CSE419 – Artificial Intelligence and Machine Learning 2021

PhD Furkan Gözükara, Toros University

https://github.com/FurkanGozukara/CSE419-Artificial-Intelligence-and-Machine-Learning-2021

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

Raw data   labels   Features?

Chardonnay

Pinot Grigio

Zinfandel

Feature examples

Raw data   labels   Features

Chardonnay

Clinton said pinot repeatedly last week on tv, "pinot, pinot, pinot"

Pinot Grigio

(1, 1, 1, 0, 0, 1, 0, 0, ...)

pinot clinton said california across tv wrong capital

Zinfandel

Occurrence of words

Feature examples

Raw data   labels

Chardonnay

Clinton said pinot repeatedly last week on tv, "pinot, pinot, pinot"

Pinot Grigio

$(4, 1, 1, 0, 0, 1, 0, 0, ...)$

pinot clinton said california across tv wrong capital

Zinfandel

Frequency of word occurrences
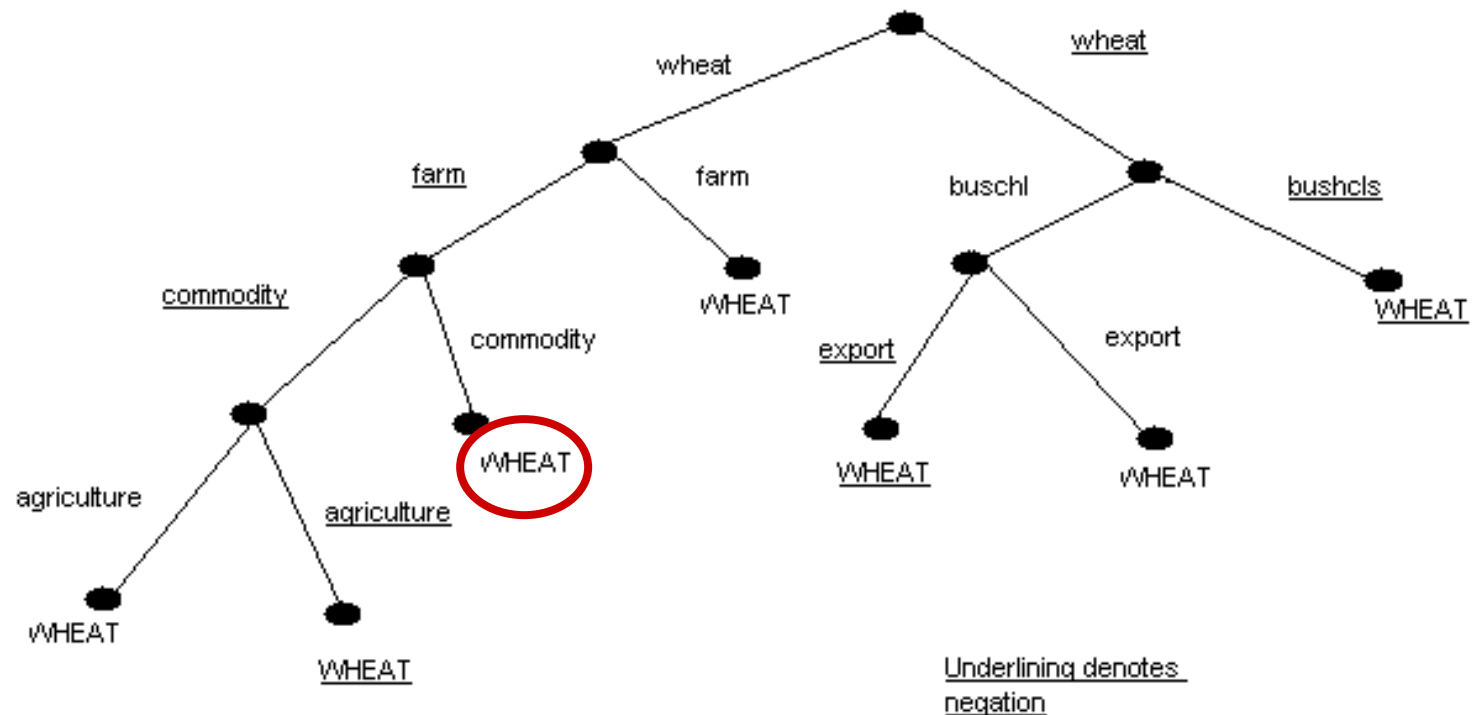The most used weighting algorithm is TF*IDF (term frequency * inverse document frequency)

