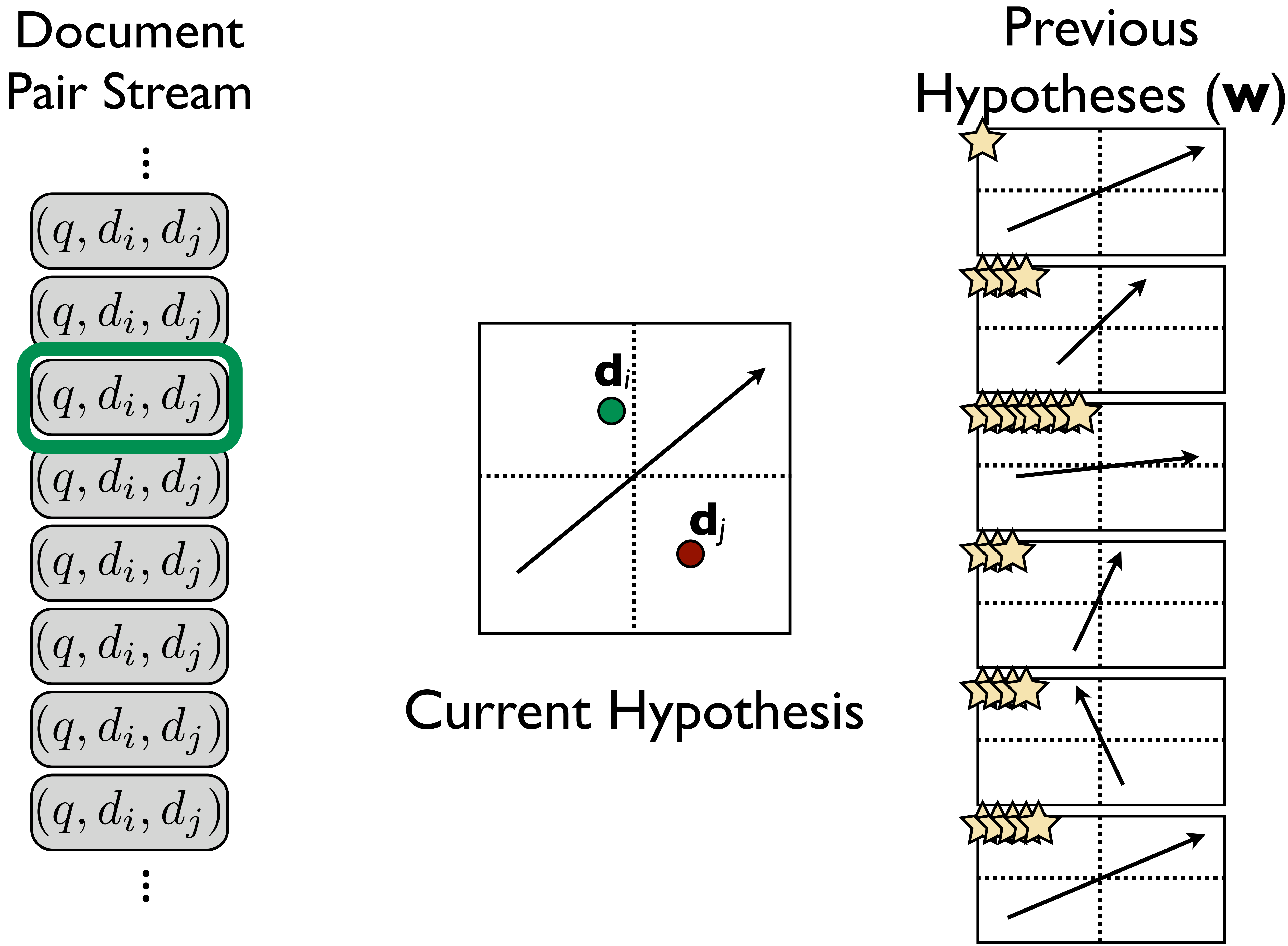
- Can't globally minimize L_s, but perceptron update rule provably bounds mis-rankings

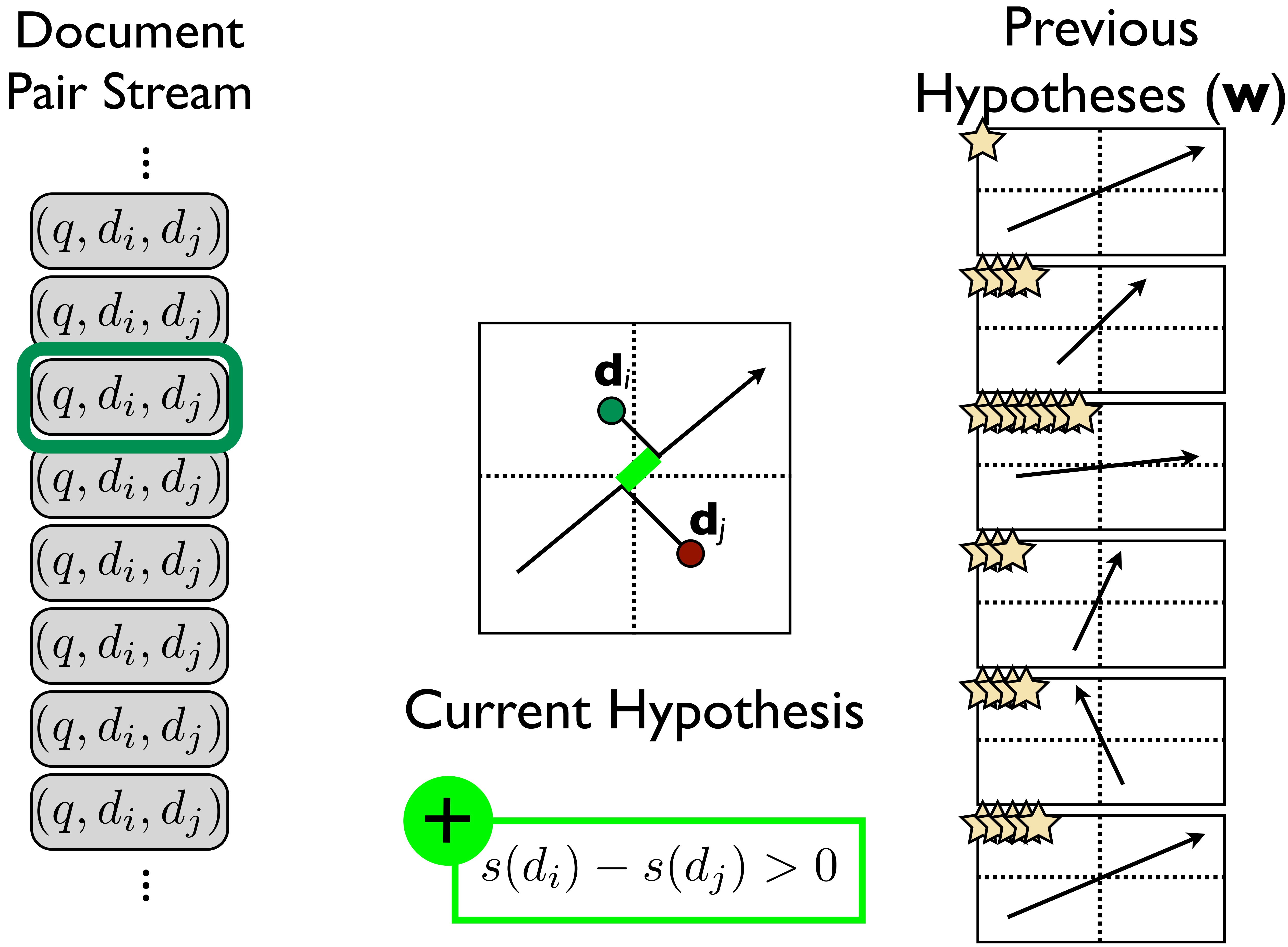


Document
Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
Current Hypothesis
Previous
Hypotheses ($\mathbf{w}$)



Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
Current Hypothesis
Previous Hypotheses (**w**)


Document
Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
Previous
Hypotheses ($\mathbf{w}$)



Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
+
$s(d_i) - s(d_j) > 0$
Previous Hypotheses ($\mathbf{w}$)

Document Pair Stream

⋮

(q, d_i, d_j)

(q, d_i, d_j)

(q, d_i, d_j)

(q, d_i, d_j)

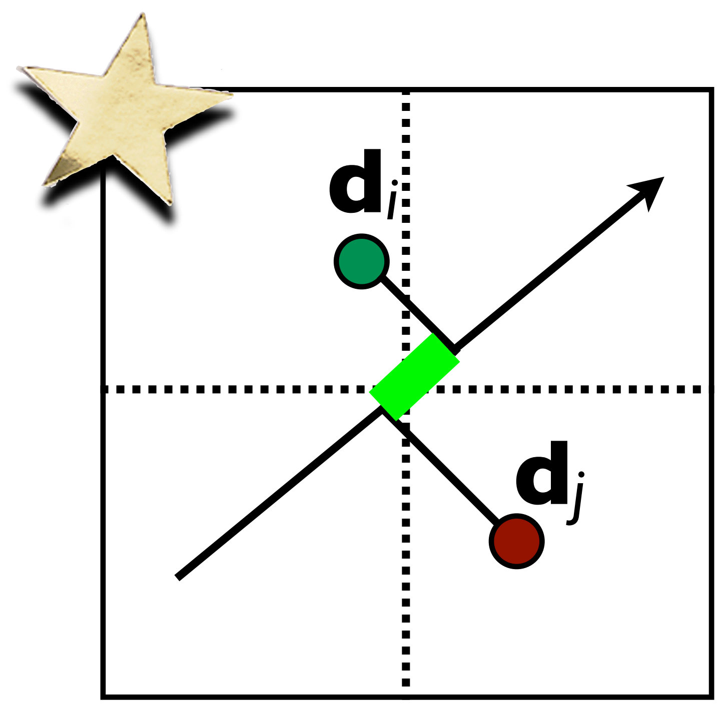
(q, d_i, d_j)

(q, d_i, d_j)

(q, d_i, d_j)

(q, d_i, d_j)

