CSE419 – Artificial Intelligence and Machine Learning 2020

PhD Furkan Gözükara, Toros University

https://github.com/FurkanGozukara/CSE419-ArtificiaHntelligence-and-Machine-Leaming-2020

Lecture 10 Ranking

Based on Asst. Prof. Dr. David Kauchak (Pomona College) Lecture Slides

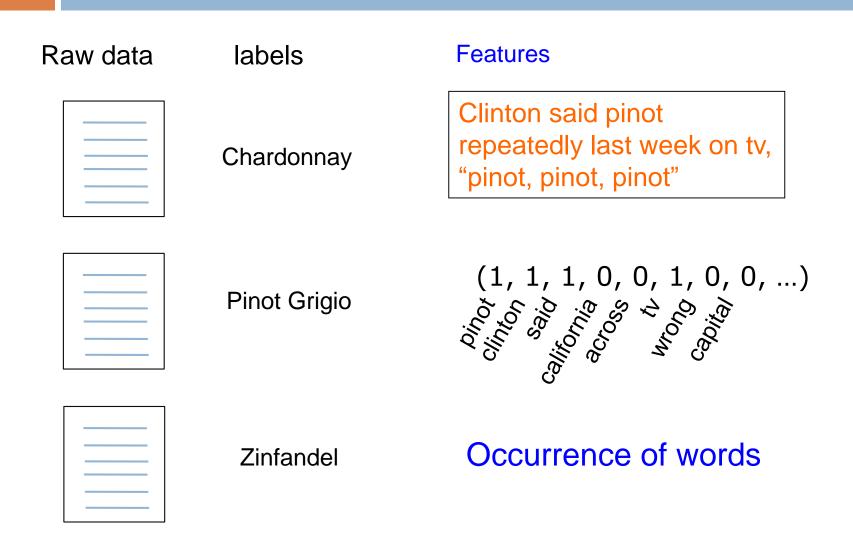
An aside: text classification

Raw data labels Chardonnay Pinot Grigio Zinfandel

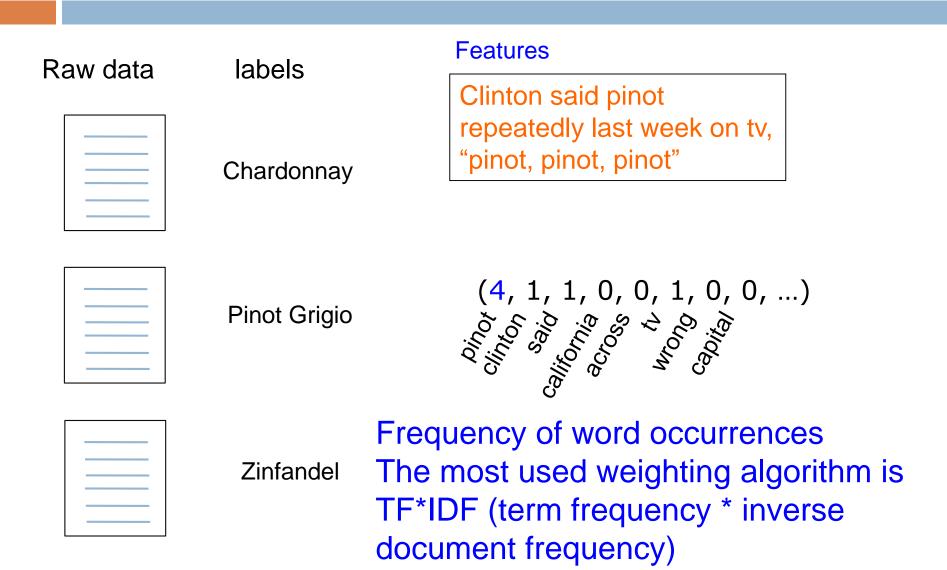
Text: raw data

Features? Raw data labels Chardonnay Pinot Grigio Zinfandel

