BEYOND BINARY CLASSIFICATION

David Kauchak
CS 158 – Fall 2019

Admin

Assignment 4

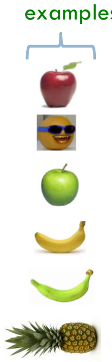
Assignment 3 early next week

If you need assignment feedback…

Multiclass classification

examples

label



apple

orange

apple

banana

banana

pineapple

Same setup where we have a set of features for each example

Rather than just two labels, now have 3 or more

real-world examples?

Real world multiclass classification

document classification

protein classification

handwriting recognition

face recognition

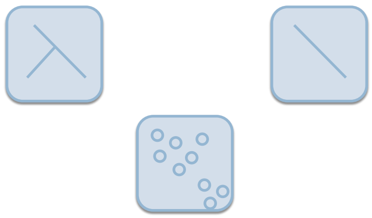
most real-world applications tend to be multiclass

sentiment analysis

autonomous vehicles

emotion recognition

Multiclass: current classifiers



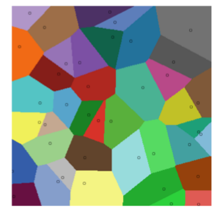
Any of these work out of the box?
With small modifications?

k-Nearest Neighbor (k-NN)

To classify an example ***d***:

- Find ***k*** nearest neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors

No algorithmic changes!



Decision Tree learning

Base cases:

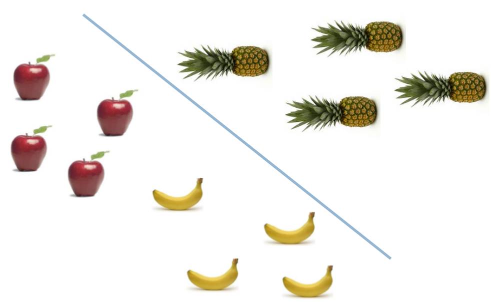
1. If all data belong to the same class, pick that label
2. If all the data have the same feature values, pick majority label
3. If we're out of features to examine, pick majority label
4. If the we don't have any data left, pick majority label of *parent*
5. *If some other stopping criteria* exists to avoid overfitting, pick majority label

Otherwise:

- calculate the "score" for each feature if we used it to split the data
- pick the feature with the highest score, partition the data based on that data value and call recursively

No algorithmic changes!

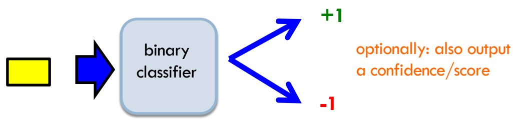
Perceptron learning



Hard to separate three classes with just one line ☹

Black box approach to multiclass

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



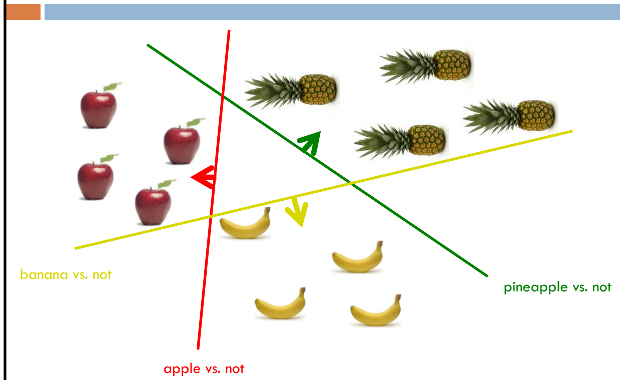
Can we solve our multiclass problem with this?