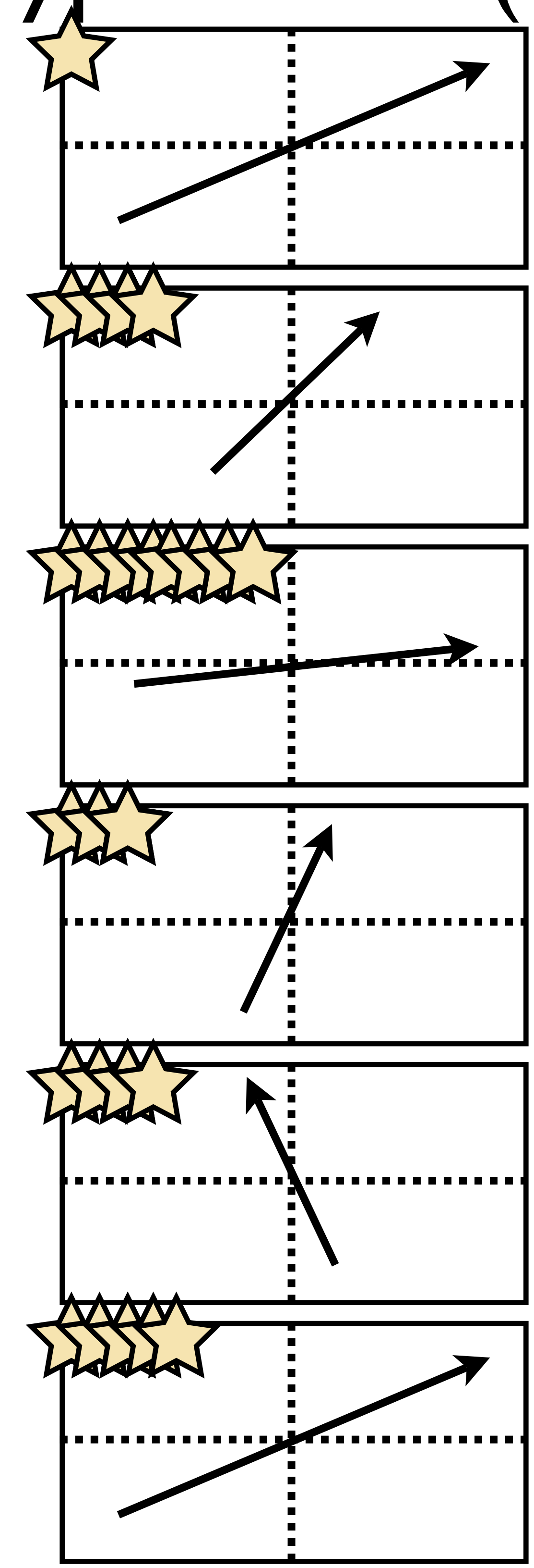
⋮

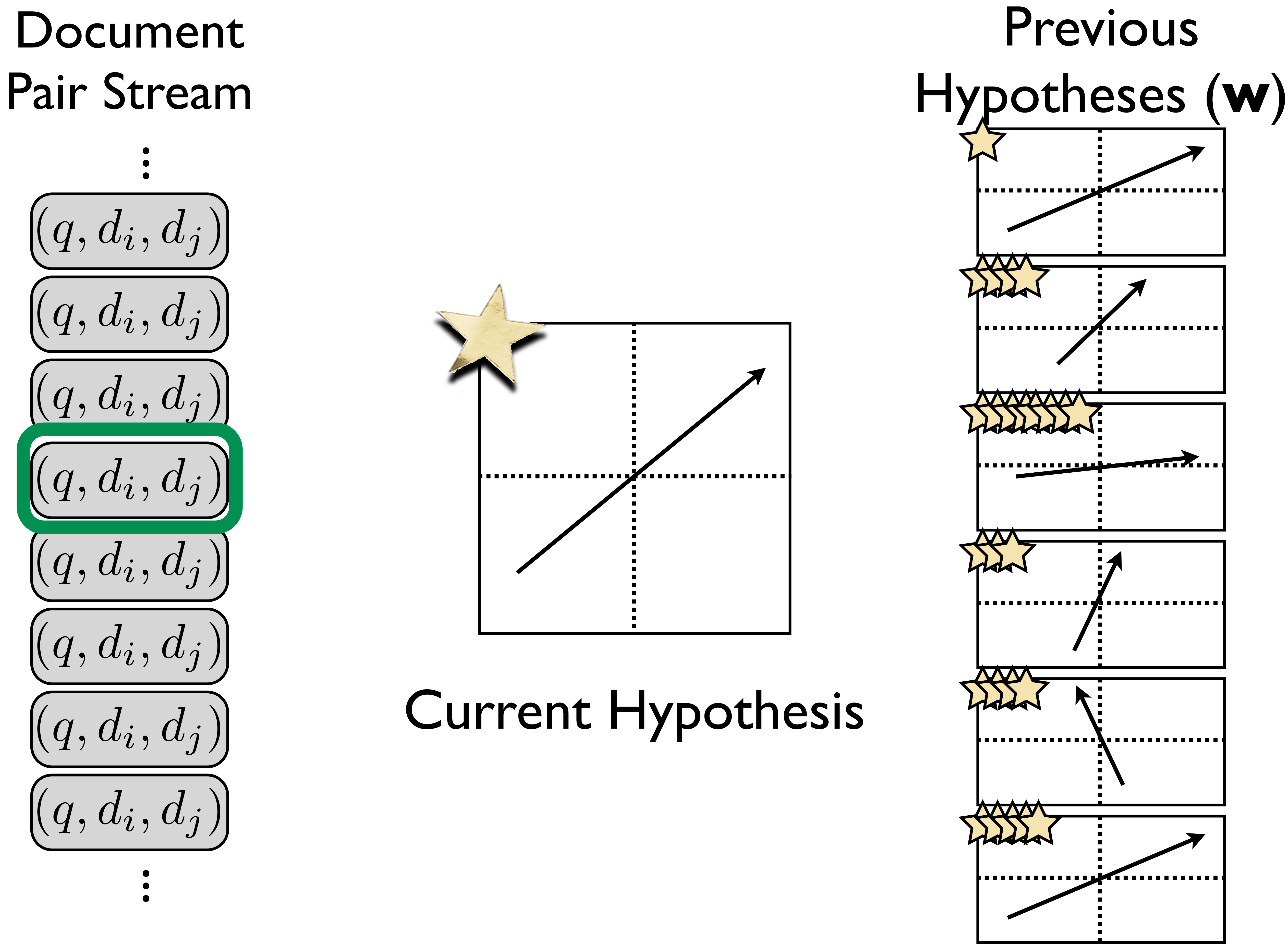


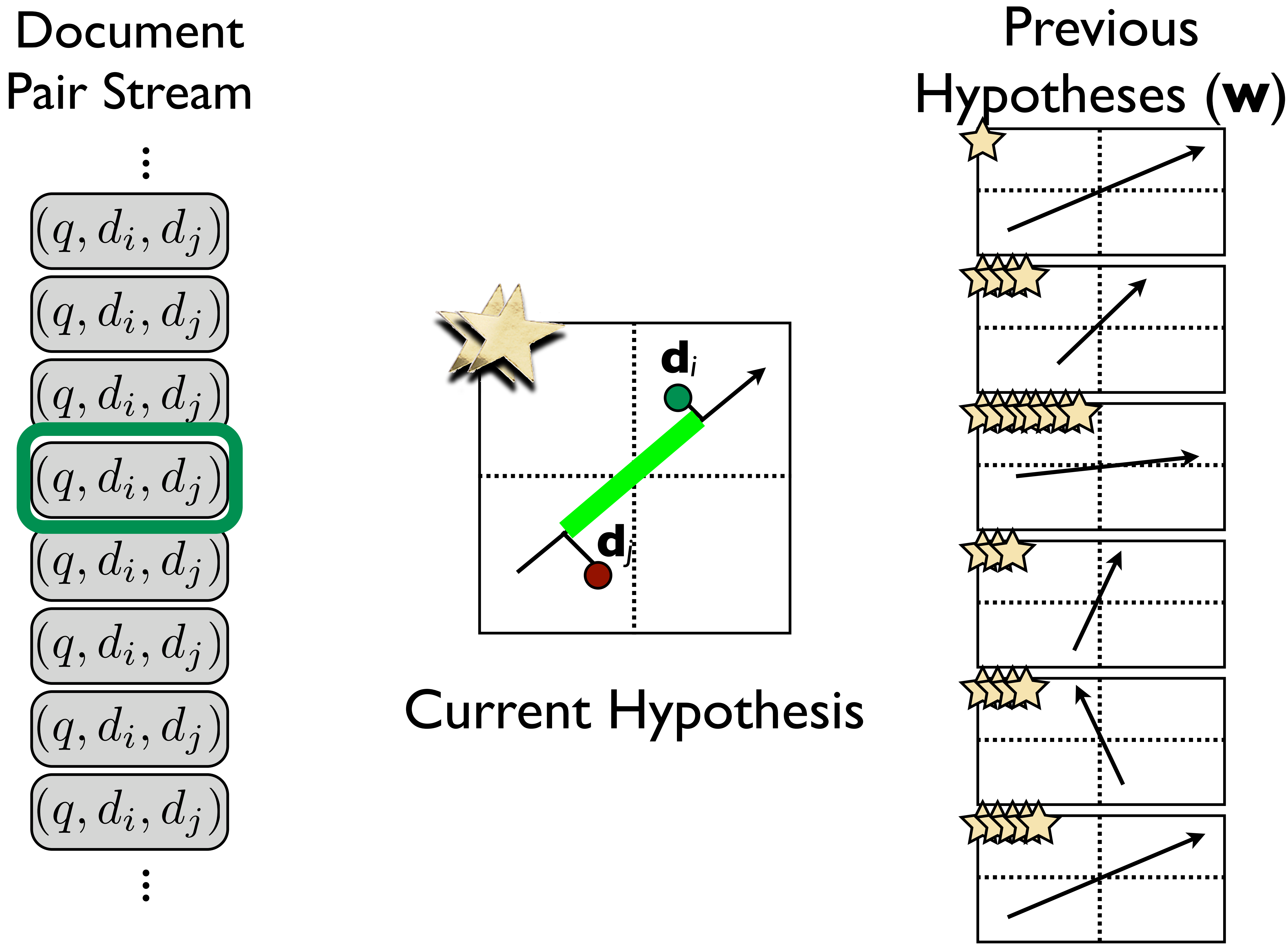
Current Hypothesis

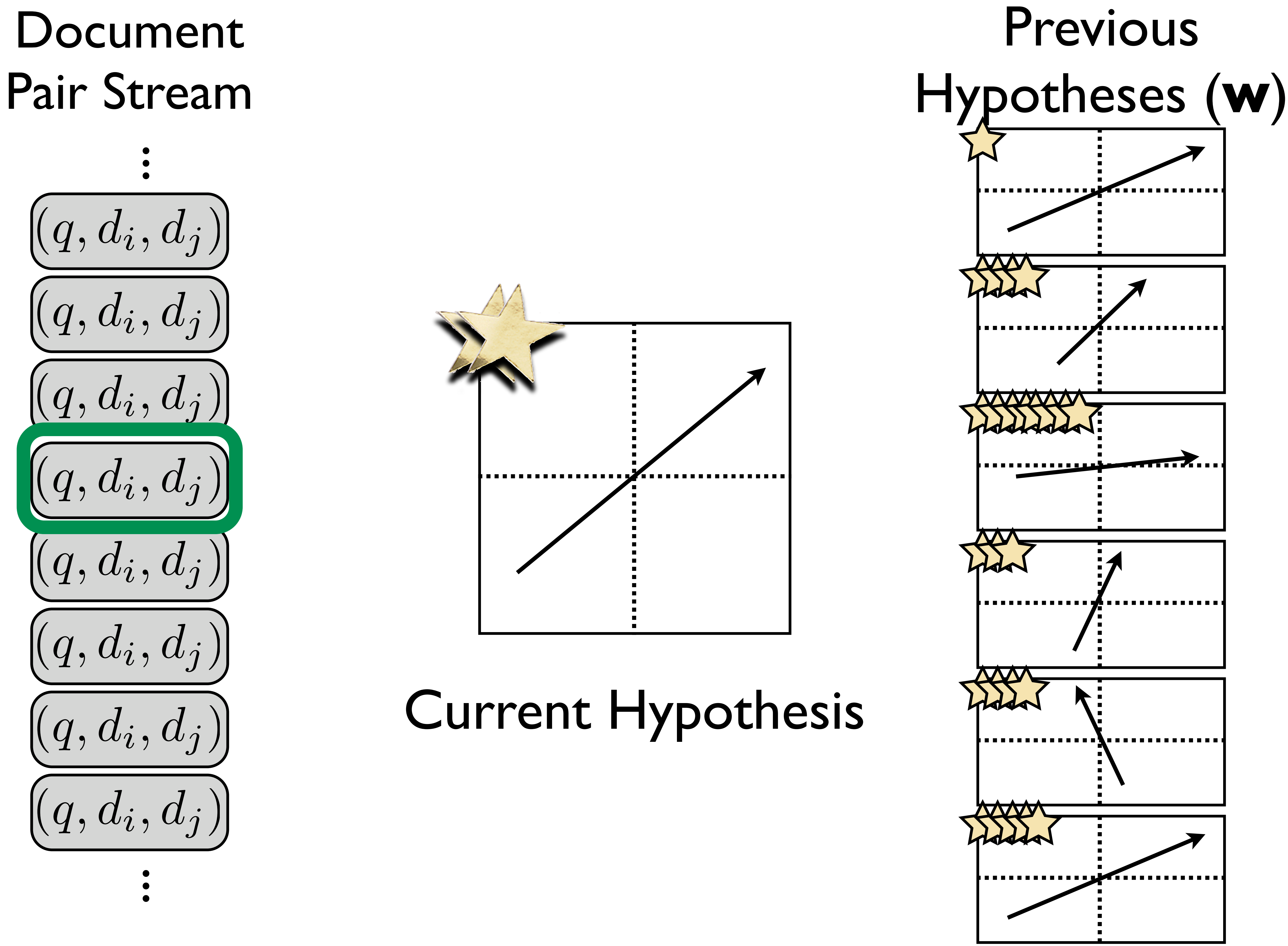
+ $s(d_i) - s(d_j) > 0$

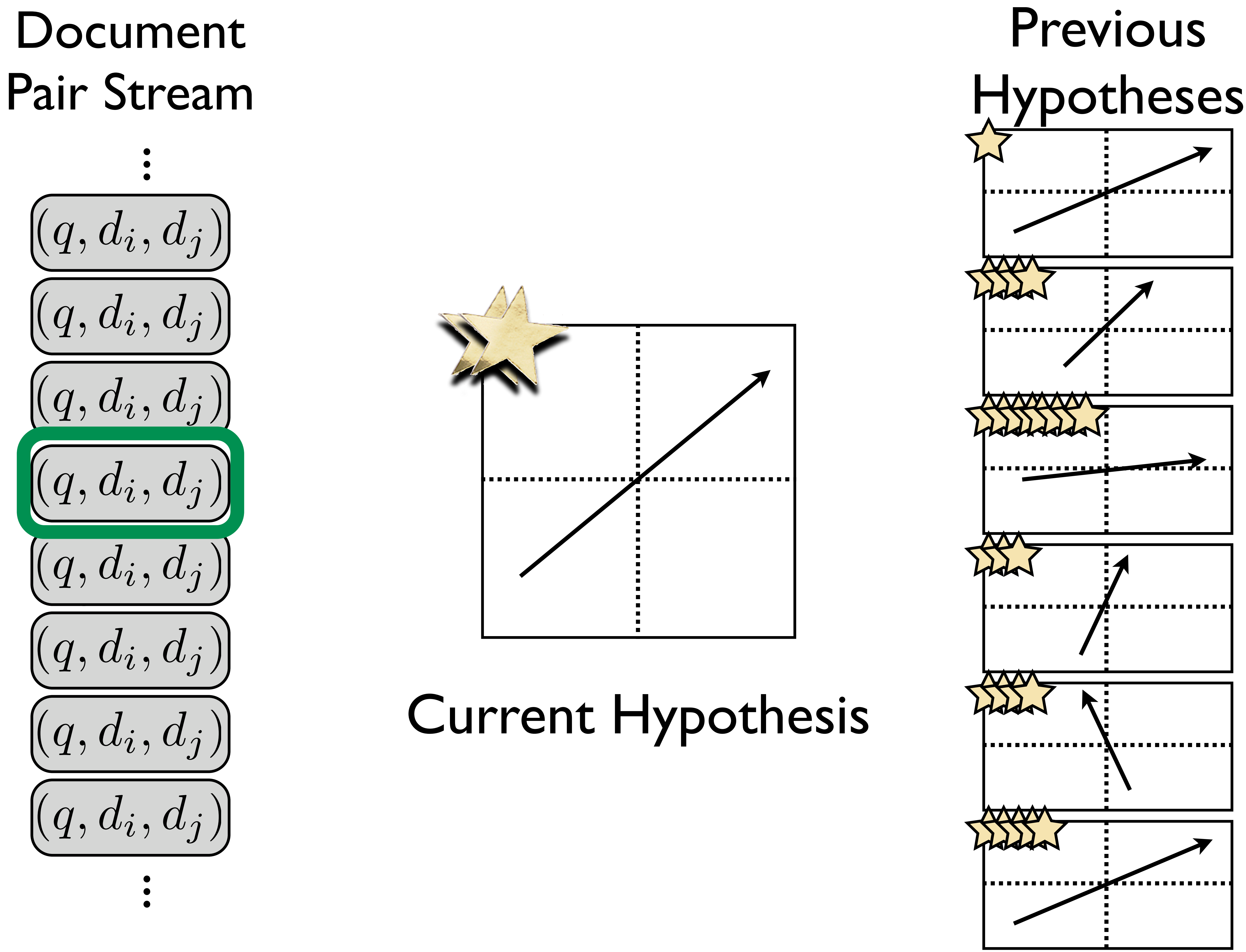
Previous Hypotheses ($\mathbf{w}$)

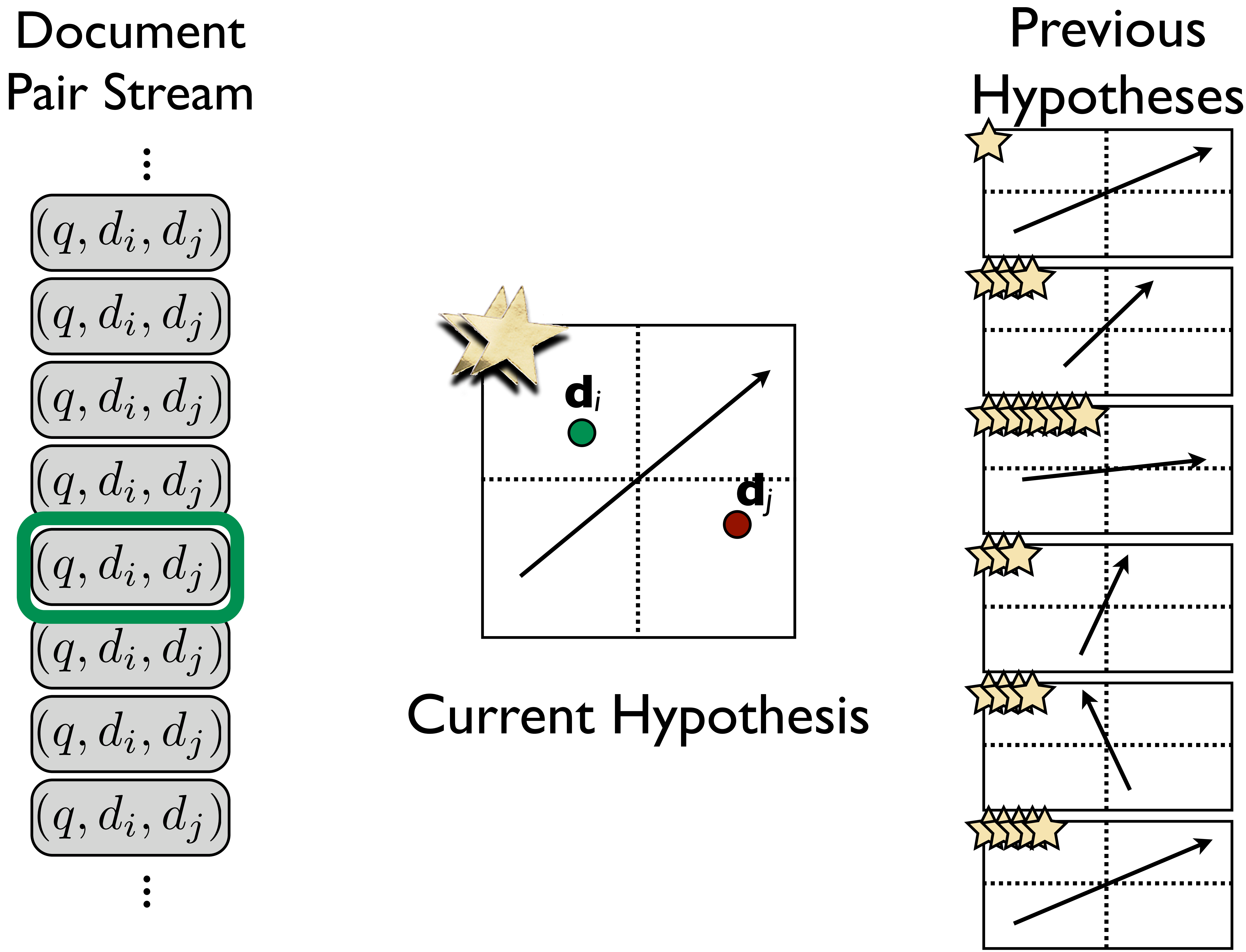



Document Pair Stream
(q, d_i, d_j)
Current Hypothesis
Previous Hypotheses ($\mathbf{w}$)