Feature examples

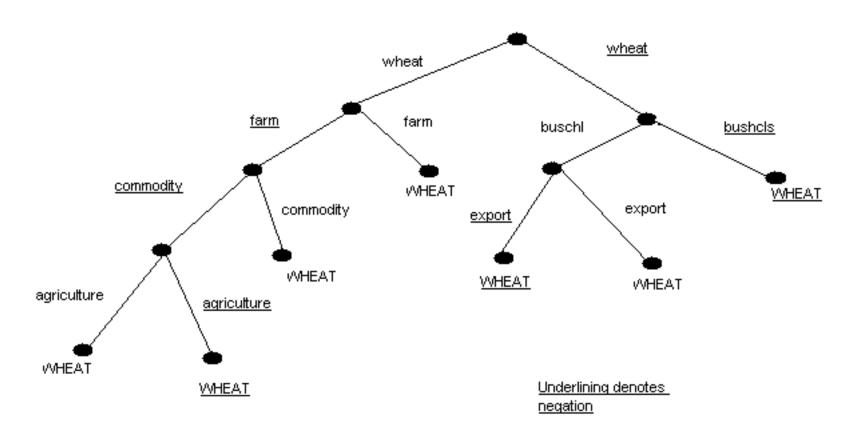


Feature examples



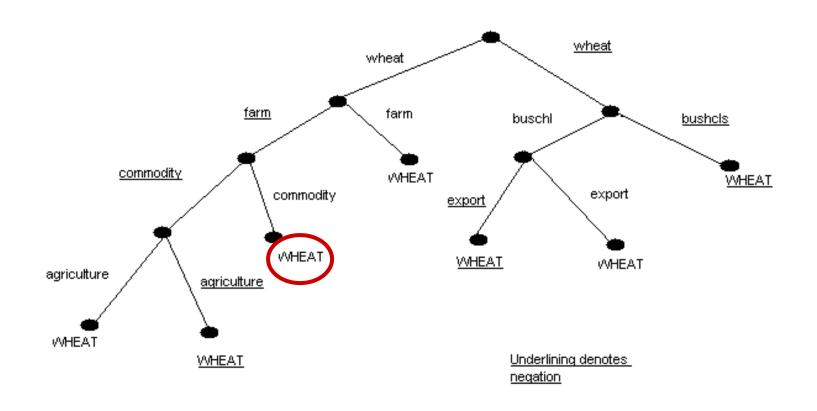
Decision trees for text

Each internal node represents whether or not the text has a particular word



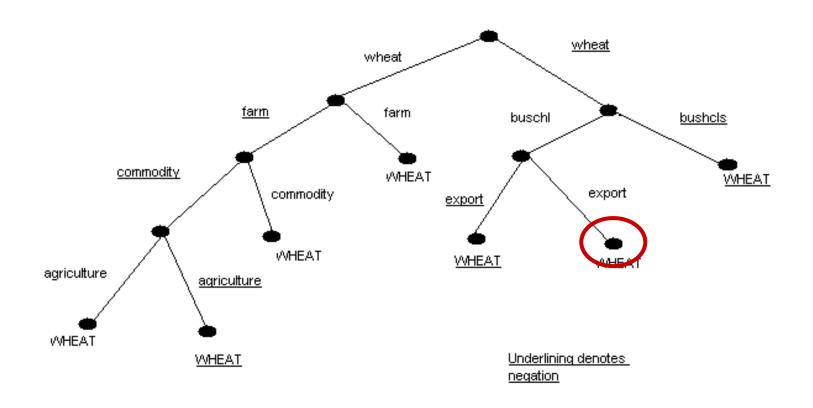
Decision trees for text

wheat is a commodity that can be found in states across the nation

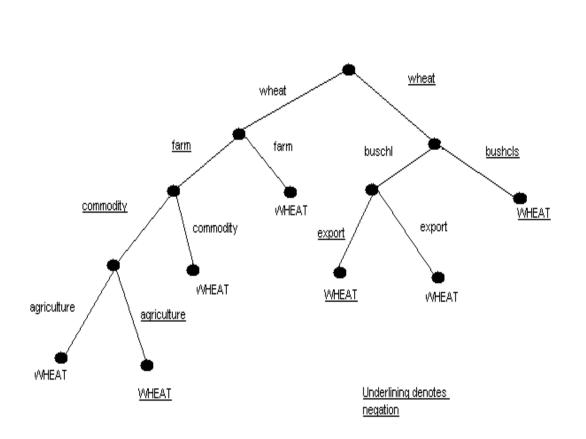


Decision trees for text

The US views technology as a commodity that it can export by the buschl.