Decision trees for text

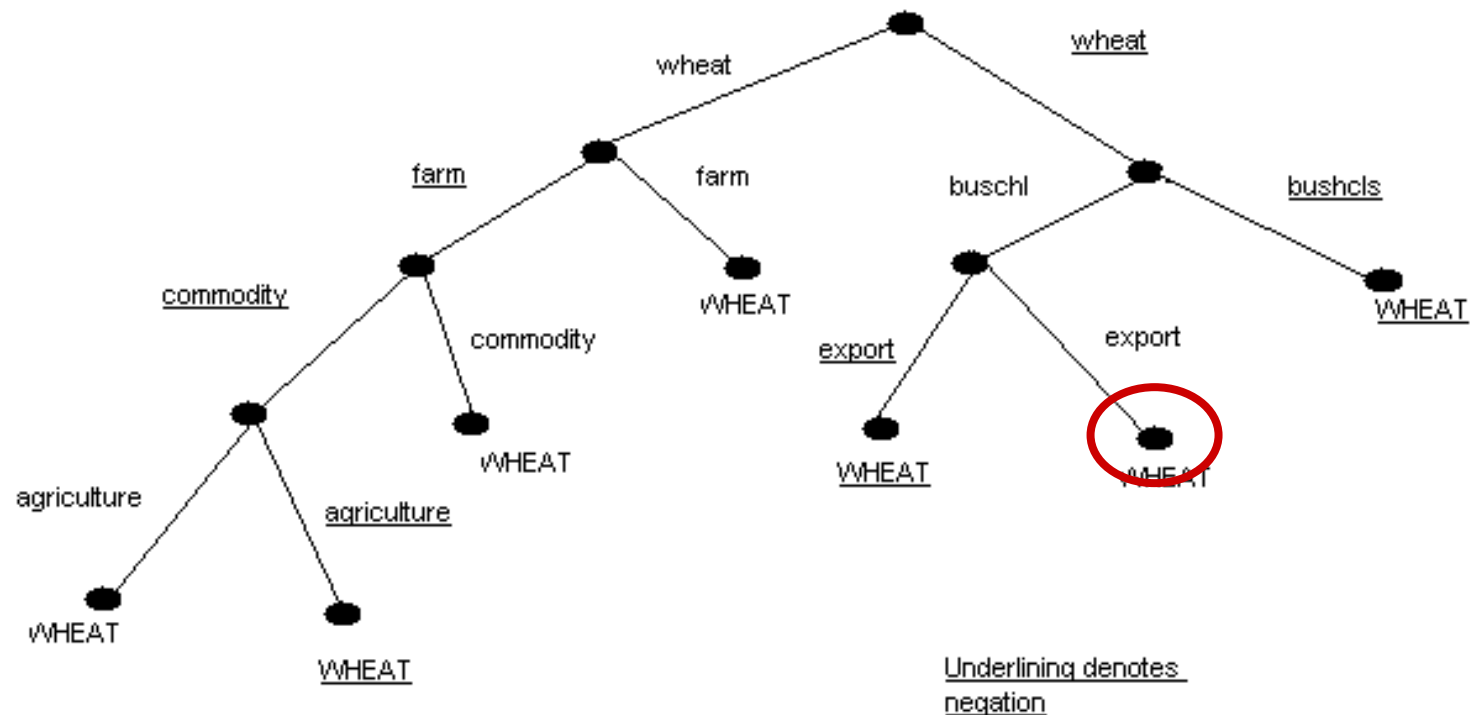Each internal node represents whether or not the text has a particular word
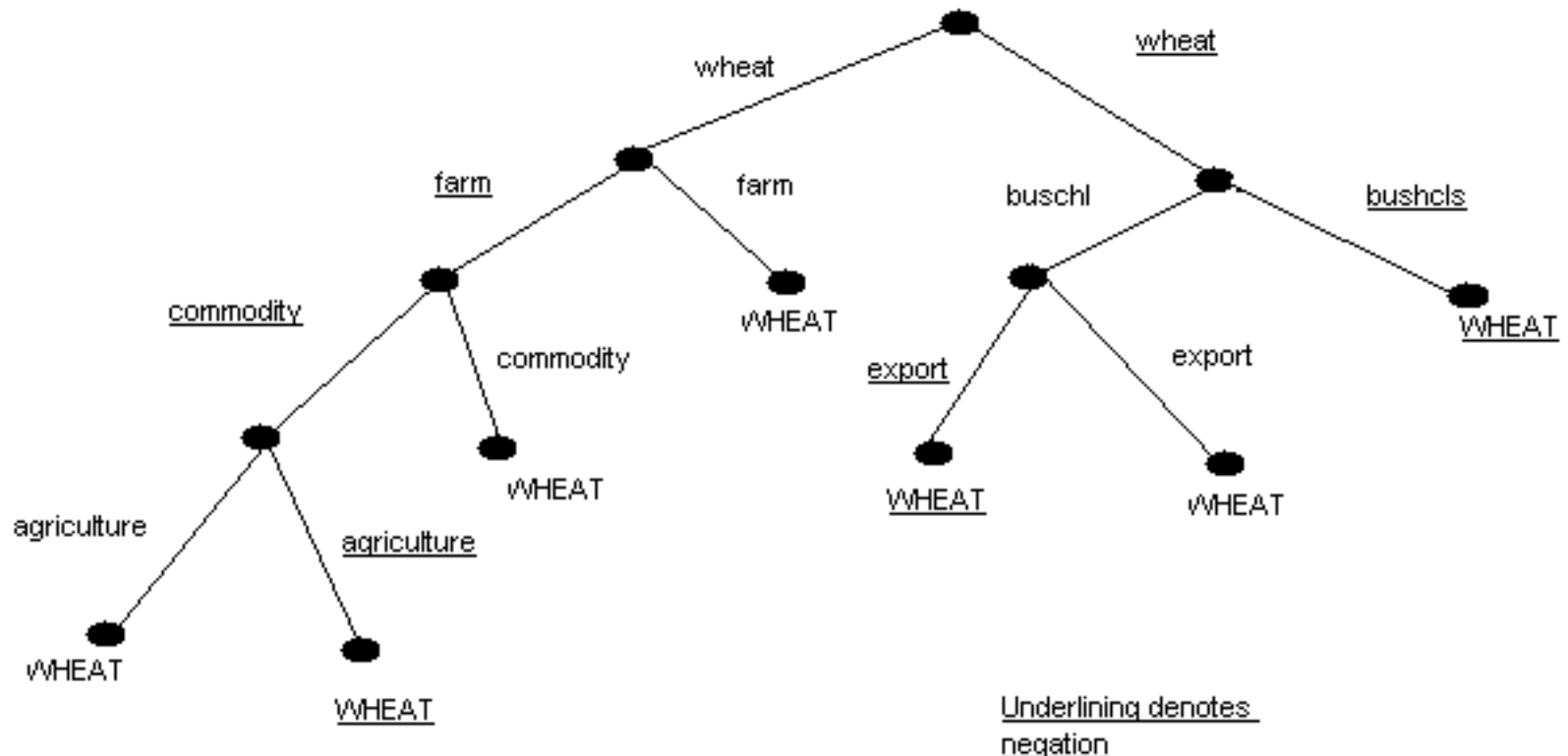
Decision trees for text