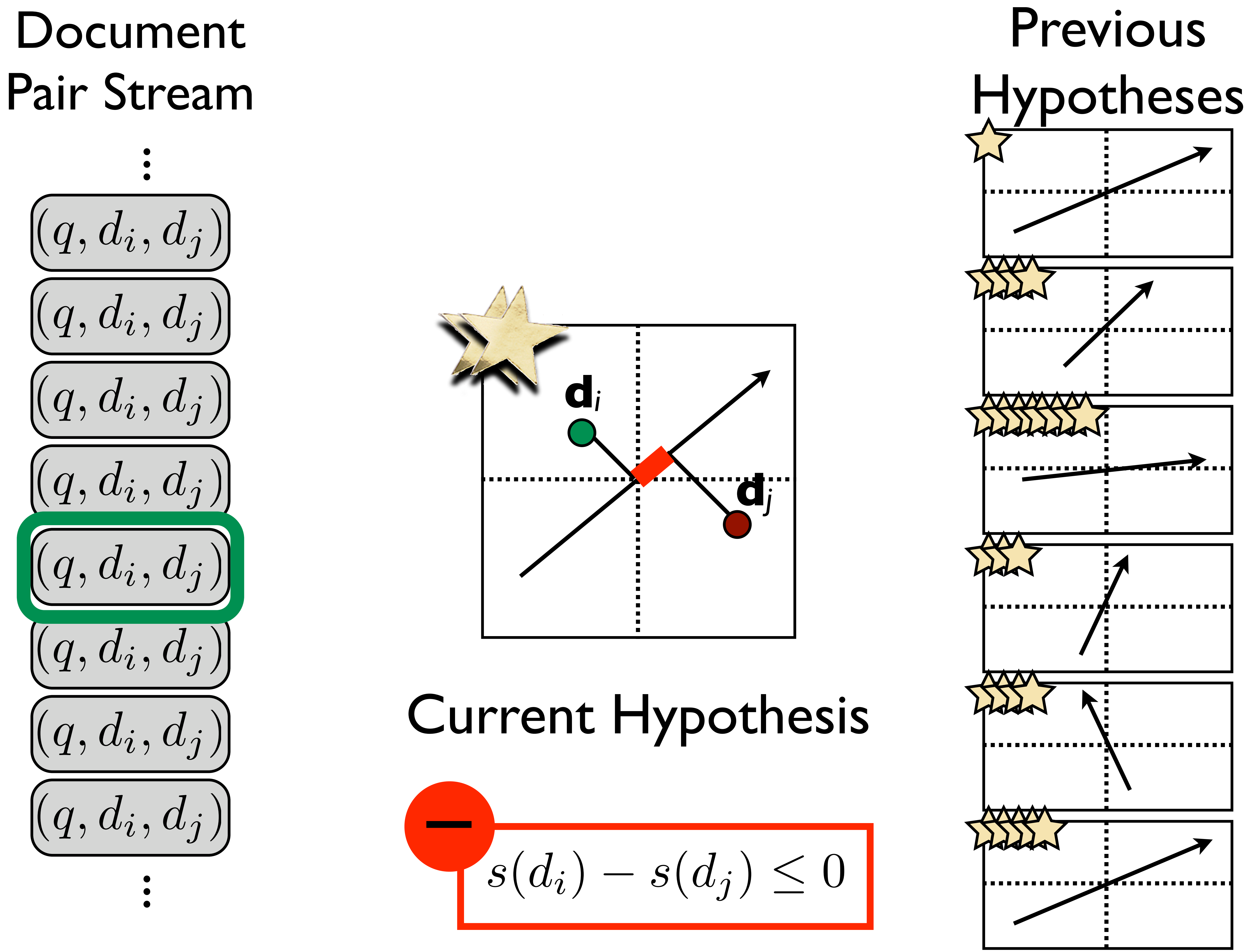

Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
Previous Hypotheses ($\mathbf{w}$)

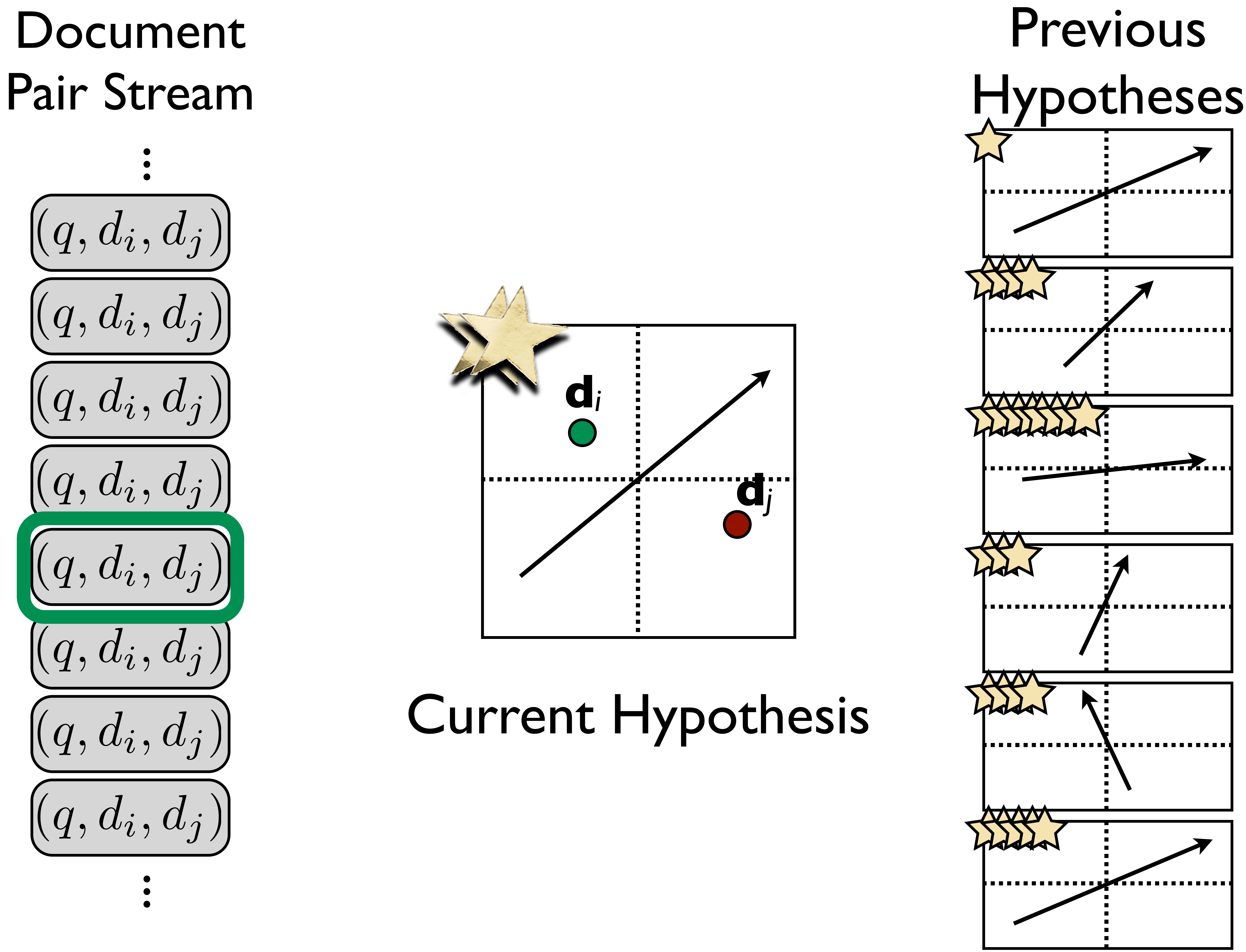

Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
Current Hypothesis
Previous Hypotheses ($\mathbf{w}$)

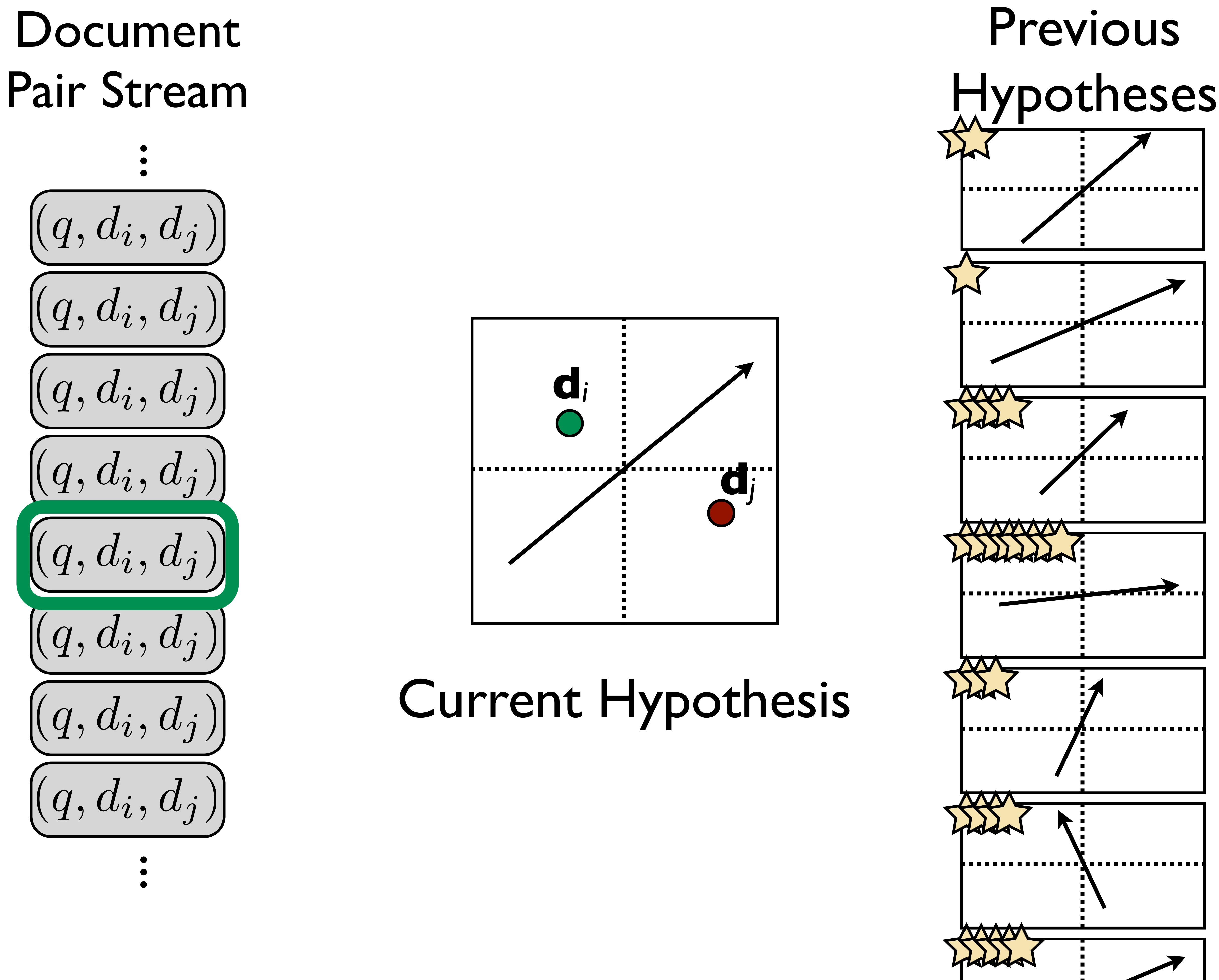

Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
Current Hypothesis
Previous Hypotheses



Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
Previous Hypotheses

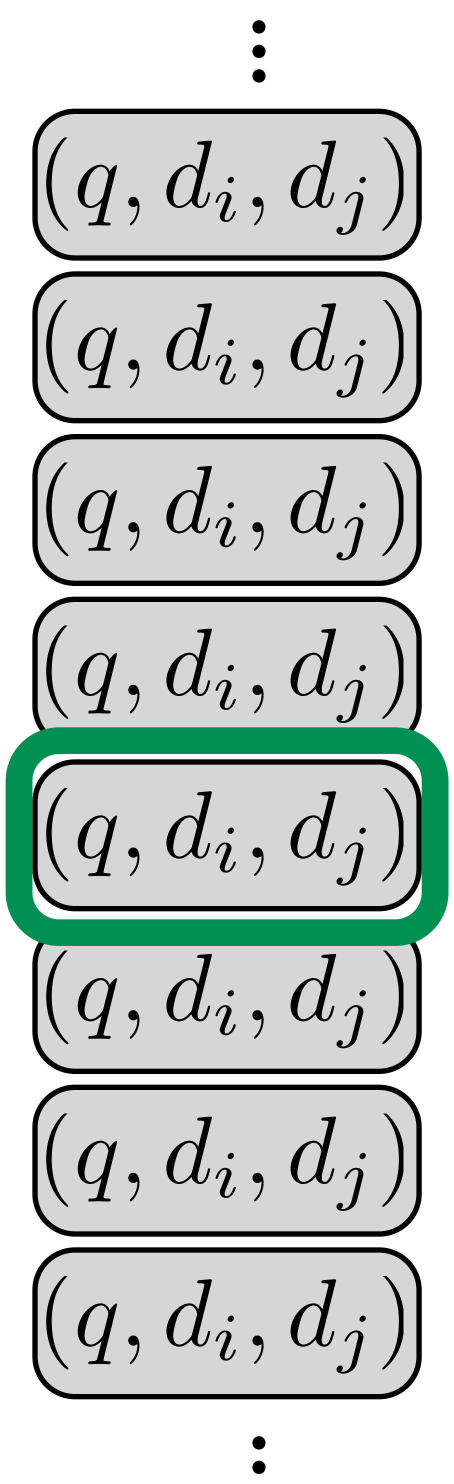

Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
$s(d_i) - s(d_j) \leq 0$
Previous Hypotheses

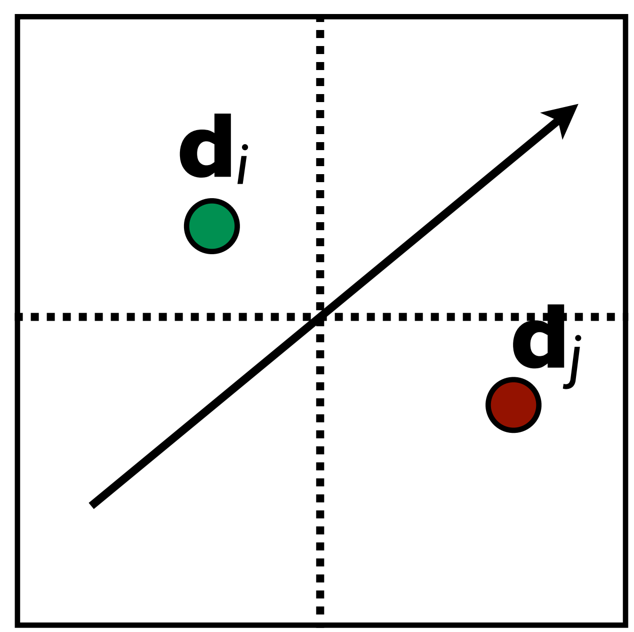

Document
Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
Previous
Hypotheses



Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
Previous Hypotheses

Document Pair Stream

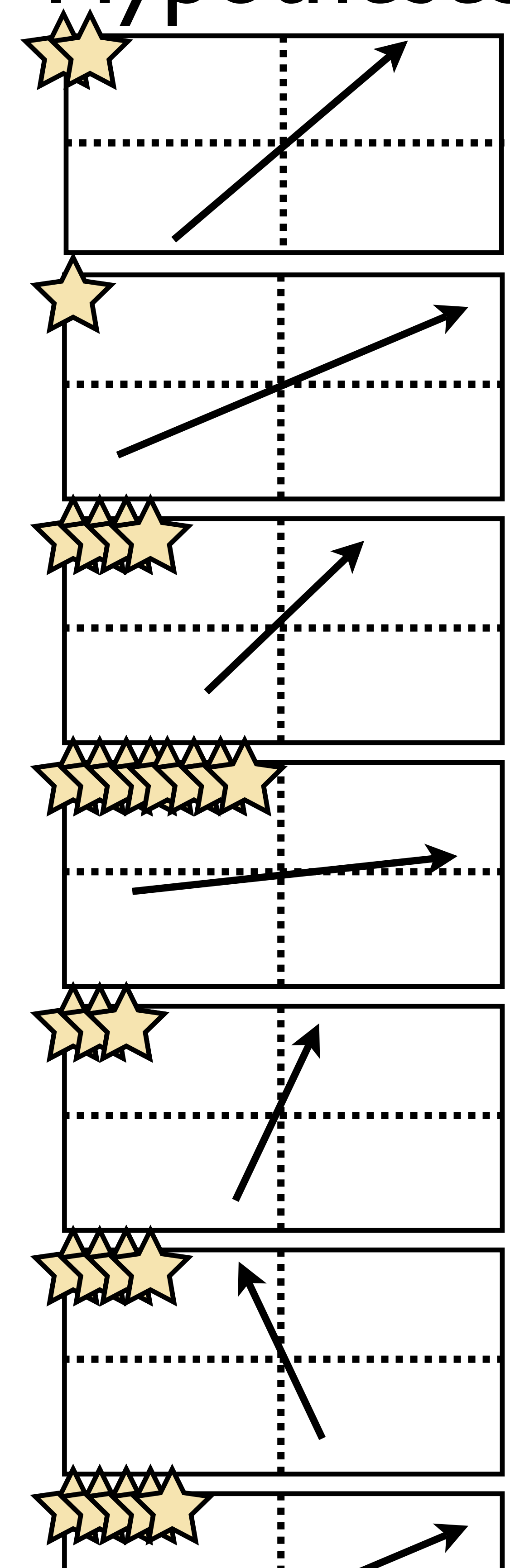


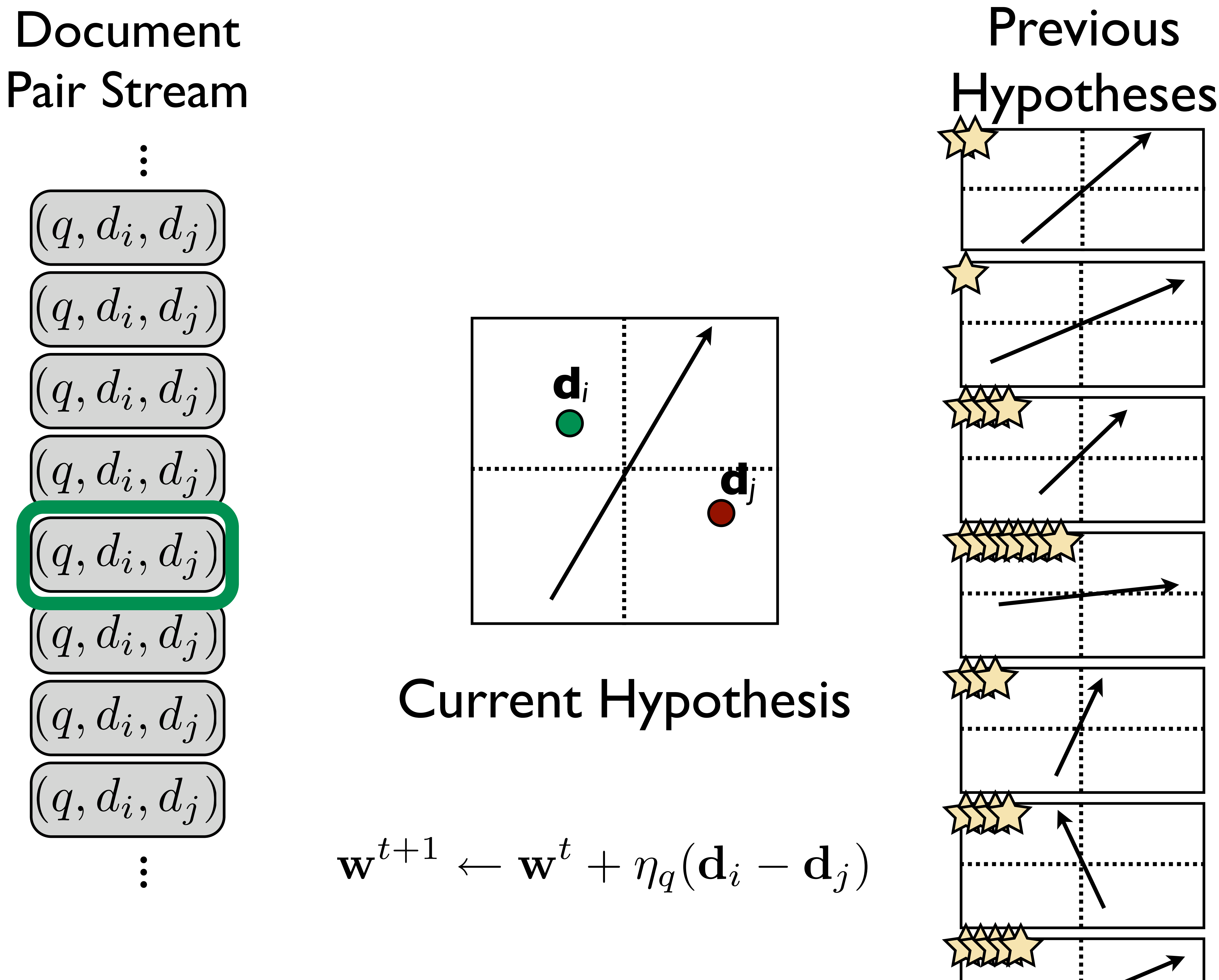


Current Hypothesis

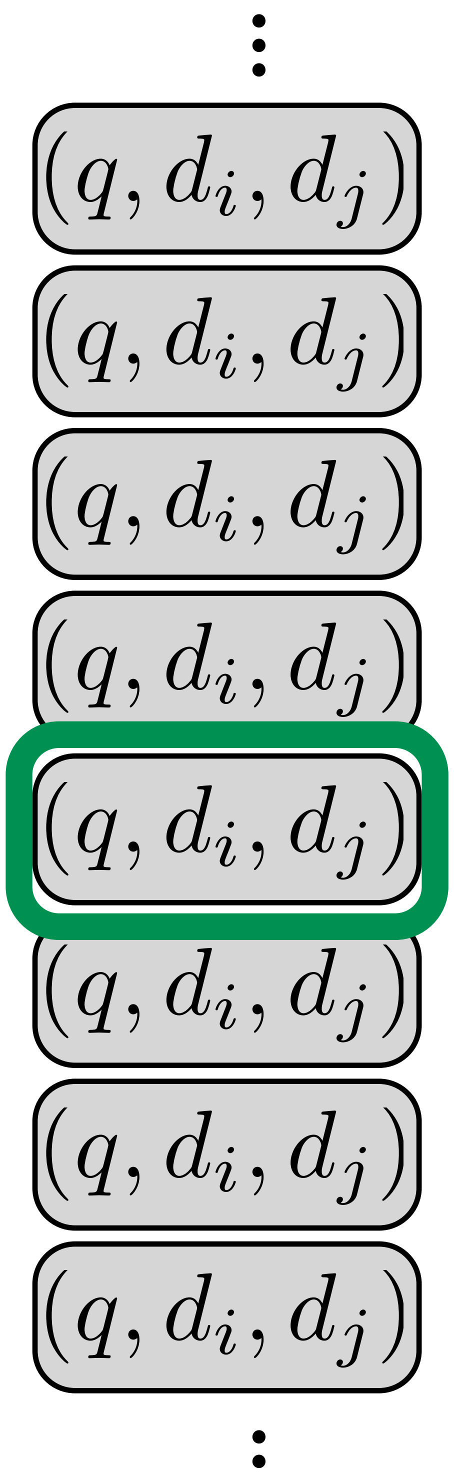
$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \eta_q(\mathbf{d}_i - \mathbf{d}_j)$$

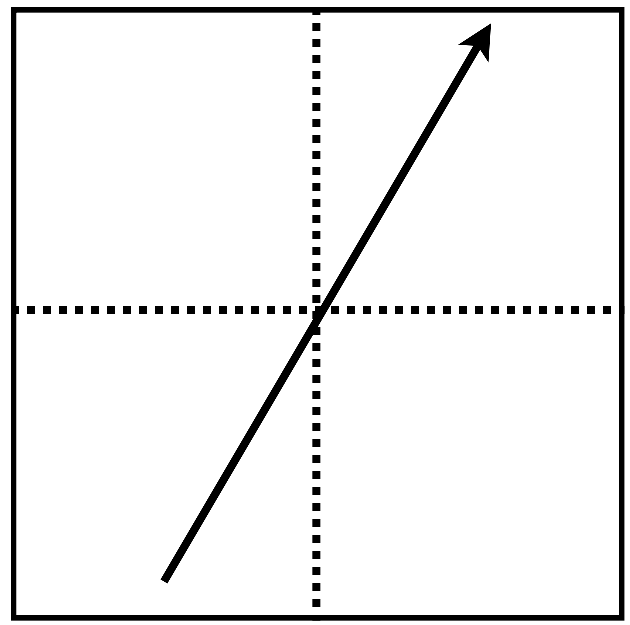
Previous Hypotheses




Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
$\mathbf{d}_i$
$\mathbf{d}_j$
Current Hypothesis
$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^{t} + \eta_q(\mathbf{d}_i - \mathbf{d}_j)$
Previous Hypotheses

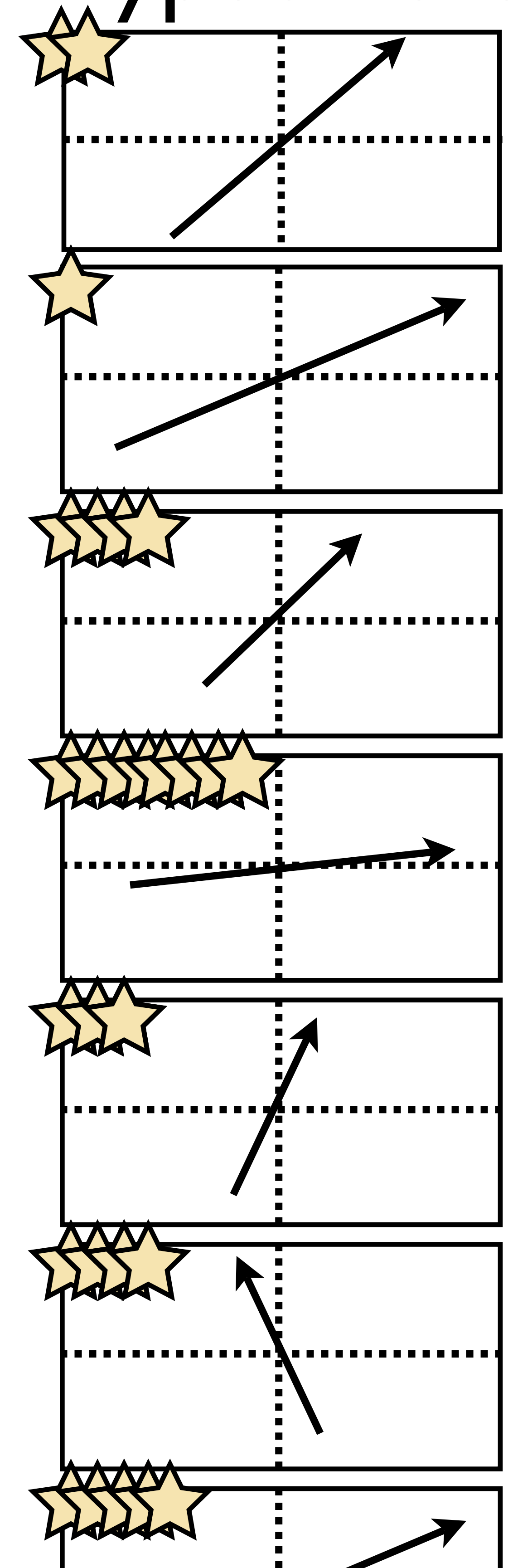
Document Pair Stream

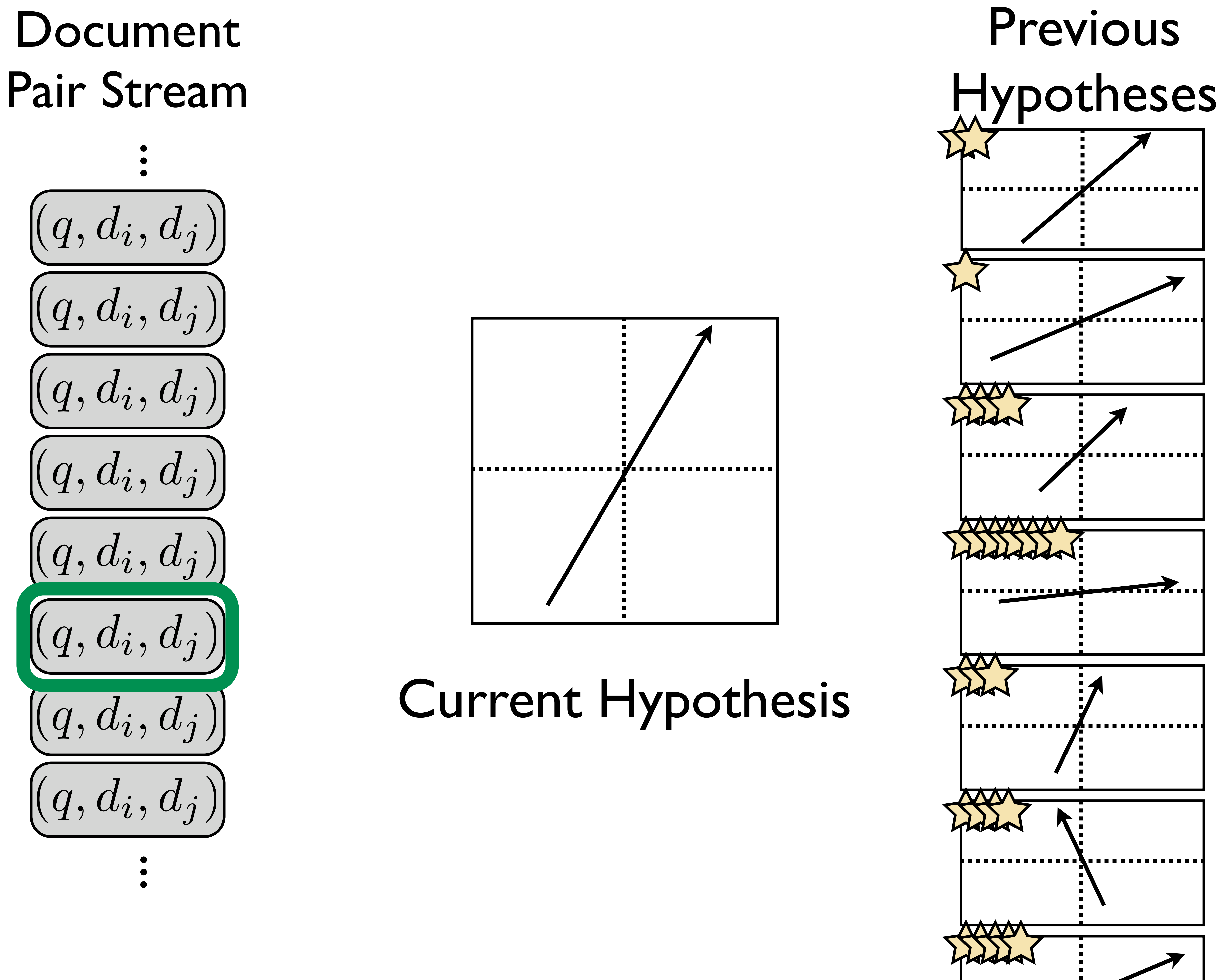




Current Hypothesis

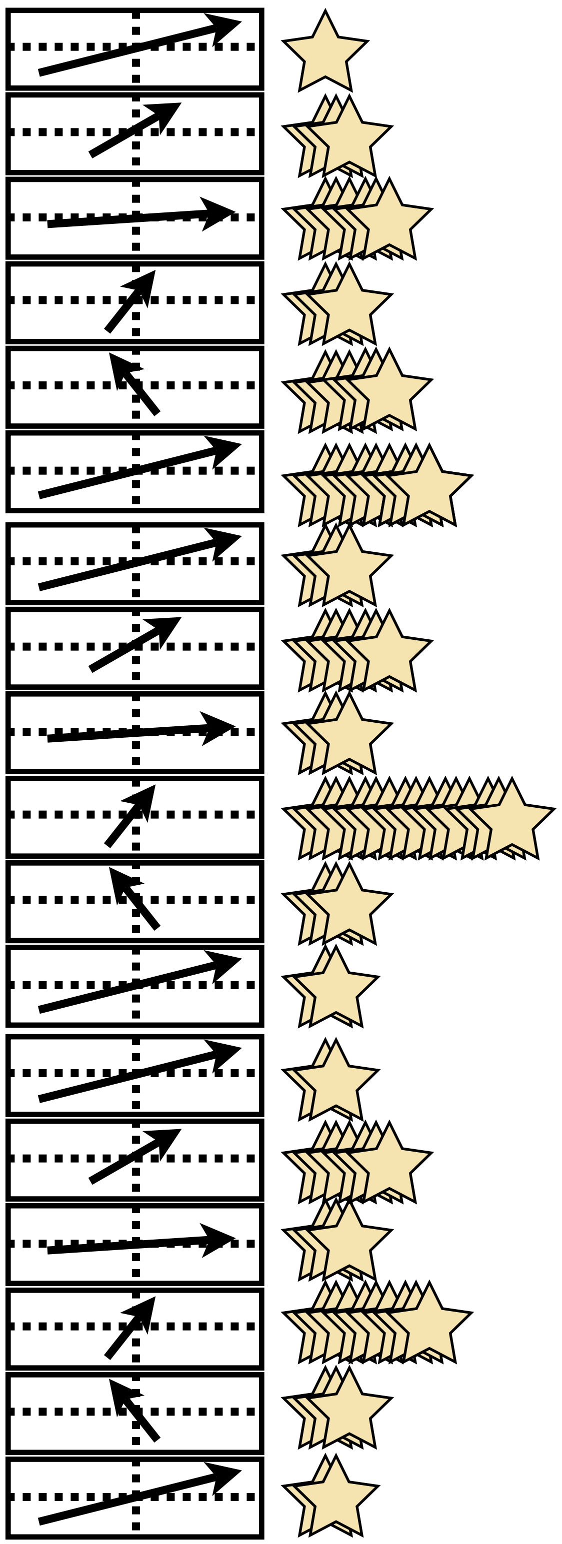
Previous Hypotheses




Document Pair Stream
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
(q, d_i, d_j)
Current Hypothesis
Previous Hypotheses