Printing out decision trees



```
(wheat
 (buschl
    predict: not wheat
    (export
     predict: not wheat
     predict: wheat))
 (farm
    (commodity
      (agriculture
       predict: not
wheat
       predict: wheat)
     predict: wheat)
    predict: wheat))
```

Ranking problems

Suggest a simpler word for the word below:

vital

Suggest a simpler word for the word below:

vital

word	frequency
important	13
necessary	12
essential	11
needed	8
critical	3
crucial	2
mandatory	1
required	1
vital	1

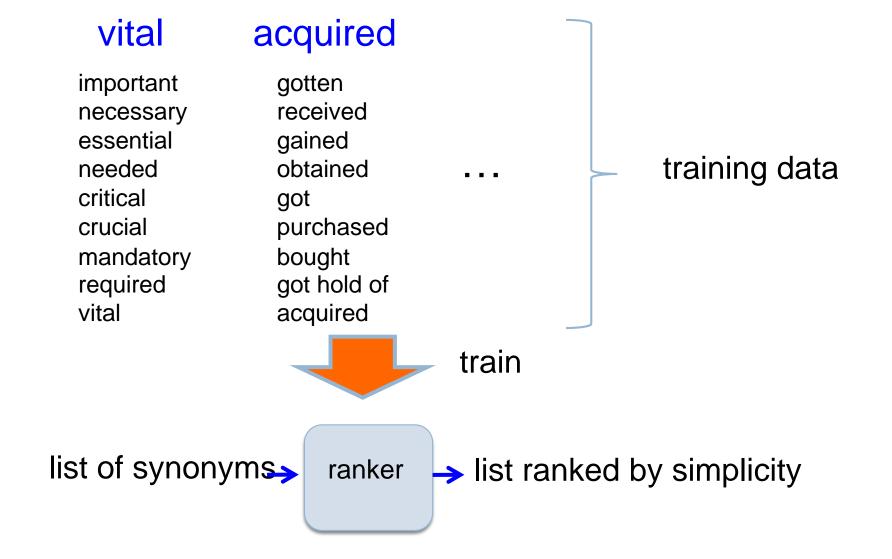
Suggest a simpler word for the word below:

acquired

Suggest a simpler word for the word below:

acquired

word	frequency
gotten	12
received	9
gained	8
obtained	5
got	3
purchased	2
bought	2
got hold of	1
acquired	1



Ranking problems in general

ranking1

ranking2

ranking3

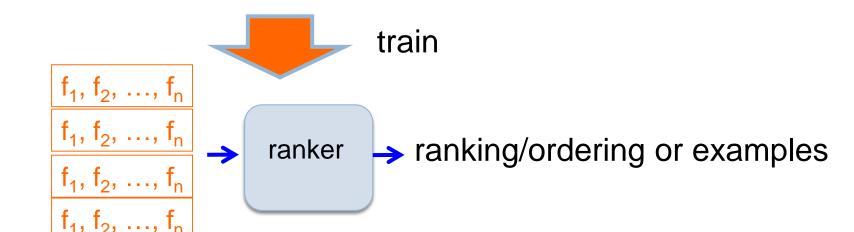
$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$

training data: a set of rankings where

each ranking consists of a set of ranked examples



Ranking problems in general

ranking1

$f_1, f_2, ..., f_n$ $f_1, f_2, ..., f_n$ $f_1, f_2, ..., f_n$ $f_1, f_2, ..., f_n$

ranking2

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$

ranking3

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$
 $f_1, f_2, ..., f_n$

training data:

a set of rankings where each ranking consists of a set of ranked examples

Real-world ranking problems?