Approach 1: One vs. all (OVA)

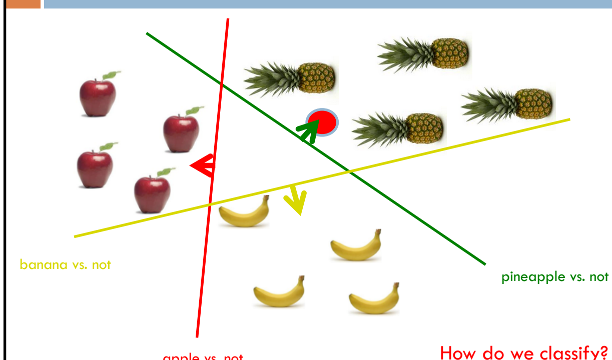
Training: for each label *L*, pose as a binary problem
- all examples with label *L* are positive
- all other examples are negative

	apple vs. not	orange vs. not	banana vs. not
apple	+1	-1	-1
orange	-1	+1	-1
apple	+1	-1	-1
banana	-1	-1	+1
banana	-1	-1	+1

OVA: linear classifiers (e.g. perceptron)

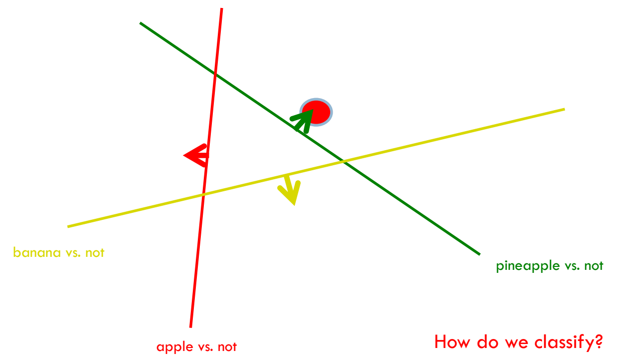


OVA: linear classifiers (e.g. perceptron)

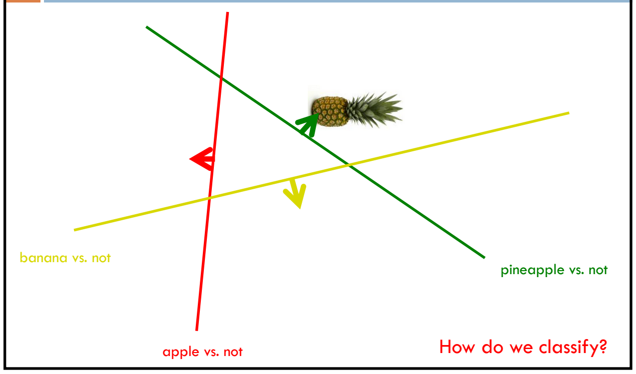


How do we classify?

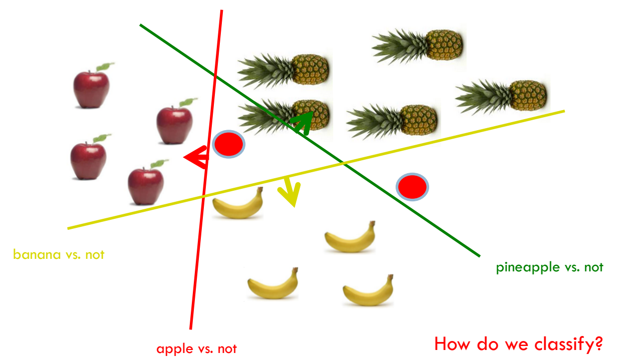
OVA: linear classifiers (e.g. perceptron)



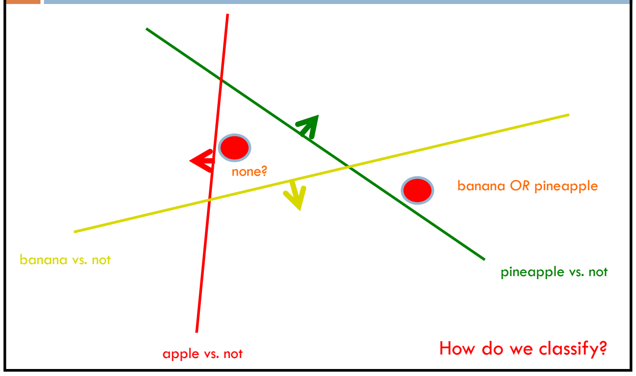
OVA: linear classifiers (e.g. perceptron)