Perceptron Variants

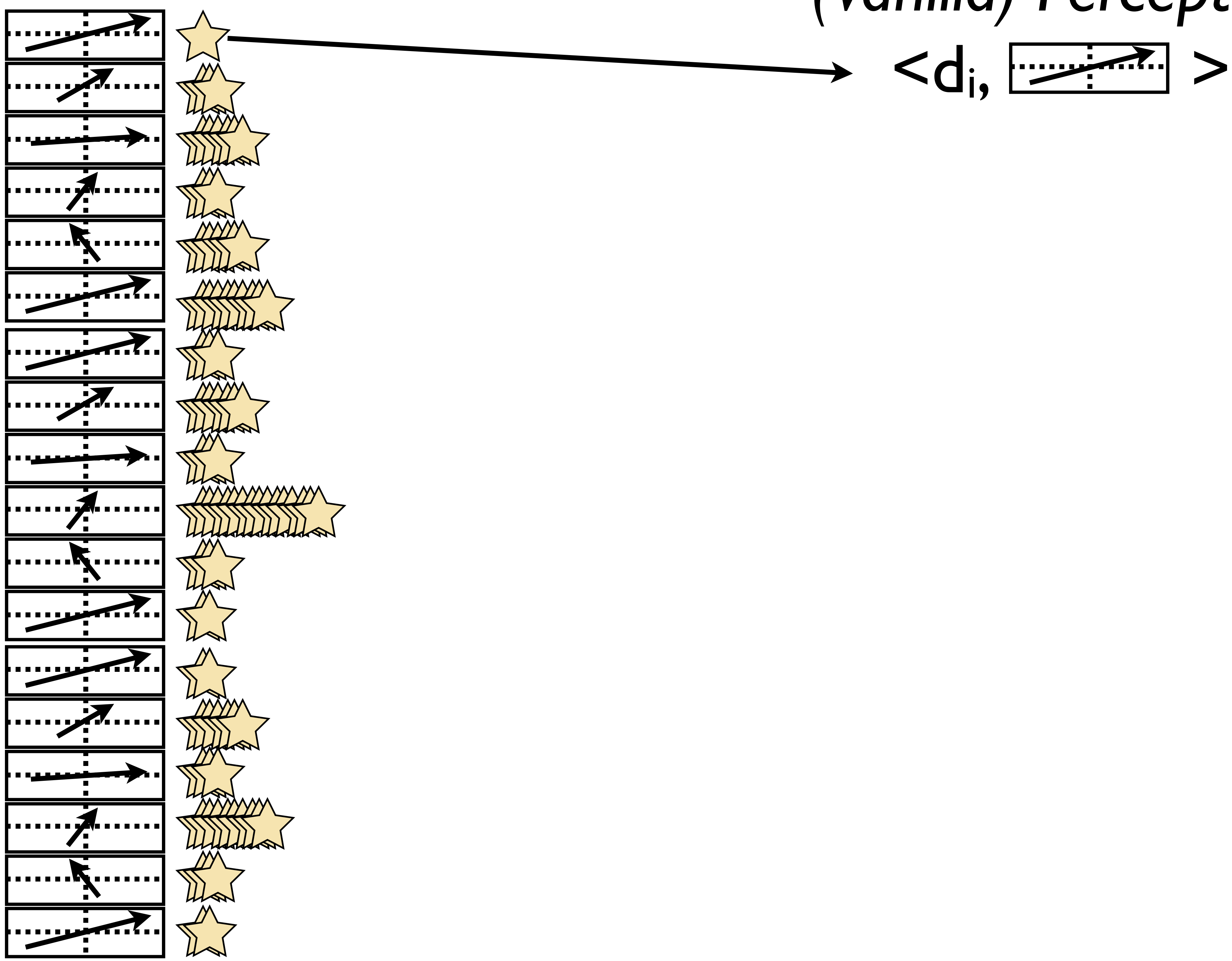
Previous Hypotheses



Perceptron Variants

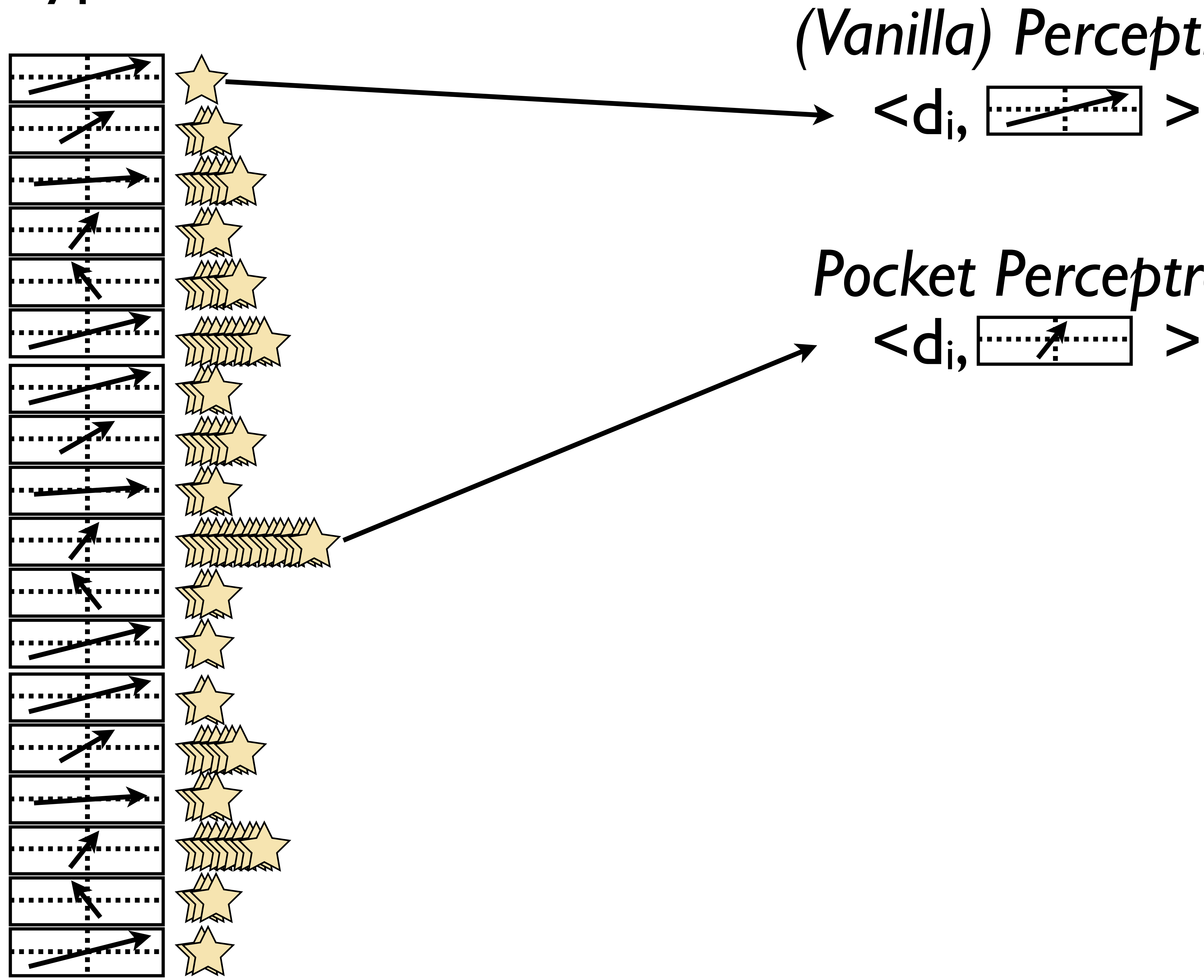
Previous Hypotheses

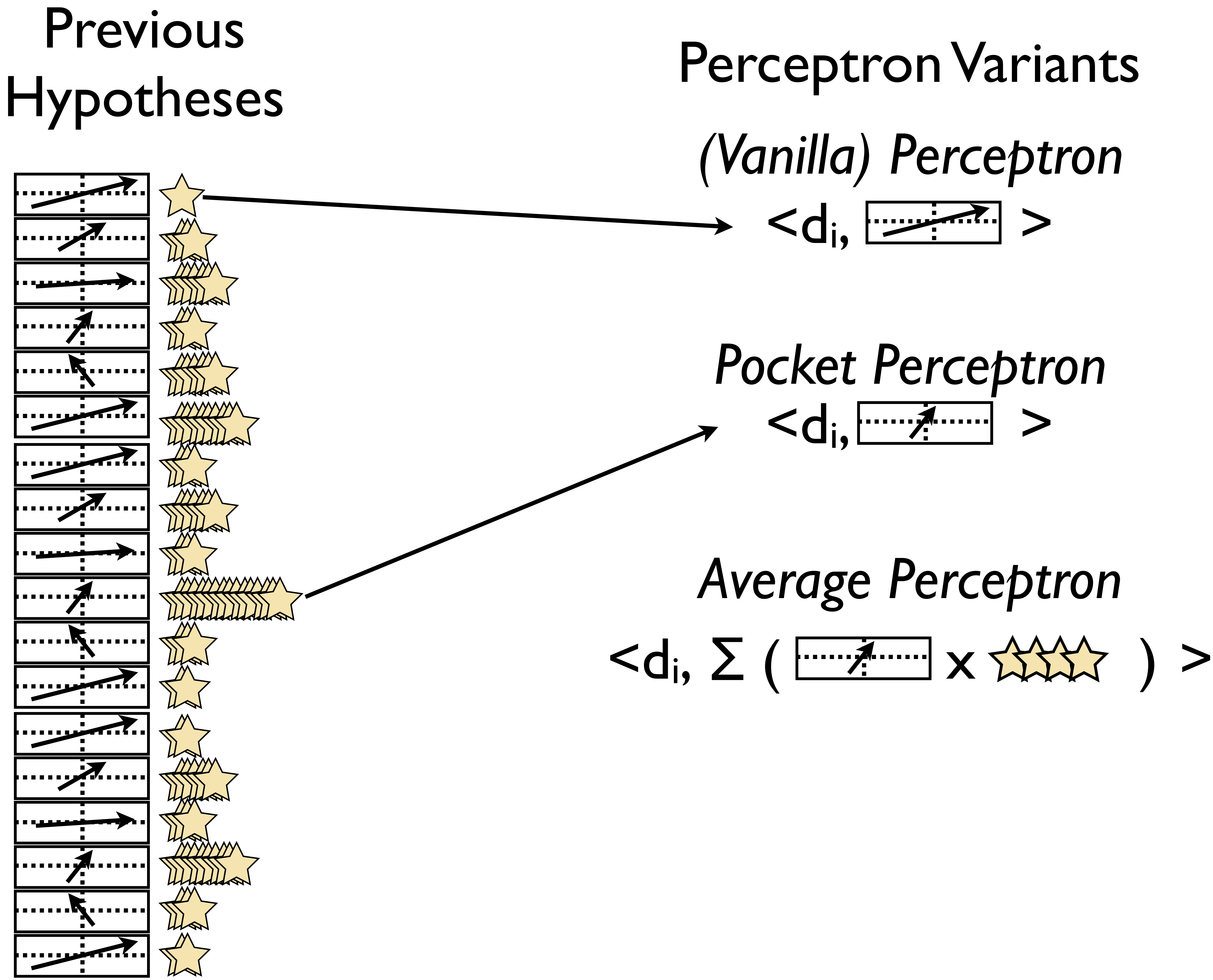
(Vanilla) Perceptron

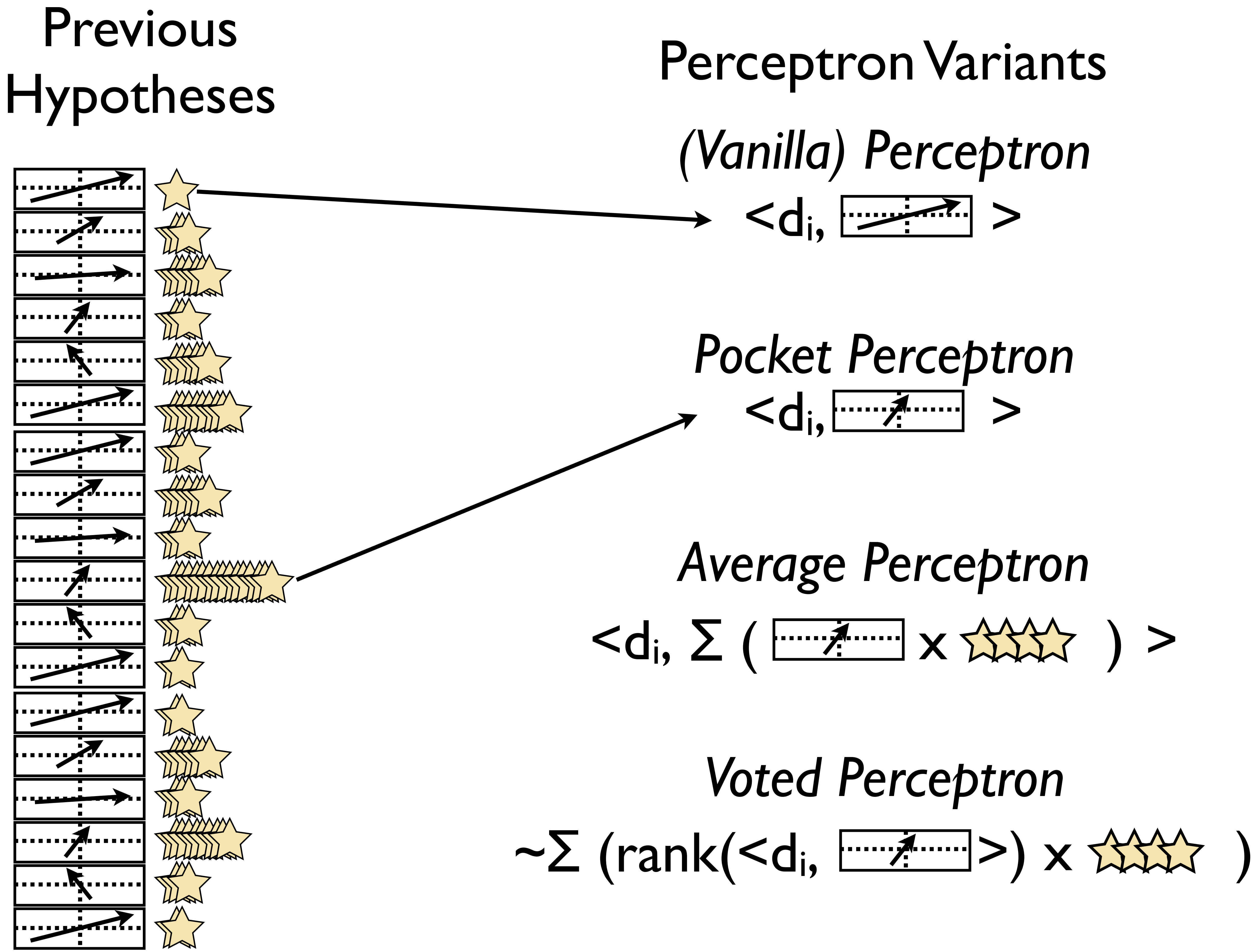


Perceptron Variants

Previous Hypotheses

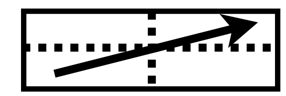



Previous Hypotheses
Perceptron Variants
(Vanilla) Perceptron
<d_i, >
Pocket Perceptron
<d_i, >
Average Perceptron
<d_i, Σ (x) >