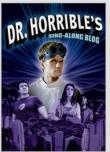
Netflix My List

Top 10 for Michael











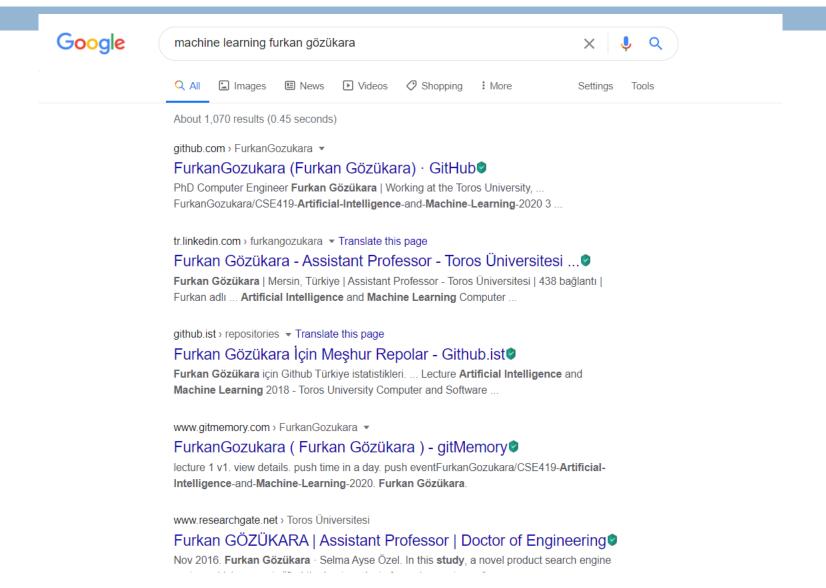


+ My List



Popular on Netflix

Search



Ranking Applications

reranking N-best output lists

- machine translation
- computational biology
- parsing

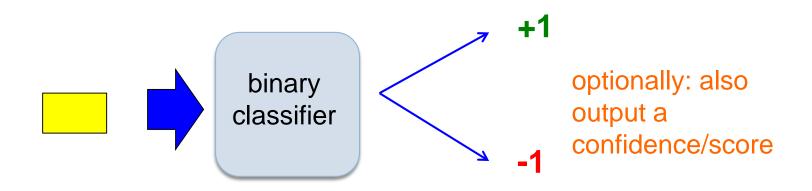
- . . .

flight search

. . .

Black box approach to ranking

Abstraction: we have a generic binary classifier, how can we use it to solve our new problem



Can we solve our ranking problem with this?

Train a classifier to decide if the first input is better than second:

- Consider all possible pairings of the examples in a ranking
- Label as positive if the first example is higher ranked, negative otherwise

ranking1

```
f_1, f_2, ..., f_n

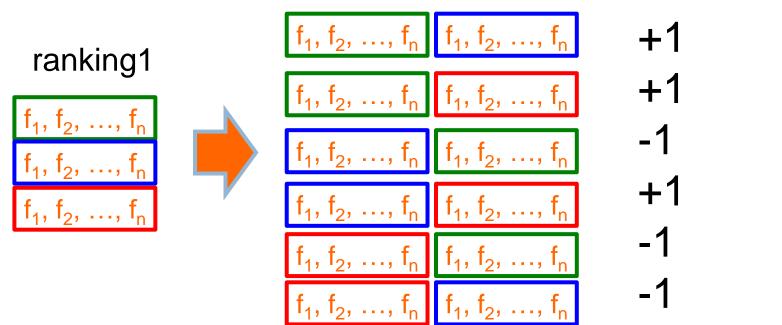
f_1, f_2, ..., f_n

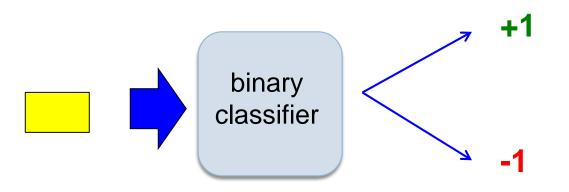
f_1, f_2, ..., f_n
```