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

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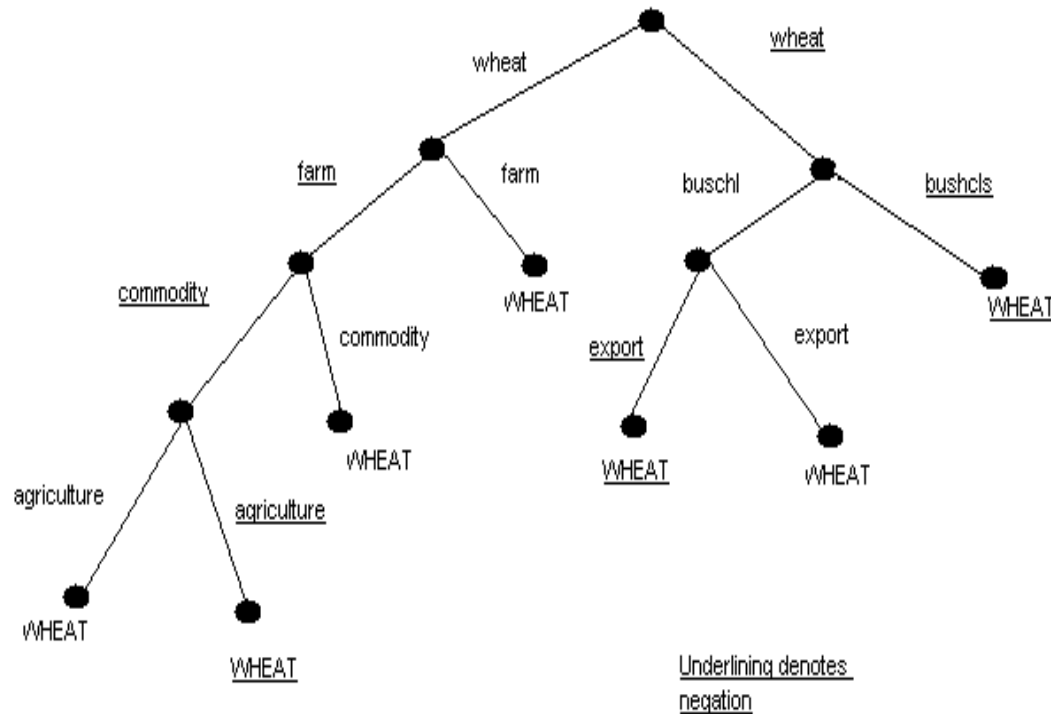
```
(wheat
  (buschl
    predict: not wheat
    (export
      predict: not wheat
      predict: wheat))
  (farm
    (commodity
      (agriculture
        predict: not wheat
        predict: wheat)
      predict: wheat)
    predict: wheat))
```

Ranking problems

Suggest a simpler word for the word below:

vital

Suggest a simpler word

Suggest a simpler word for the word below:

<p style="text-align:center;color:blue">vital</p>

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

Suggest a simpler word

Suggest a simpler word for the word below:

acquired

Suggest a simpler word

Suggest a simpler word for the word below:

<p style="text-align:center;color:blue;">acquired</p>

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

Suggest a simpler word

vital	acquired		
important	gotten		
necessary	received		
essential	gained		
needed	obtained	…	training data
critical	got		
crucial	purchased		
mandatory	bought		
required	got hold of		
vital	acquired		

train

list of synonyms → ranker → list ranked by simplicity
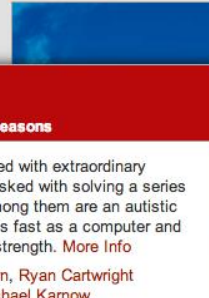
Ranking problems in general

ranking1   ranking2   ranking3

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

…

training data:
a set of rankings where each ranking consists of a set of ranked examples

train

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

ranker → ranking/ordering or examples

Ranking problems in general

ranking1   ranking2   ranking3

$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$

…

training data:
a set of rankings where each ranking consists of a set of ranked examples

Real-world ranking problems?

Netflix My List

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

binary classifier

+1

-1

optionally: also output a confidence/score
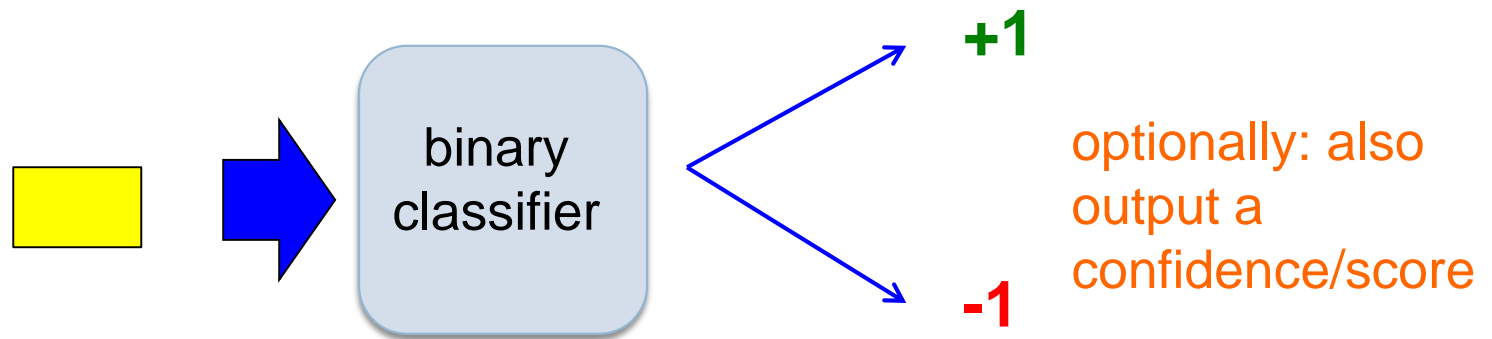
Can we solve our ranking problem with this?