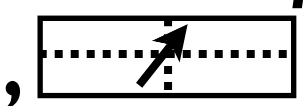

Previous Hypotheses
Perceptron Variants
(Vanilla) Perceptron
<di, >
Pocket Perceptron
<di, >
Average Perceptron
<di, Σ (x) >
Voted Perceptron
~Σ (rank(<di, >) x)

Perceptron Variants

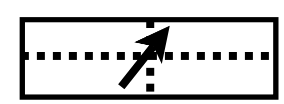
(Vanilla) Perceptron

$<d_i,$  $>$

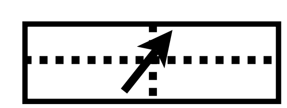
Pocket Perceptron

$<d_i,$  $>$

Average Perceptron

$<d_i, \Sigma\ ($  $\times$  $)\ >$

Voted Perceptron

$\sim\Sigma\ (\text{rank}(<d_i,$  $>) \times$  $)$

Possible poor final hypothesis & unstable

Perceptron Variants

(Vanilla) Perceptron

$< d_i, \quad >$

Pocket Perceptron

$< d_i, \quad >$

Average Perceptron

$< d_i, \Sigma (\quad \text{x} \quad) >$

Voted Perceptron

$\sim\Sigma \; (\text{rank}(< d_i, \quad >) \; \text{x} \quad)$

Possible poor final hypothesis & unstable

Better hypothesis, but still unstable

Perceptron Variants

(Vanilla) Perceptron

$<d_i, \quad >$

Pocket Perceptron

$<d_i, \quad >$

Average Perceptron

$<d_i, \Sigma \, (\quad \times \quad) >$

Voted Perceptron

$\sim\Sigma \, (\mathrm{rank}(<d_i, \quad >) \times \quad)$

Possible poor final hypothesis & unstable

Better hypothesis, but still unstable

More stable, but slow convergence

Perceptron Variants

(Vanilla) Perceptron

$< d_i, \quad >$

Pocket Perceptron

$< d_i, \quad >$

Average Perceptron

$< d_i, \Sigma (\quad x \quad) >$

Voted Perceptron

$\sim\Sigma$ (rank($< d_i, \quad >$) x $\quad$)

Possible poor final hypothesis & unstable

Better hypothesis, but still unstable

More stable, but slow convergence

More stable, slow convergence & testing

Perceptron Variants

(Vanilla) Perceptron

$<d_i, \quad >$

Pocket Perceptron

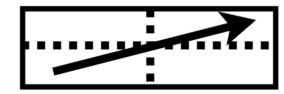
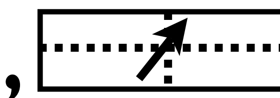
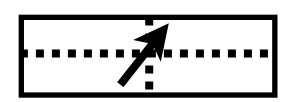
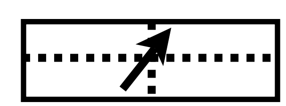
$<d_i, \quad >$

Average Perceptron

$<d_i, \Sigma (\quad x \quad) >$

Voted Perceptron

$\sim\Sigma (rank(<d_i, \quad >) \ x \quad)$

Space Complexity	Test Time Complexity	Perceptron Variants
O(1)	O(1)	*(Vanilla) Perceptron* <d_i,  >
O(1)	O(1)	*Pocket Perceptron* <d_i,  >
O(1)	O(1)	*Average Perceptron* <d_i, Σ ( x ) >
O(N)	O(N)	*Voted Perceptron* ~Σ (rank(<d_i,  >) x )

Space Complexity	Test Time Complexity	Perceptron Variants
O(1)	O(1)	*(Vanilla) Perceptron* $< d_i, \quad >$
O(1)	O(1)	*Pocket Perceptron* $< d_i, \quad >$
O(1)		*Average Perceptron* $d_i, \Sigma (\quad \times \quad) >$
O(N)	O(N)	*Voted Perceptron* $\sim\Sigma\ (\mathrm{rank}(< d_i, \quad >) \times \quad)$

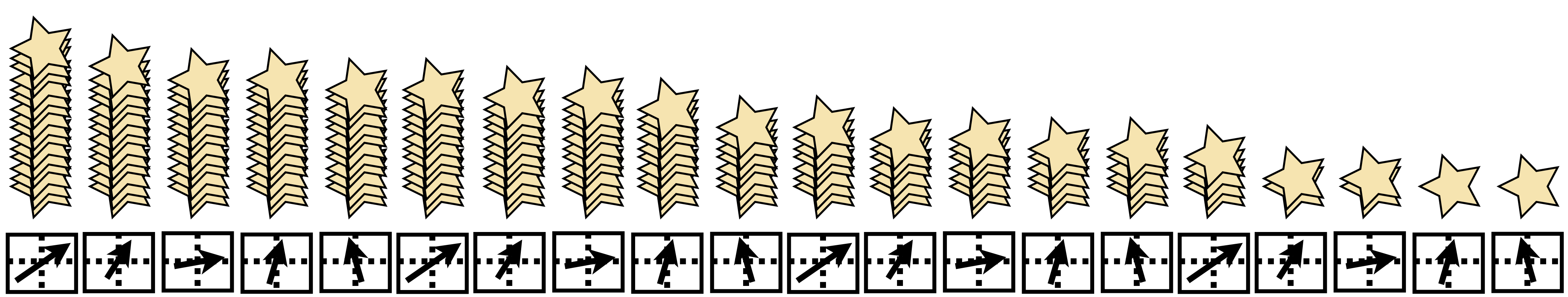
N≈1.5 million
in 50 iterations
on TD2003

Committee Perceptron

- Generalization of Pocket/Average/Voted
- Only use K-best hypotheses

Committee Perceptron

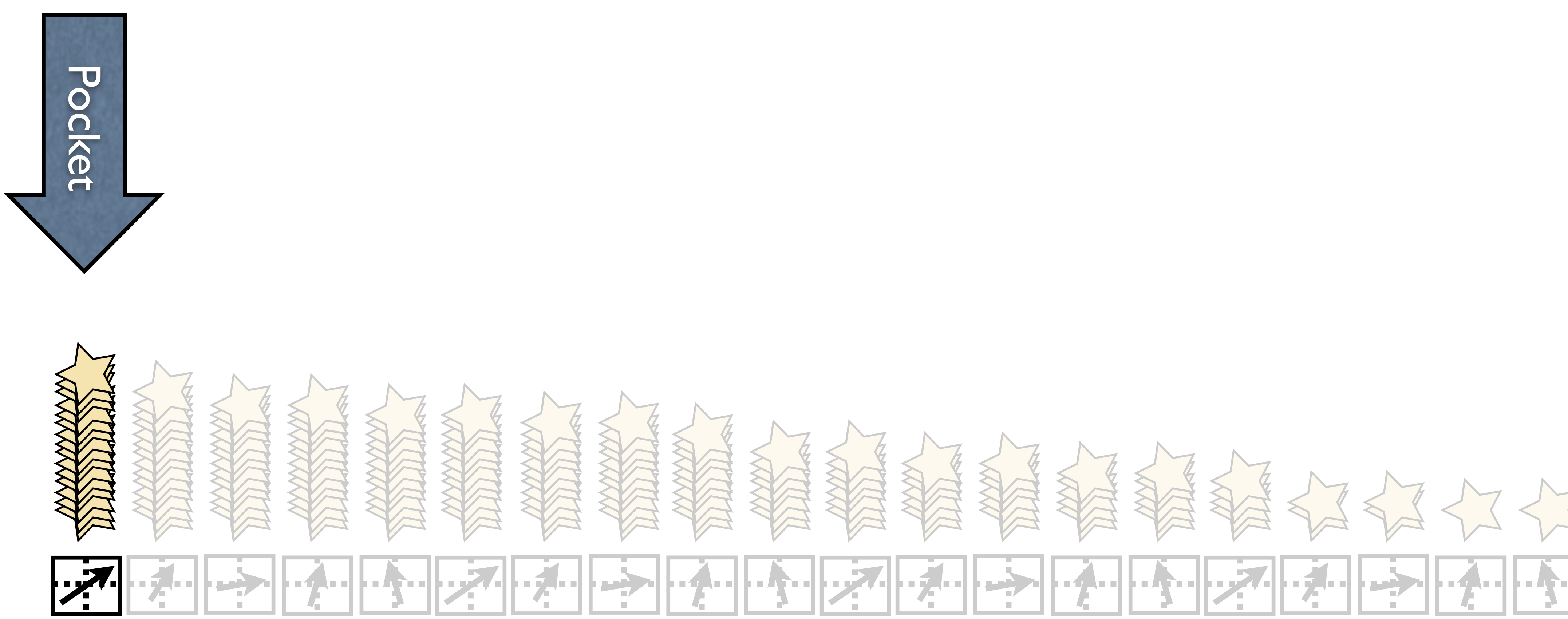
- Generalization of Pocket/Average/Voted
- Only use K-best hypotheses



Committee Perceptron

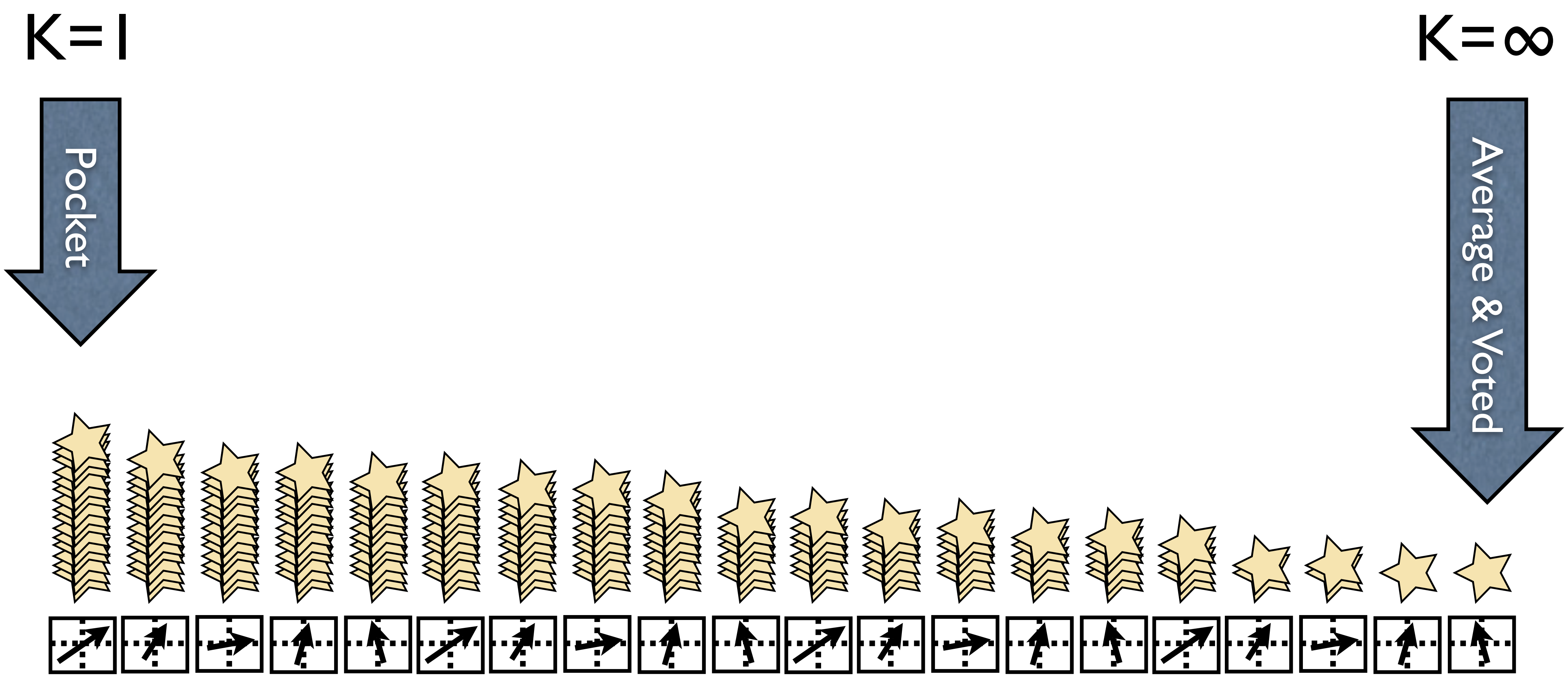
- Generalization of Pocket/Average/Voted
- Only use K-best hypotheses

K=1



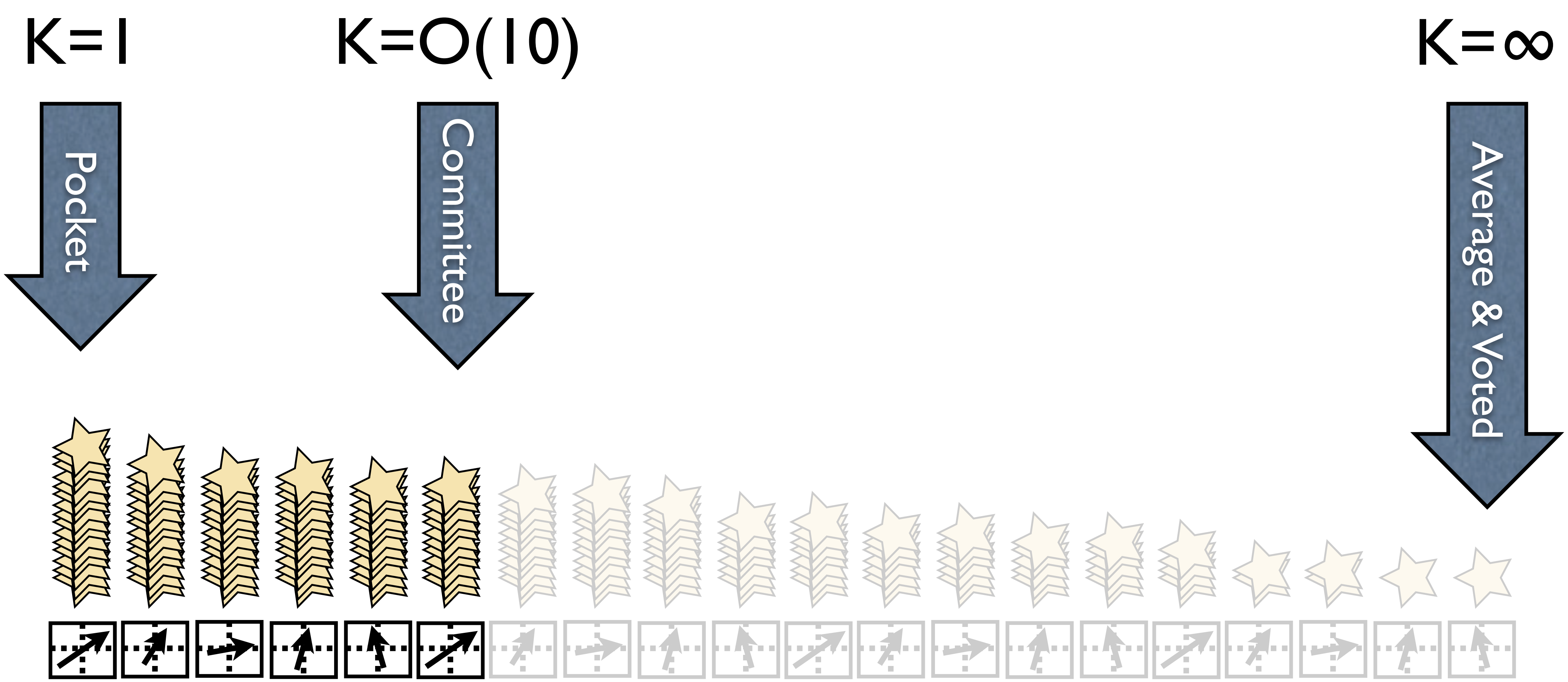
Committee Perceptron

- Generalization of Pocket/Average/Voted
- Only use K-best hypotheses



Committee Perceptron

- Generalization of Pocket/Average/Voted
- Only use K-best hypotheses



Committee Perceptron

- Empirically faster training and better performance than other perceptron variants
- Constant space & time complexity
- Comparable or better performance than baseline algorithms with a fraction of the training time

Test Collection: LETOR Data set

(Liu et. al, 2007)