Train a classifier to decide if the first input is better than second:

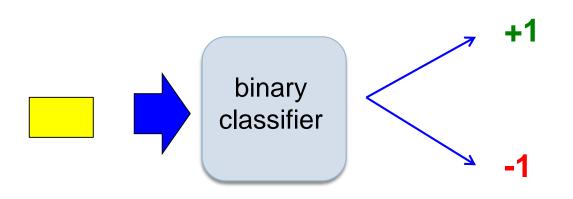
- Consider all possible pairings of the examples in a ranking
- Label as positive if the first example is higher ranked,

negative otherwise new examples binary label

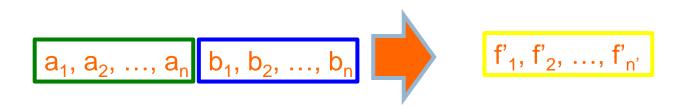




Our binary classifier only takes one example as input



Our binary classifier only takes one example as input



How can we do this?
We want features that compare the two examples.

Combined feature vector

Many approaches! Will depend on domain and classifier

Two common approaches:

- 1. difference: $f'_i = a_i b_i$
- greater than/less than:

$$f'_{i} = \begin{cases} 1 & \text{if } a_{i} > b_{i} \\ 0 & \text{otherwise} \end{cases}$$

Training

new examples

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

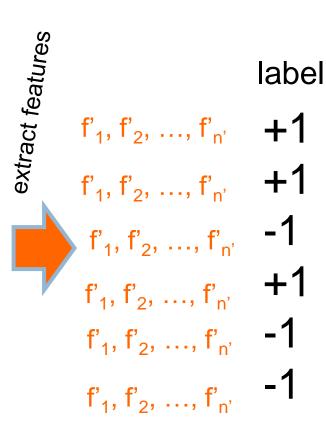
$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

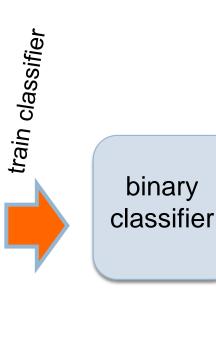
$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

$$f_1, f_2, ..., f_n | f_1, f_2, ..., f_n$$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

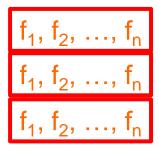
$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$