Predict better vs. worse

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
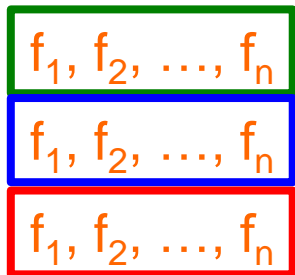
$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$

Predict better vs. worse

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

ranking1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ | $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

Predict better vs. worse

binary classifier

+1

-1

Our binary classifier only takes one example as input
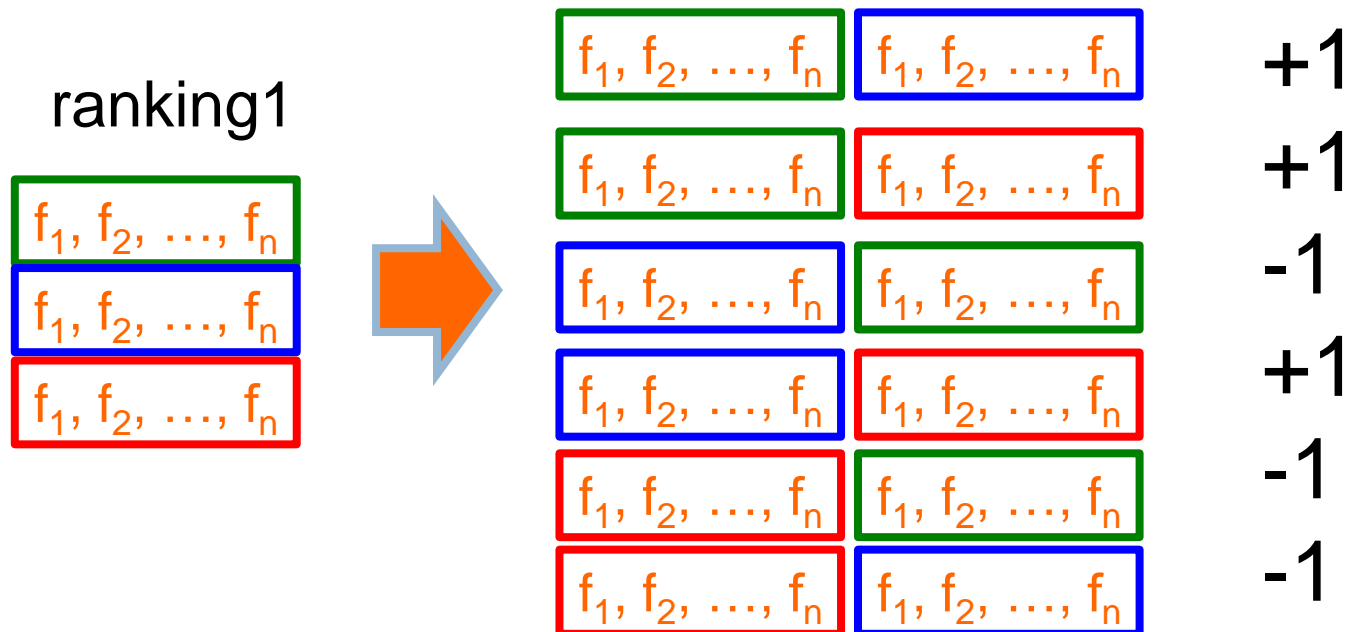
Predict better vs. worse

Our binary classifier only takes one example as input
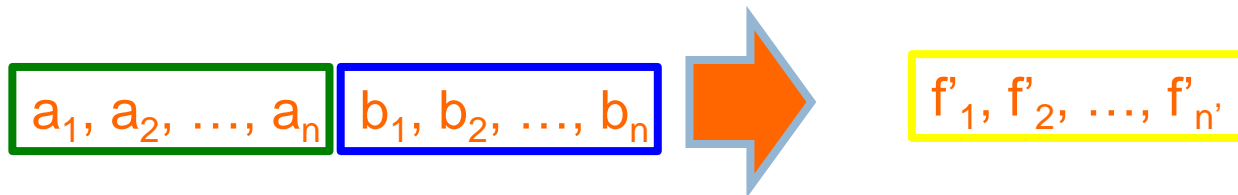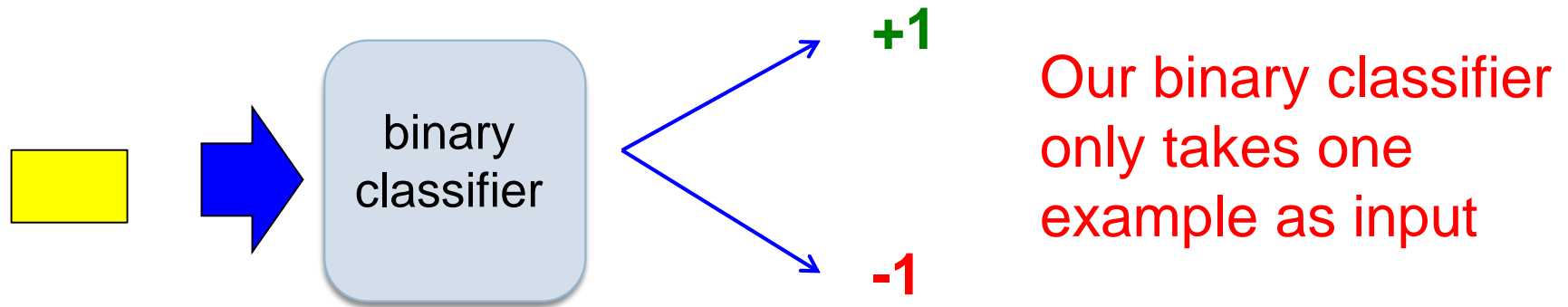
How can we do this?
We want features that compare the two examples.