- 3 standard IR data sets:
  OHSUMED (106 queries / 25 features)
  TREC Topic Distillation 2003 (50 / 44)
  TREC Topic Distillation 2004 (75 / 44)
- Standard feature set: TF, IDF-based features, BM25 scores, PageRank, etc.
- Binary (TREC) and 3-level (OHSUMED) relevance judgements

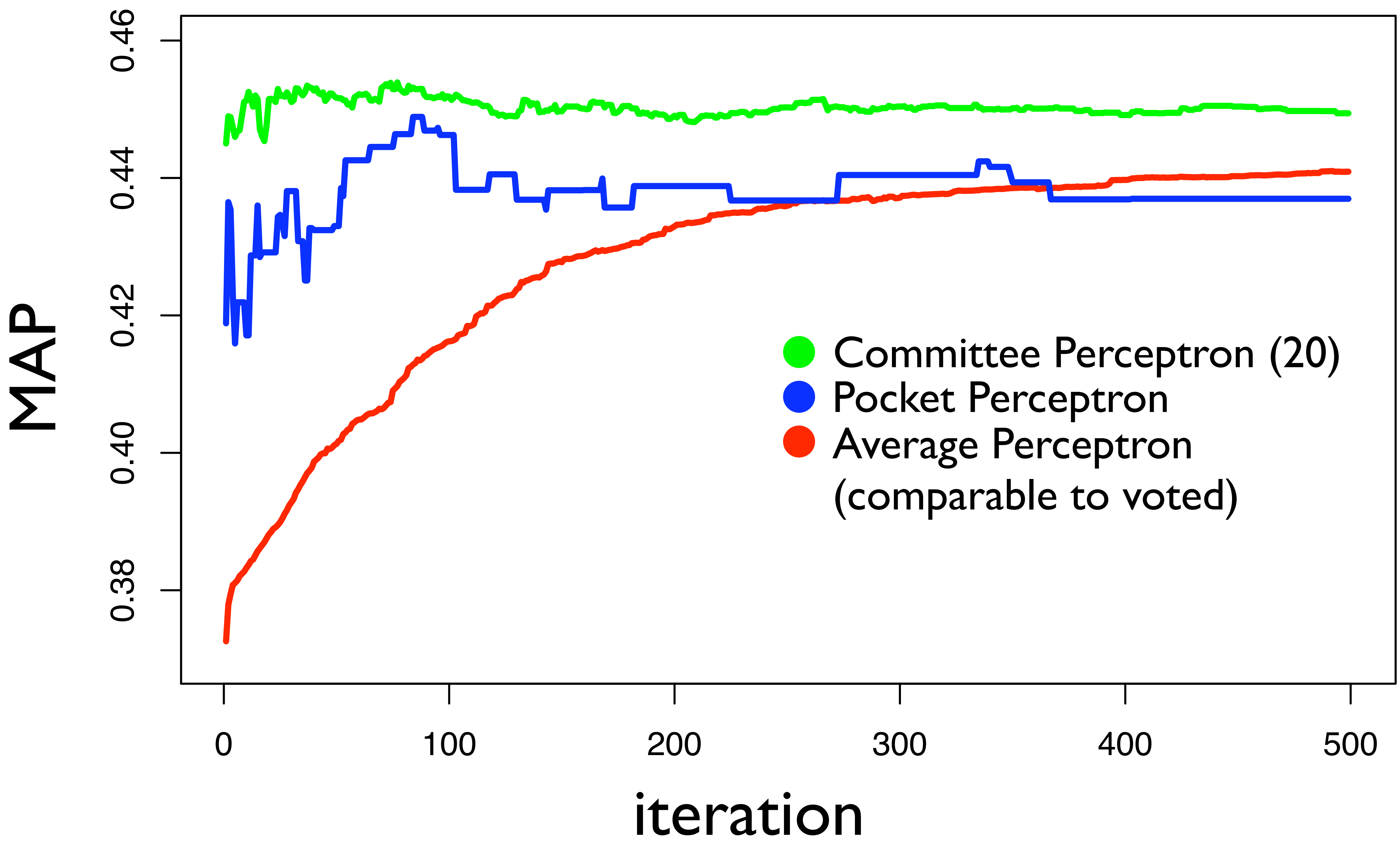
Test Collection: LETOR Data set

(Liu et. al, 2007)

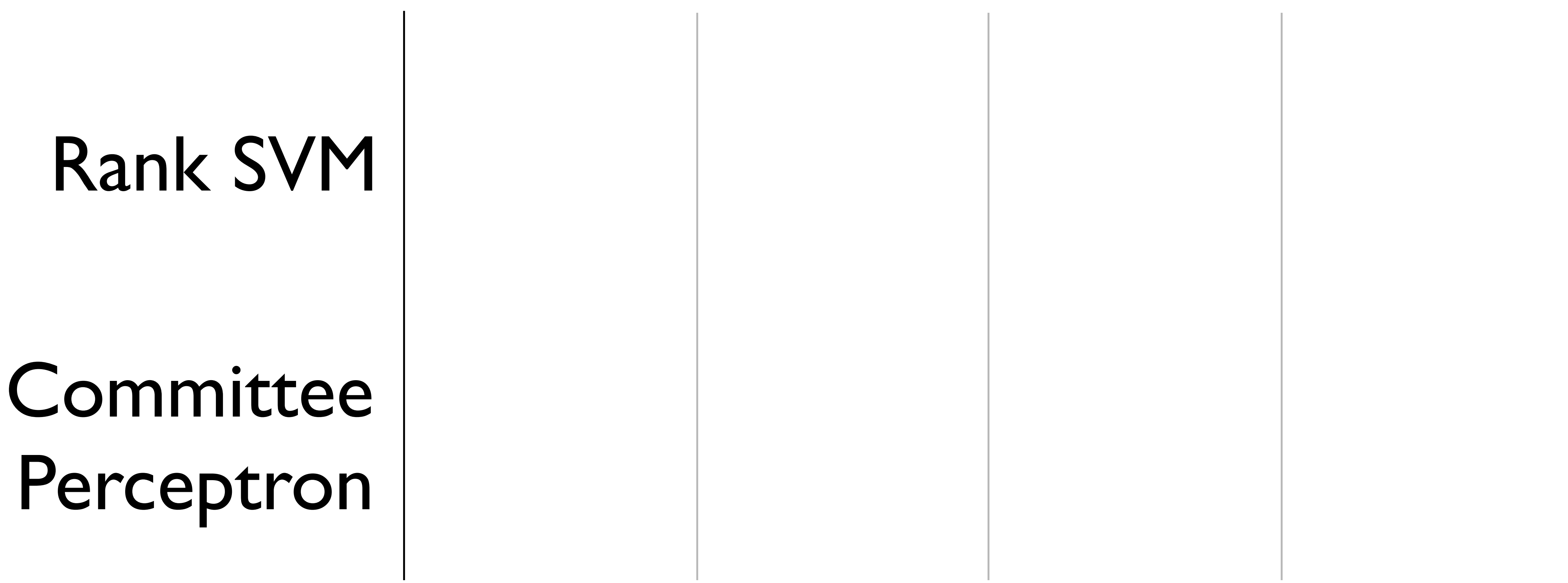
- Two baseline algorithms:
  RankSVM & RankBoost

- Strong baselines (circa early 2007)