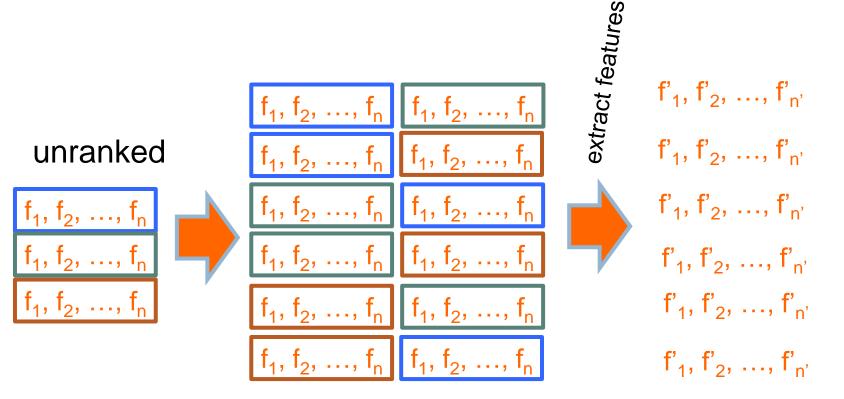
binary classifier

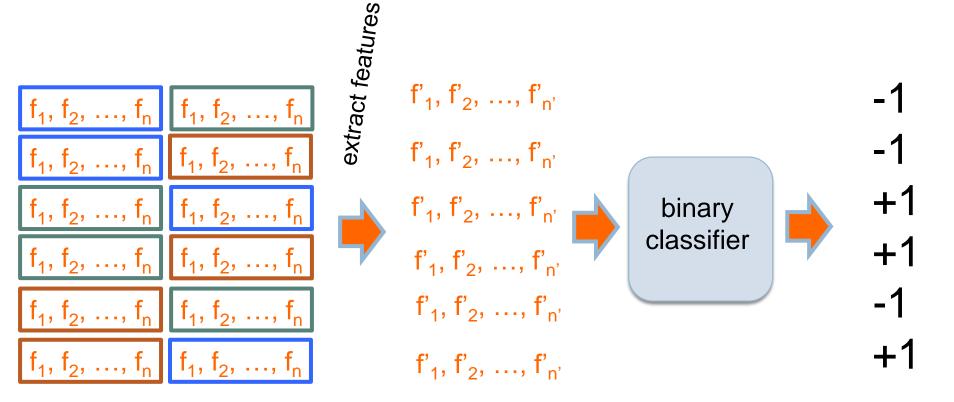
unranked

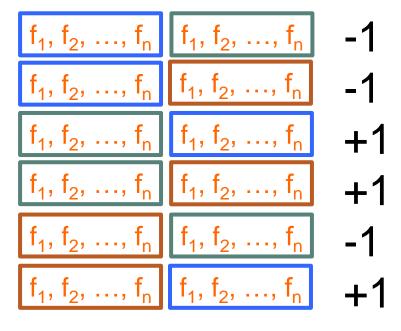




ranking?







What is the ranking? Algorithm?

for each binary example
$$e_{jk}$$
:
label[j] += $f_{jk}(e_{jk})$
label[k] -= $f_{jk}(e_{jk})$

rank according to label scores

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$ -1

$$f_n | f_1, f_2, ..., f_n |$$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

$$f_1, f_2, ..., f_n | f_1, f_2, ..., f_n$$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

$$f_1, f_2, ..., f_n$$
 $f_1, f_2, ..., f_n$

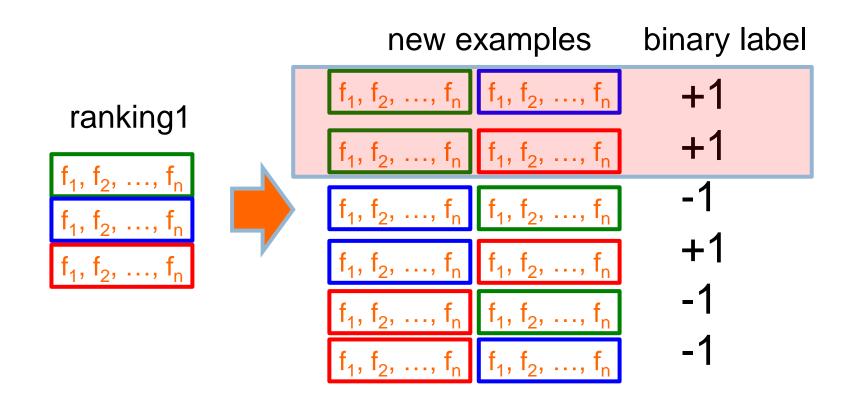




$$f_1, f_2, ..., f_n$$

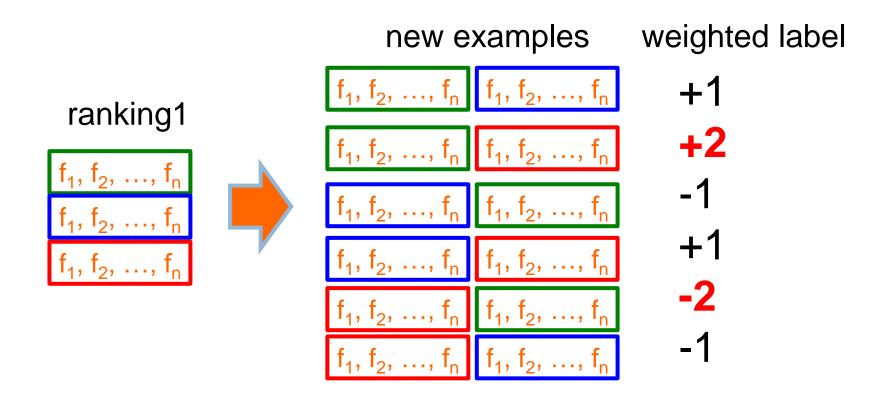


An improvement?