Combined feature vector

Many approaches! Will depend on domain and classifier

Two common approaches:

1. difference: $f'_i = a_i - b_i$

2. greater than/less than:
$$f'_i = \begin{cases} 1 & if \ a_i > b_i \\ 0 & otherwise \end{cases}$$

Training

new examples

$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$

extract features

	label
$f'_1, f'_2, \ldots, f'_{n'}$	+1
$f'_1, f'_2, \ldots, f'_{n'}$	+1
$f'_1, f'_2, \ldots, f'_{n'}$	-1
$f'_1, f'_2, \ldots, f'_{n'}$	+1
$f'_1, f'_2, \ldots, f'_{n'}$	-1
$f'_1, f'_2, \ldots, f'_{n'}$	-1

train classifier
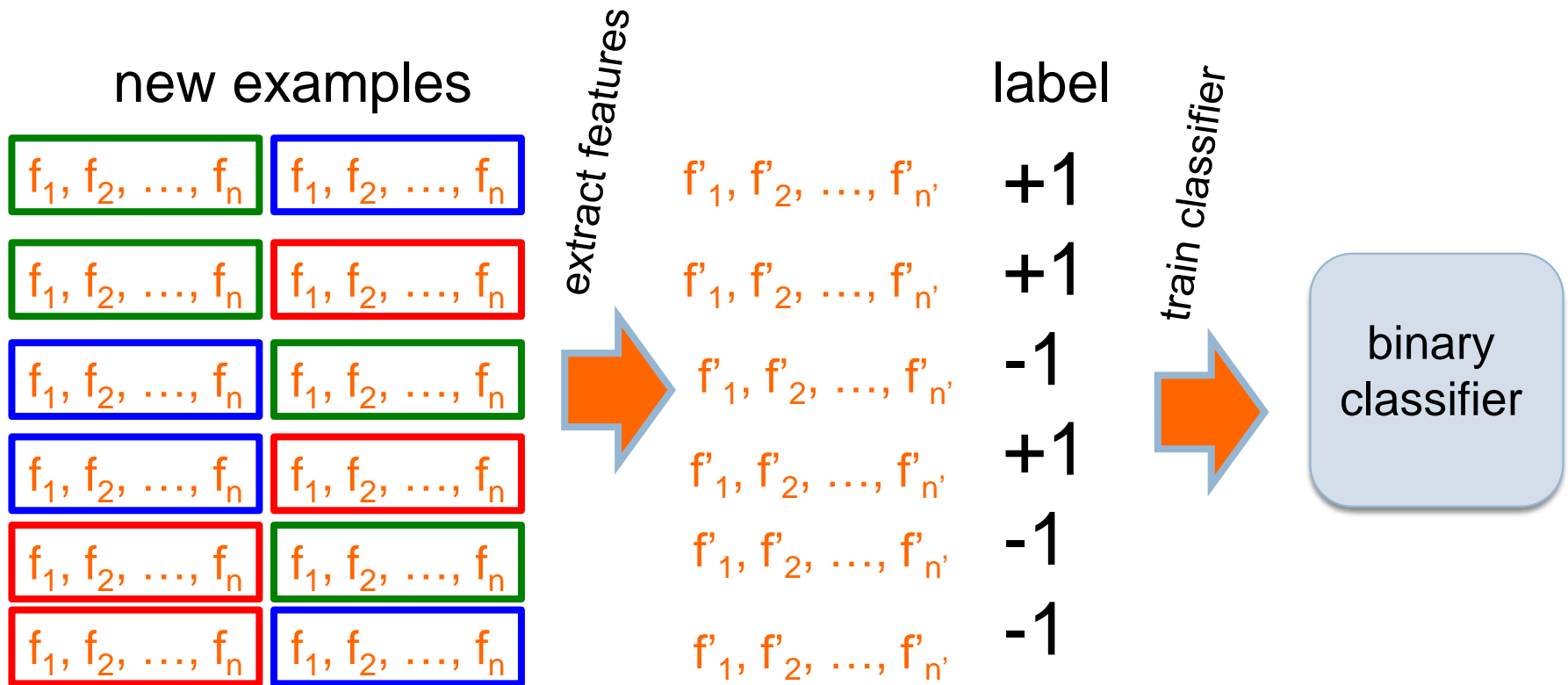
binary classifier

Testing

binary classifier

unranked

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

ranking?

Testing

unranked

$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

extract features

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

Testing

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$

extract features

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

$f'_1, f'_2, \ldots, f'_{n'}$

binary classifier

-1

-1

+1

+1

-1

+1

Testing

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

What is the ranking?
Algorithm?

Testing

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

rank according to label scores

An improvement?