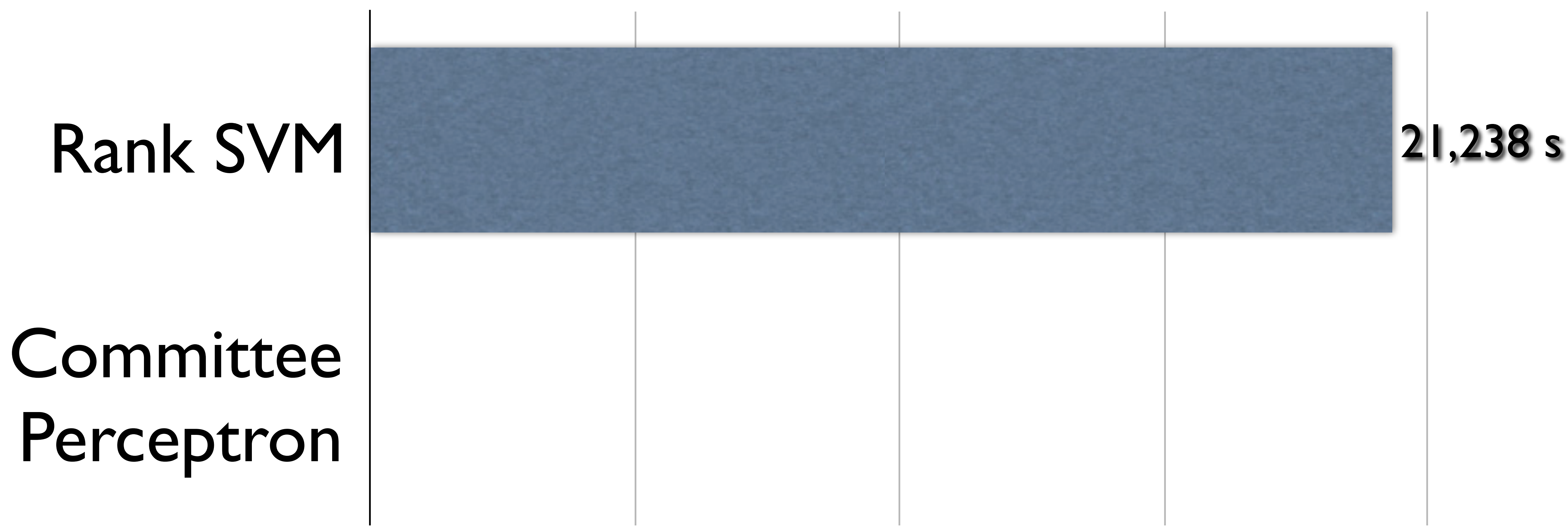
MAP for Perceptron Variants on OHSUMED



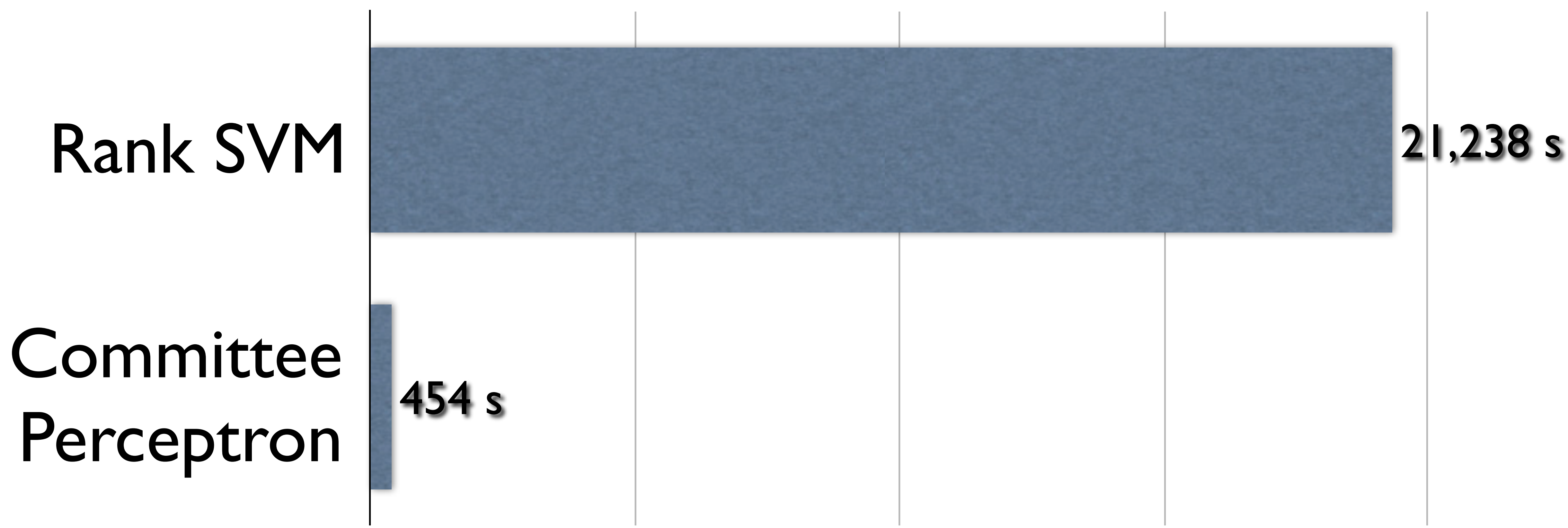
Training Time



Training Time

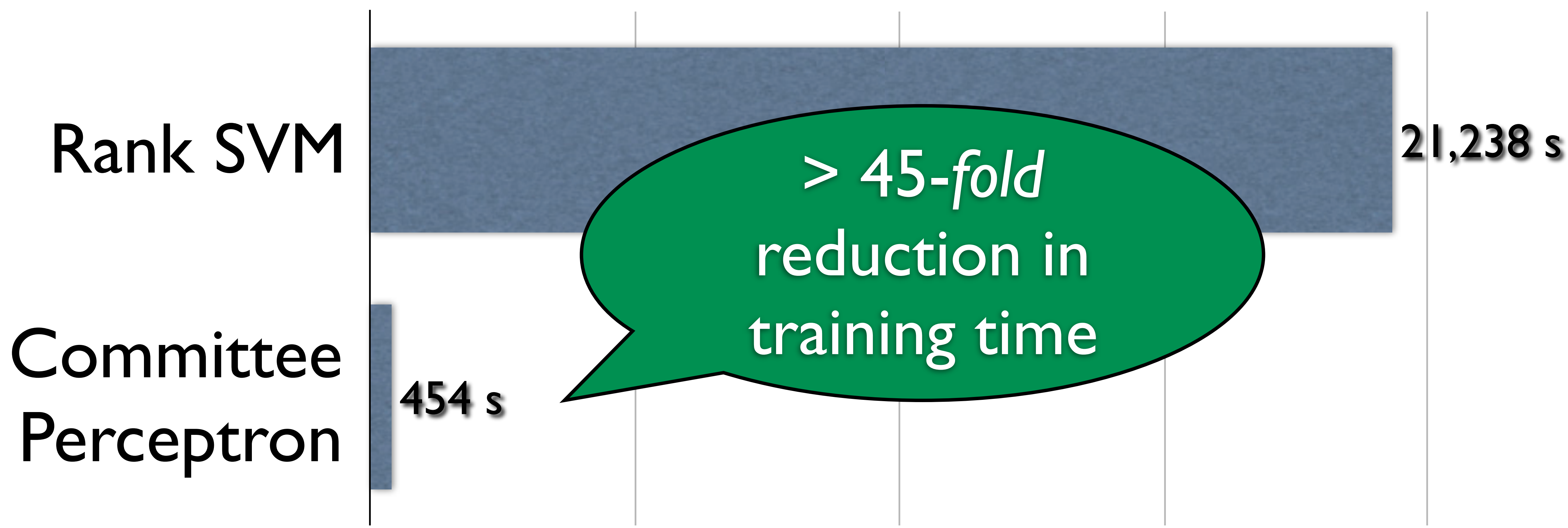


Training Time



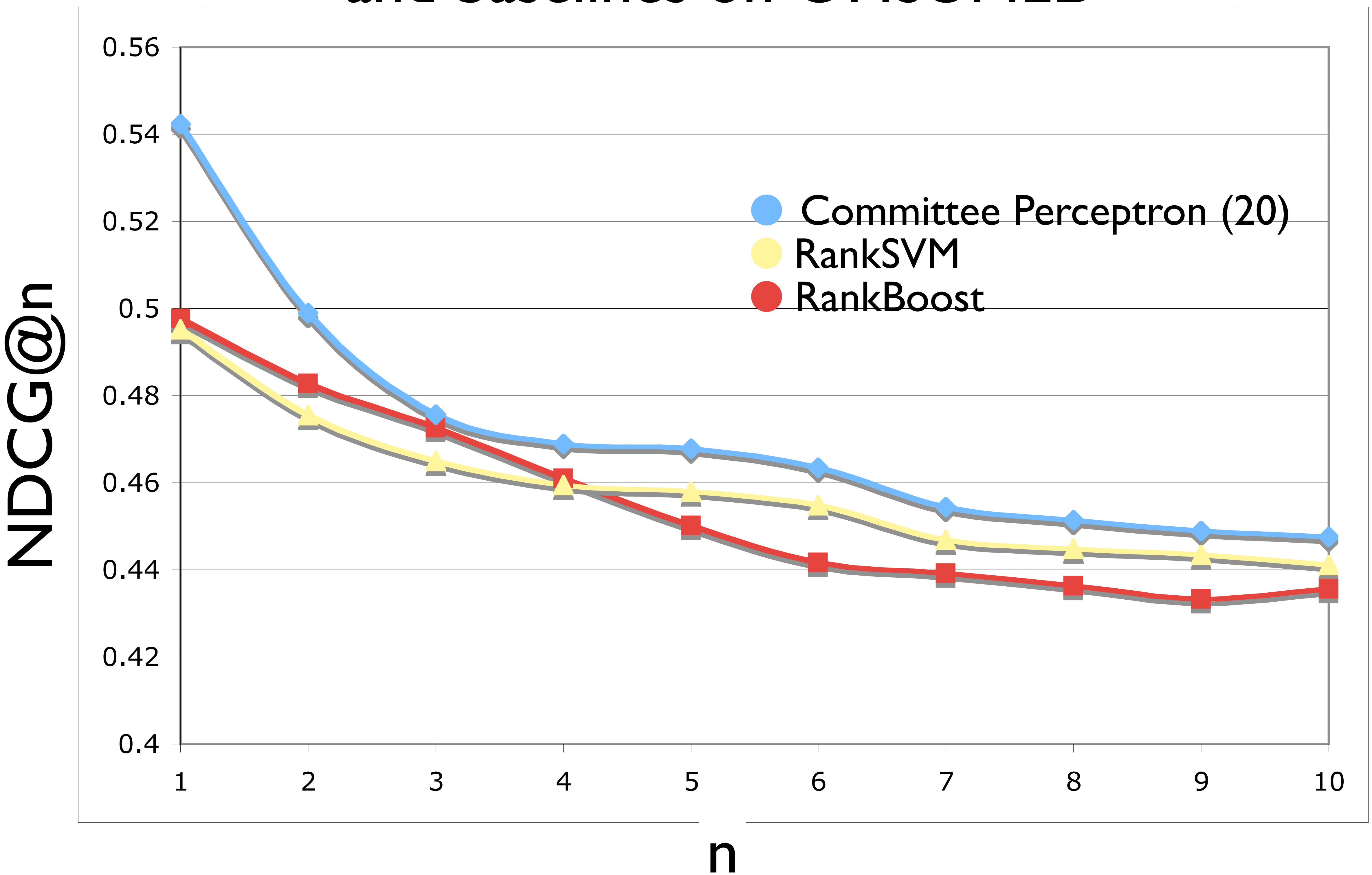
Committee Perceptron: K=20, iterations=50

Training Time

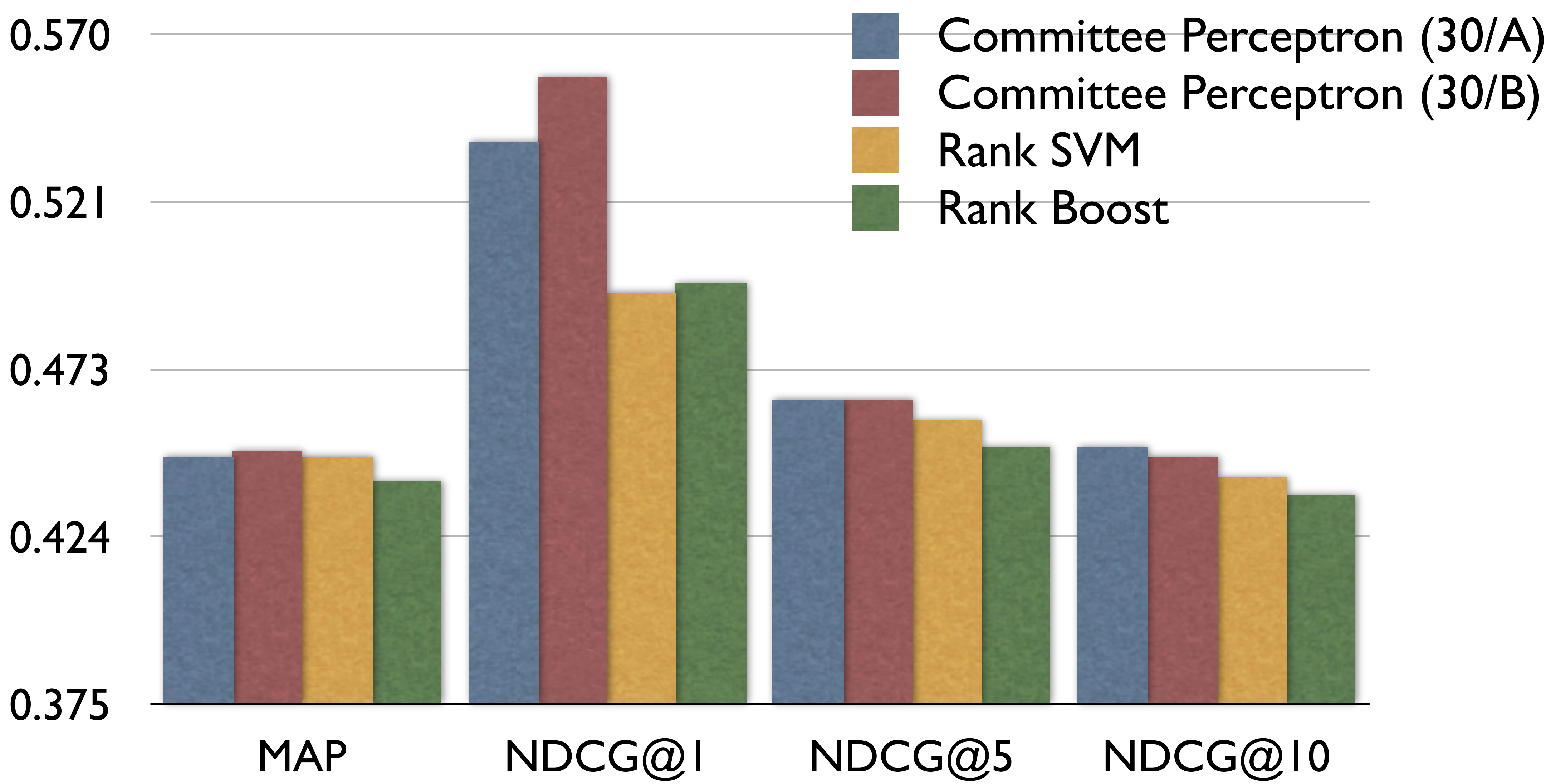


Committee Perceptron: K=20, iterations=50

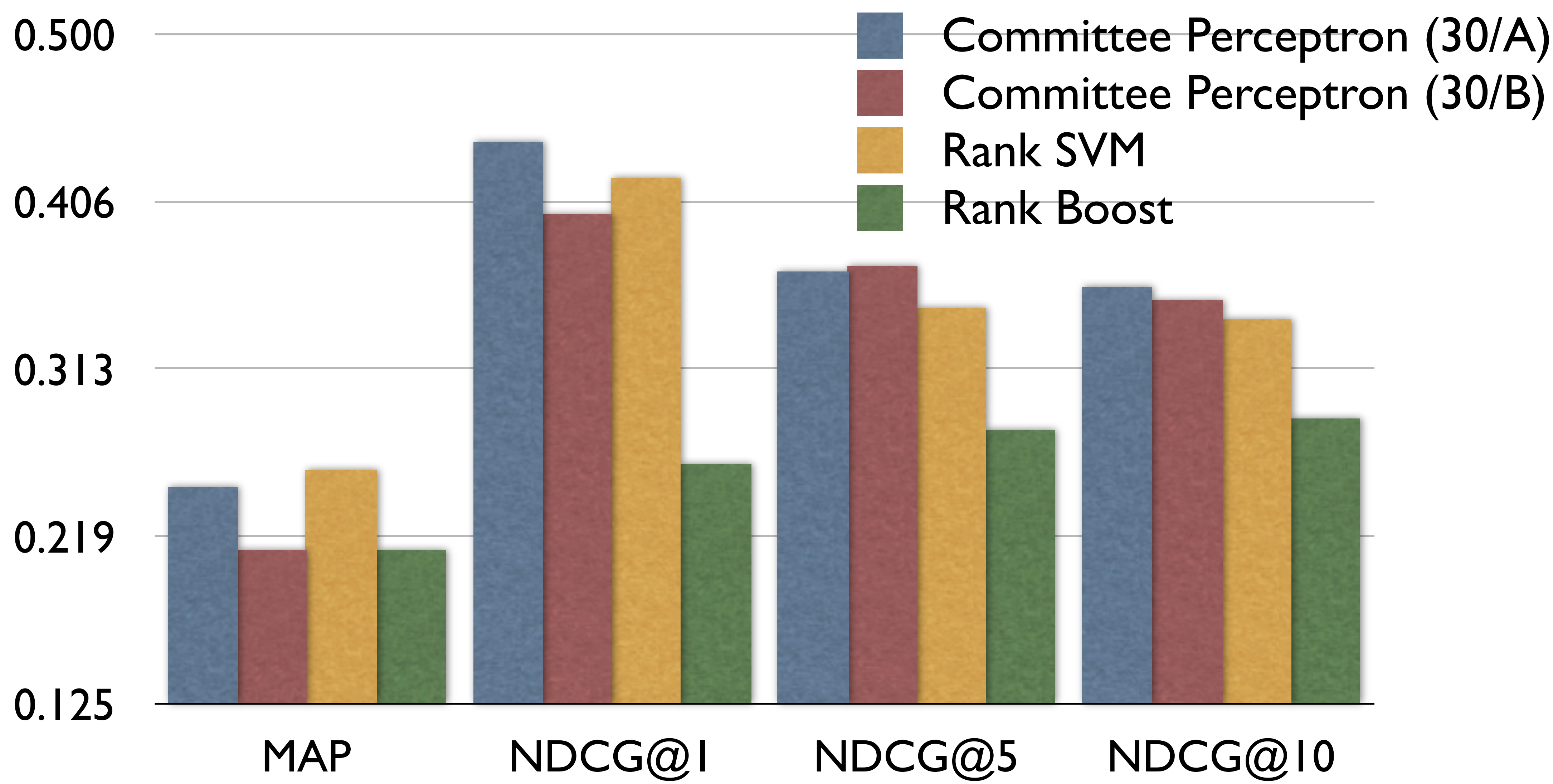
NDCG@n for Committee Perceptron and baselines on OHSUMED



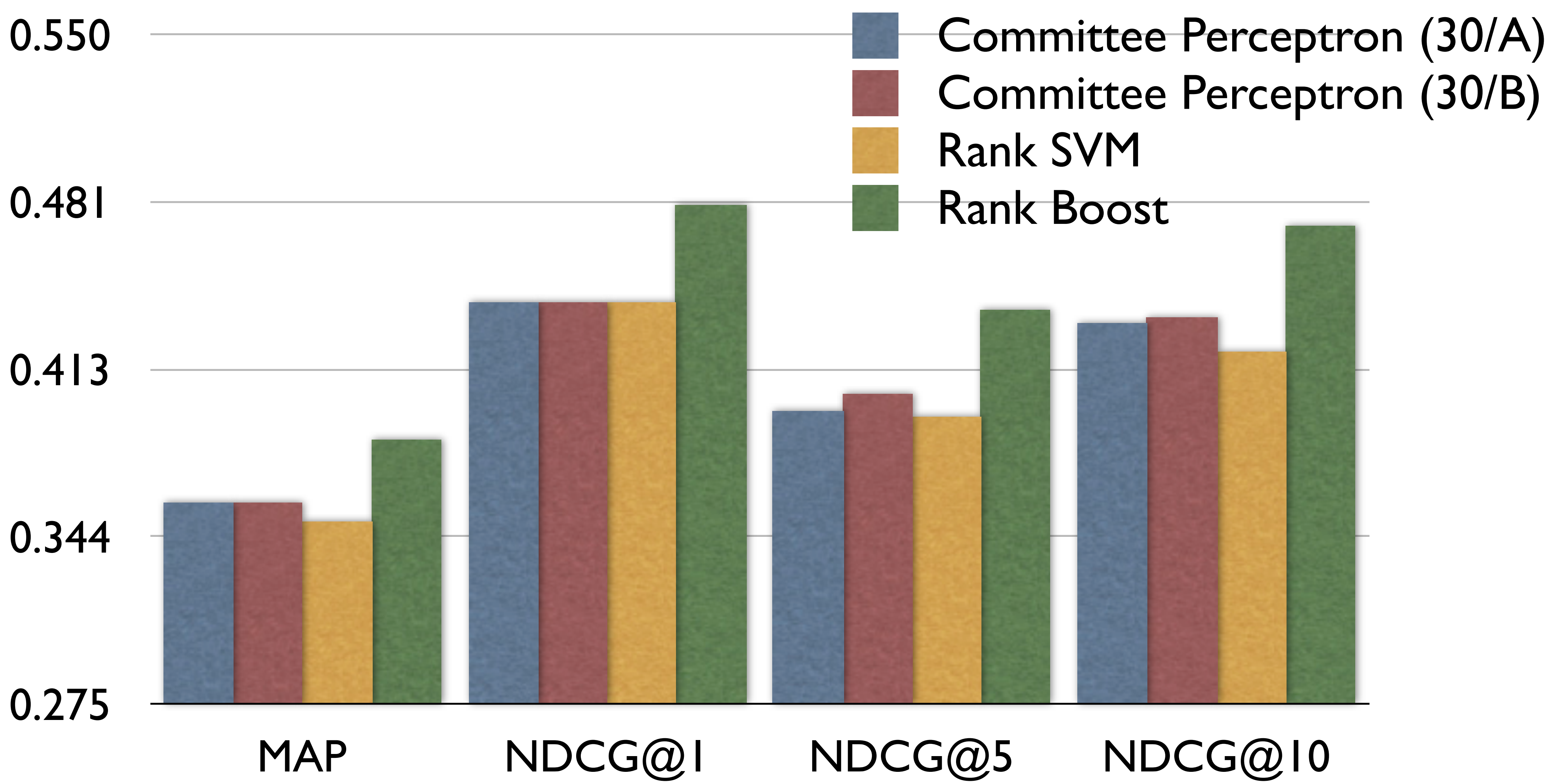
CP Performance: OHSUMED



CP Performance: TD2003



CP Performance: TD2004



Conclusions

- Committee Perceptron is a *fast* algorithm
  - faster & more stable than other perceptron variants
  - O(1) space and test-time complexity
  - >45-*fold* training time reduction vs. RankSVM
- Comparable or better performance to RankSVM and RankBoost

Thank You!

CP Pseudo-Code

Input: Iterations T, Committee Size K, Training document pairs $S = \{(q, d_i, d_j) | d_i \succ_q d_j\}$.
Output: Weight vectors and success counters $W = \{(\mathbf{w}^k, c_k)\}$, $|W| = K$

1. Initialize $l = 0$, success counter $c_l = 0$, initial parameters $\mathbf{w}^0$, committee $W = \emptyset$, query balancing factor $\eta_q = 1/|S_q|$.

2. For $t = 0, \ldots, T$:

    For each training sample $(\mathbf{d}_i, \mathbf{d}_j)$:

    If $s(\mathbf{d}_j, \mathbf{w}^l) \geq s(\mathbf{d}_i, \mathbf{w}^l)$ then — A mis-ranking

    $(\mathbf{w}^{\min}, c^{\min}) \in W$ s.t. $c^{\min} = \min_k c_k \in K$

    If $c_l > c^{\min}$ then — Maintaining a fixed-size priority queue

    add $(\mathbf{w}^l, c_l)$ to W

    while $|W| > K$, remove $(\mathbf{w}^{\min}, c^{\min})$ from W

    update: $\mathbf{w}^{i+l} = \mathbf{w}^l + \eta_q(\mathbf{d}_i - \mathbf{d}_j)$, $l = l + 1$ — Hypothesis update

    Else update: $c_l = c_l + 1$ — Success counter update

3. Output: W

Pairwise Preferences & Performance Measures

$$\Phi_{Z,m} = \frac{1}{Z} \sum_{i=1}^{m} \frac{i}{r_i}$$

$\Phi_{Z,m}$	Z	m
Average Precision	R	R
Precision at K	K	m_k s.t. $r_{m_k} \leq K < r_{m_{k+1}}$
Reciprocal Rank	1	1
R-Precision	R	m_r s.t. $r_{m_r} \leq R < r_{m_{r+1}}$

R = number of relevant documents

r_i = rank of i^{th} relevant document