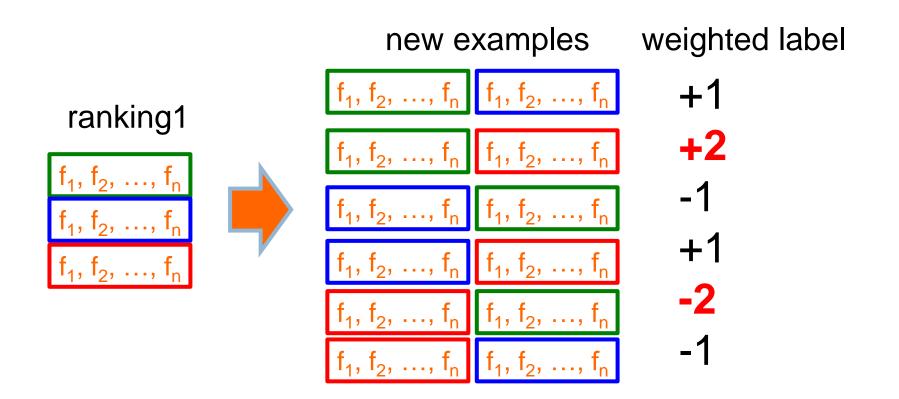
Are these two examples the same?

Weighted binary classification



Weight based on *distance* in ranking

Weighted binary classification



In general can weight with any consistent distance metric Can we solve this problem?

If the classifier outputs a confidence, then we've learned a *distance* measure between examples

During testing we want to rank the examples based on the learned distance measure

Ideas?

If the classifier outputs a confidence, then we've learned a *distance* measure between examples

During testing we want to rank the examples based on the learned distance measure

Sort the examples and use the output of the binary classifier as the similarity between examples!

Ranking evaluation

ranking prediction

f ₁ , f ₂ ,, f _n	1	1
$f_1, f_2,, f_n$	2	3
$f_1, f_2,, f_n$	3	2
f ₁ , f ₂ ,, f _n	4	5
f ₁ , f ₂ ,, f _n	5	4

Ideas?

Idea 1: accuracy

ranking prediction

Any problems with this?

Doesn't capture "near" correct

ranking prediction prediction

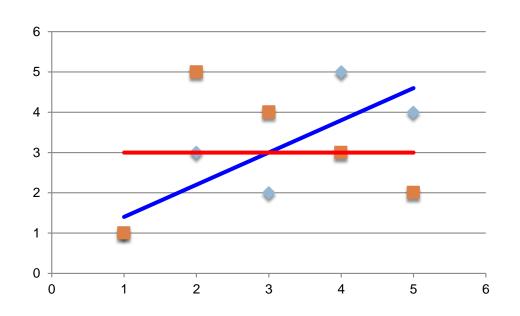
$f_1, f_2,, f_n$	1	1	1
$f_1, f_2,, f_n$	2	3	5
f ₁ , f ₂ ,, f _n	3	2	4
f ₁ , f ₂ ,, f _n	4	5	3
f ₁ , f ₂ ,, f _n	5	4	2

$$1/5 = 0.2$$

Idea 2: correlation

ranking prediction prediction

1
5
4
3
2



Look at the correlation between the ranking and the prediction

Very Useful Articles To Read

- □ The ABCs of Learning to Rank
- tf-idf Model for Page Ranking
- □ Tf-Idf and Cosine similarity
- Analyzing Documents with TF-IDF