new examples   binary label
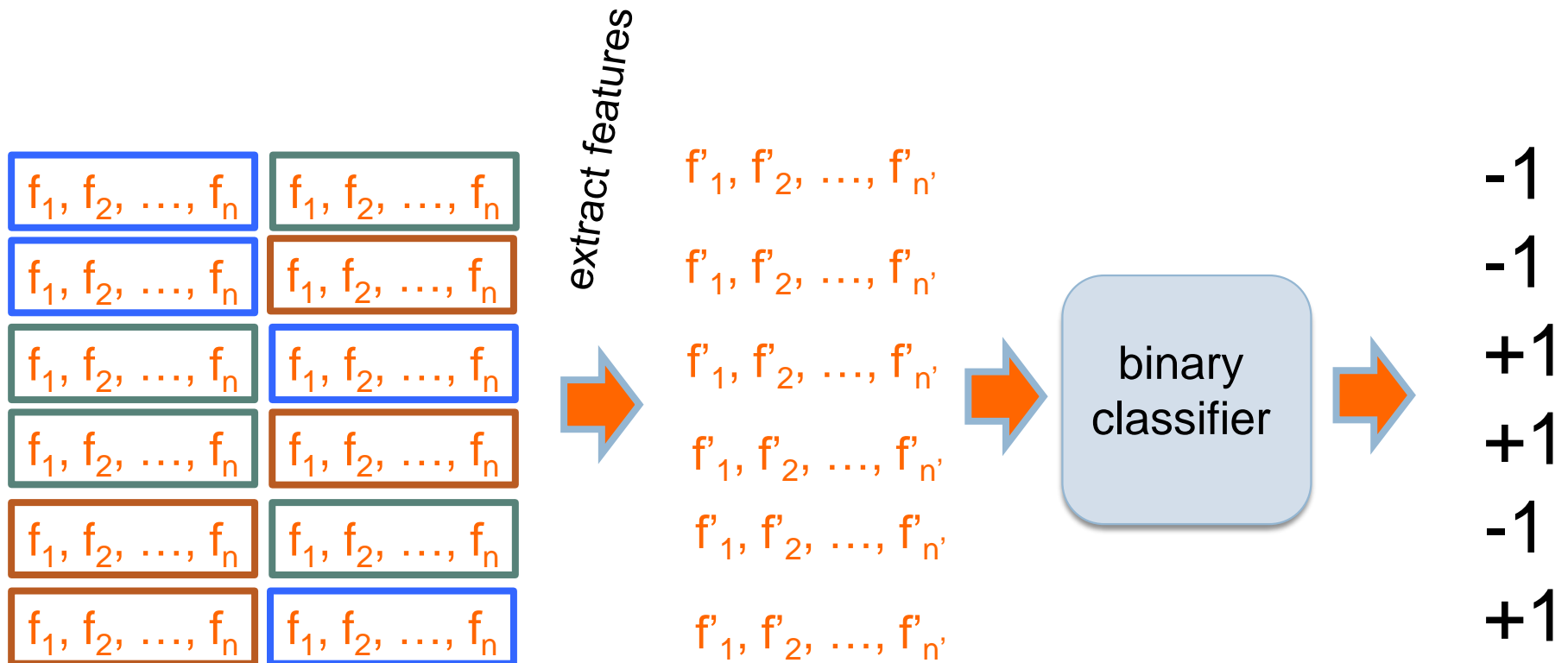
ranking1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ +1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1

$f_1, f_2, \ldots, f_n$ $f_1, f_2, \ldots, f_n$ -1
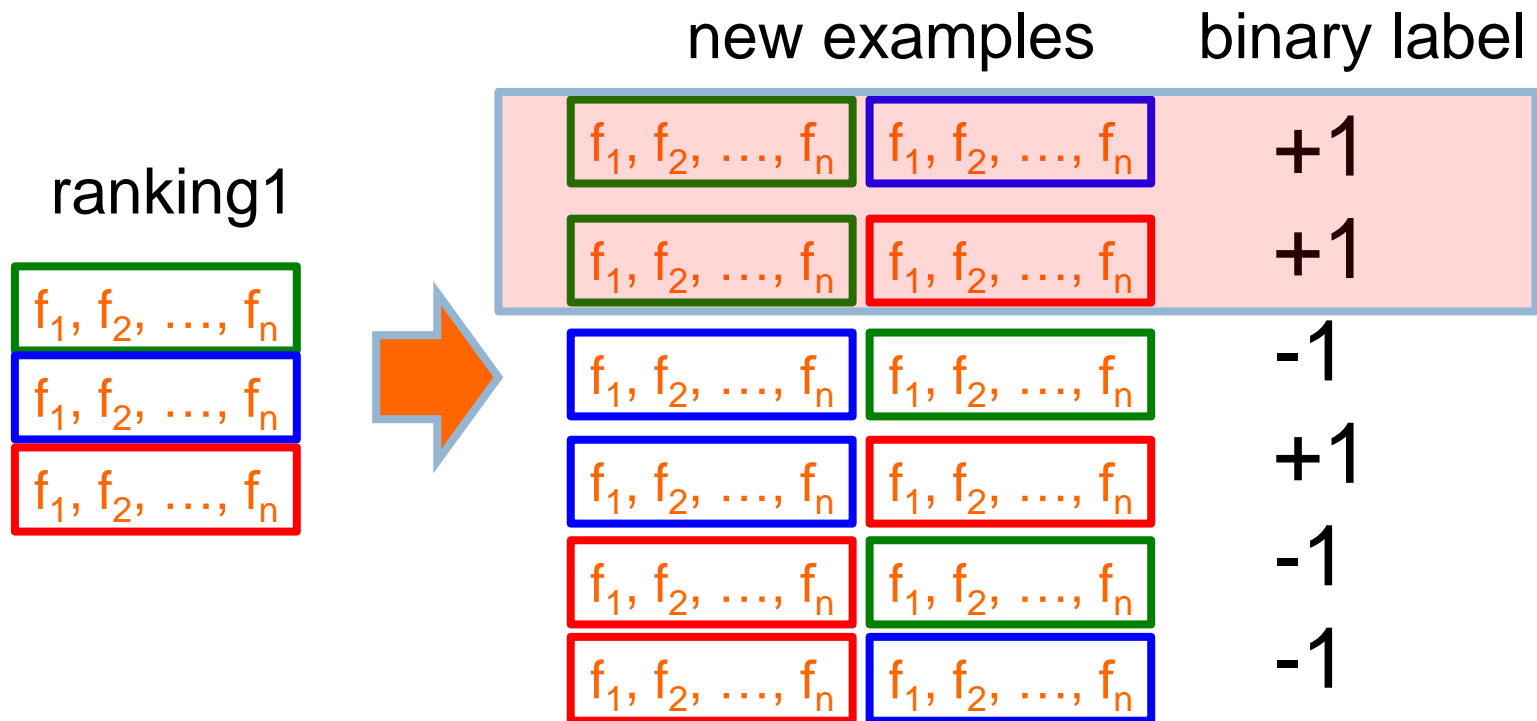
Are these two examples the same?

Weighted binary classification

new examples   weighted label

ranking1

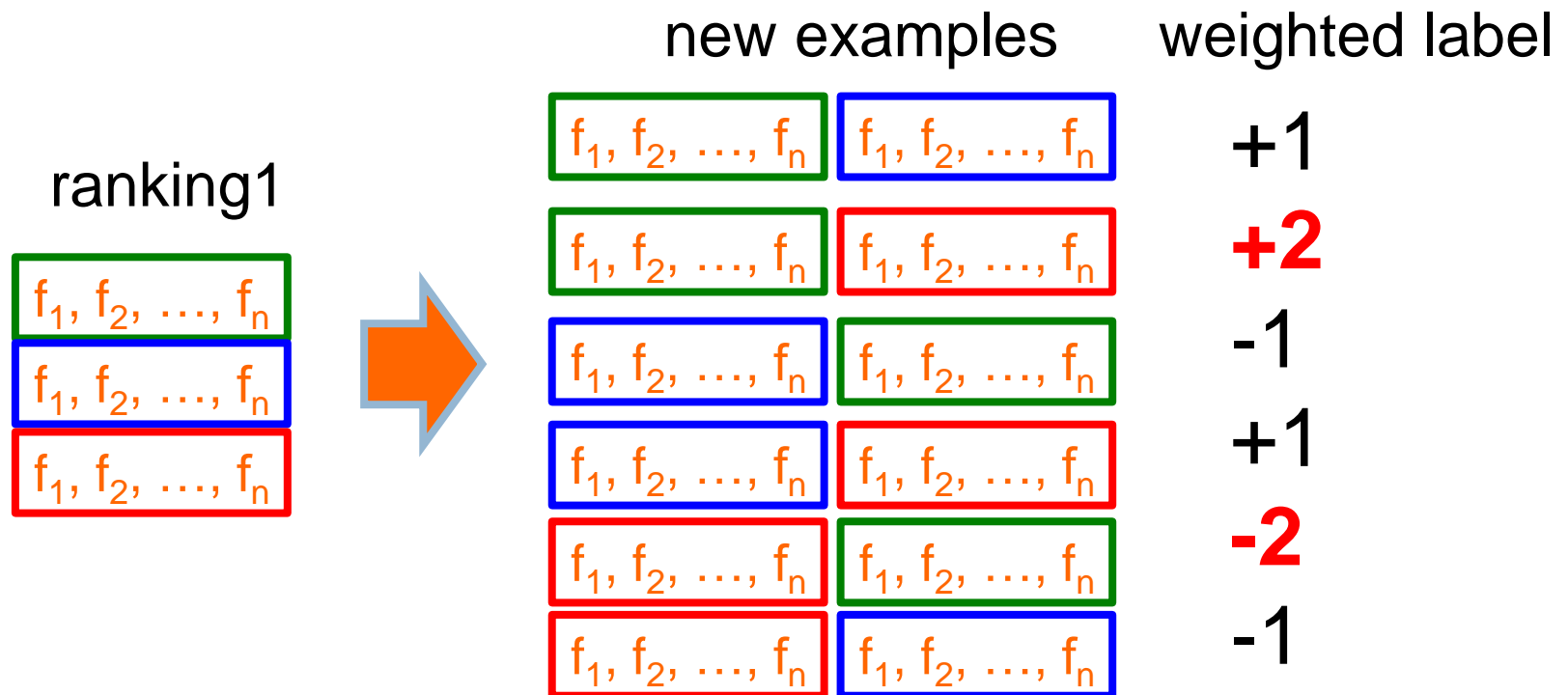|		
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	+1
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	**+2**
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	-1
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	+1
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	**-2**
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	-1

$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$
$f_1, f_2, \ldots, f_n$

Weight based on *distance* in ranking

Weighted binary classification

new examples   weighted label

ranking1

new examples		weighted label
$f_1, f_2, \ldots, f_n$ (green)	$f_1, f_2, \ldots, f_n$ (blue)	+1
$f_1, f_2, \ldots, f_n$ (green)	$f_1, f_2, \ldots, f_n$ (red)	**+2**
$f_1, f_2, \ldots, f_n$ (blue)	$f_1, f_2, \ldots, f_n$ (green)	-1
$f_1, f_2, \ldots, f_n$ (blue)	$f_1, f_2, \ldots, f_n$ (red)	+1
$f_1, f_2, \ldots, f_n$ (red)	$f_1, f_2, \ldots, f_n$ (green)	**-2**
$f_1, f_2, \ldots, f_n$ (red)	$f_1, f_2, \ldots, f_n$ (blue)	-1

ranking1:
$f_1, f_2, \ldots, f_n$ (green)
$f_1, f_2, \ldots, f_n$ (blue)
$f_1, f_2, \ldots, f_n$ (red)
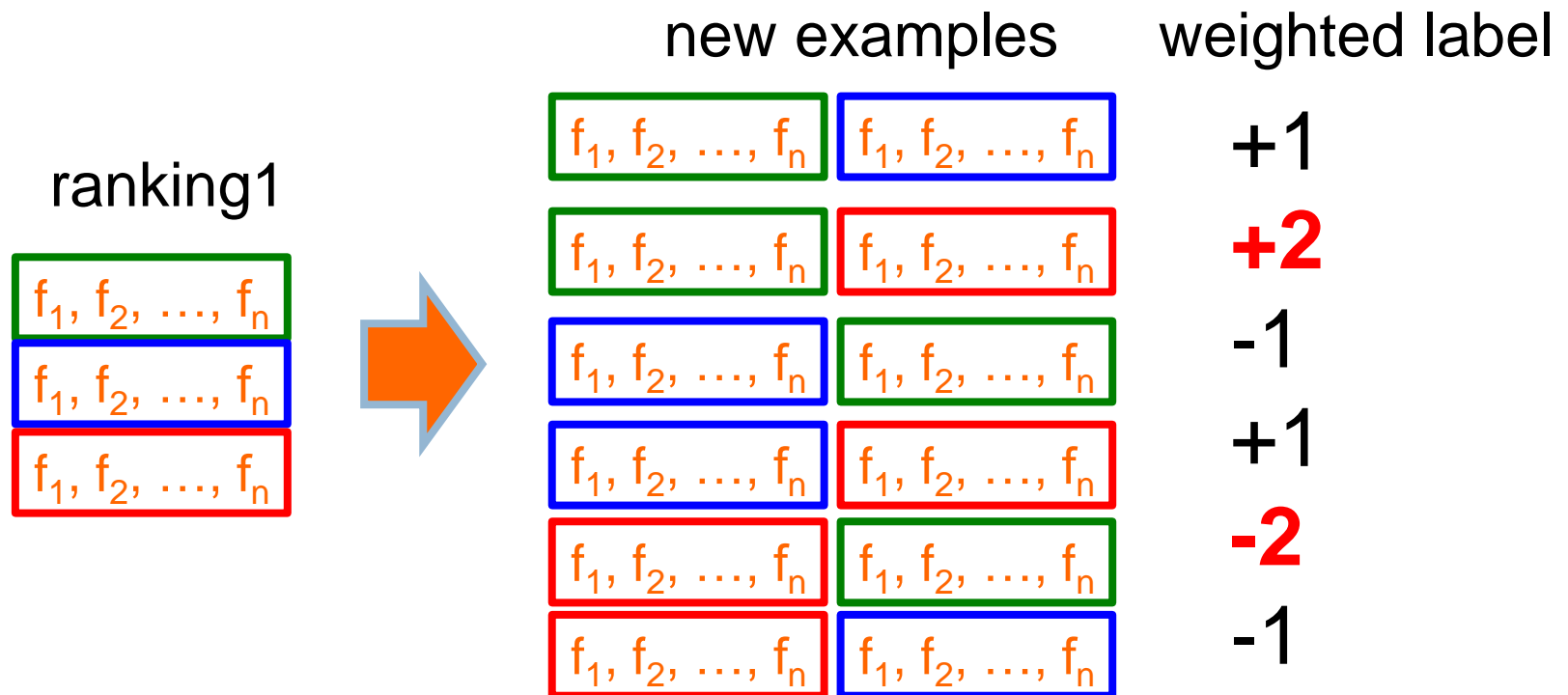
In general can weight with any consistent distance metric

Can we solve this problem?

Testing

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?

Testing

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
$f_1, f_2, \ldots, f_n$	1	1
$f_1, f_2, \ldots, f_n$	2	3
$f_1, f_2, \ldots, f_n$	3	2
$f_1, f_2, \ldots, f_n$	4	5
$f_1, f_2, \ldots, f_n$	5	4

Ideas?

Idea 1: accuracy

	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

$1/5 = 0.2$

Any problems with this?

Doesn't capture "near" correct

	ranking	prediction	prediction
$f_1, f_2, \ldots, f_n$	1	1	1
$f_1, f_2, \ldots, f_n$	2	3	5
$f_1, f_2, \ldots, f_n$	3	2	4
$f_1, f_2, \ldots, f_n$	4	5	3
$f_1, f_2, \ldots, f_n$	5	4	2

$$1/5 = 0.2$$

Idea 2: correlation

ranking prediction prediction

1	1	1
2	3	5
3	2	4
4	5	3
5	4	2
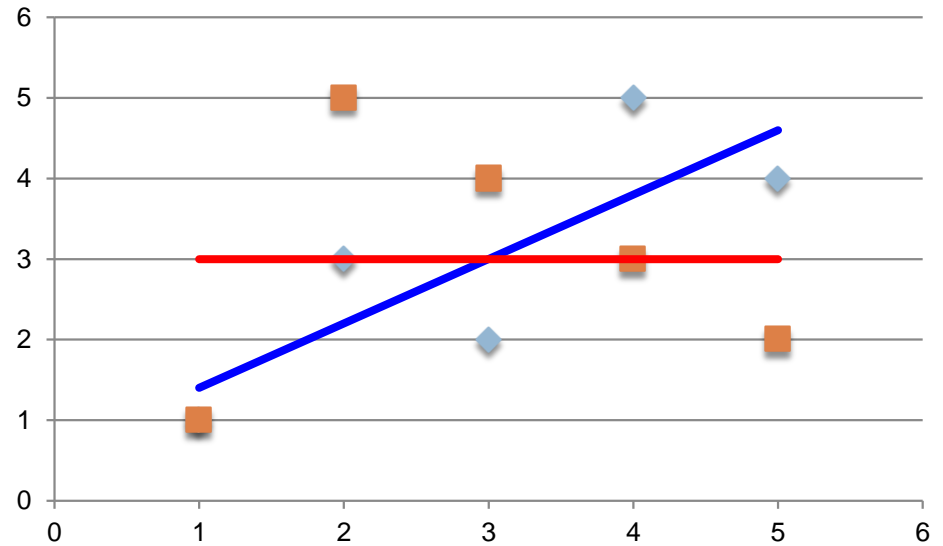
Look at the correlation between the ranking and the prediction

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