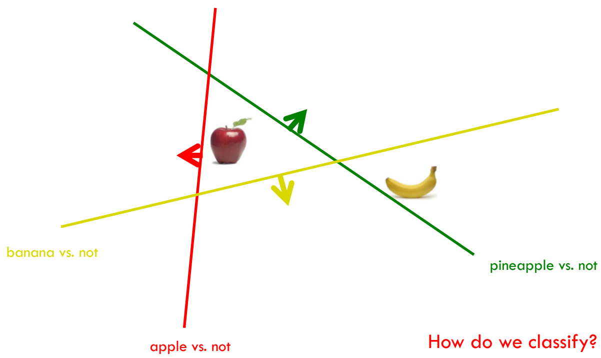
OVA: linear classifiers (e.g. perceptron)



OVA: linear classifiers (e.g. perceptron)



OVA: linear classifiers (e.g. perceptron)

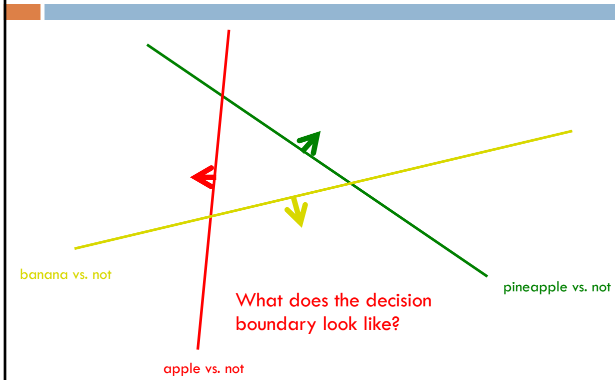


OVA: classify

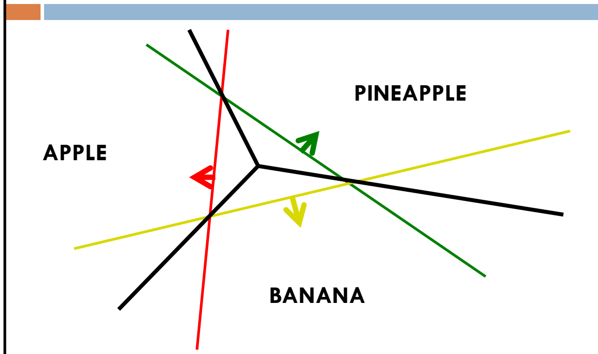
Classify:

- If classifier doesn't provide confidence (this is rare) and there is ambiguity, pick one of the ones in conflict
- Otherwise:
  - pick the most confident positive
  - if none vote positive, pick *least* confident negative

OVA: linear classifiers (e.g. perceptron)



OVA: linear classifiers (e.g. perceptron)



OVA: classify, perceptron

Classify:

- If classifier doesn't provide confidence (this is rare) and there is ambiguity, pick majority in conflict
- Otherwise:
  - pick the most confident positive
  - if none vote positive, pick *least* confident negative

How do we calculate this for the perceptron?

OVA: classify, perceptron

Classify:

- If classifier doesn't provide confidence (this is rare) and there is ambiguity, pick majority in conflict
- Otherwise:
  - pick the most confident positive
  - if none vote positive, pick *least* confident negative

$$prediction = b + \sum_{i=1}^{n} w_i f_i$$

Distance from the hyperplane

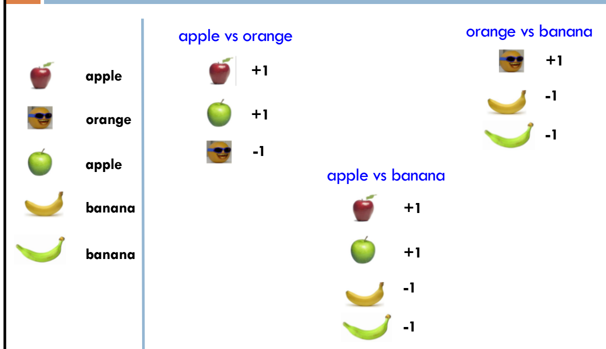
Approach 2: All vs. all (AVA)

Training:

For each pair of labels, train a classifier to distinguish between them

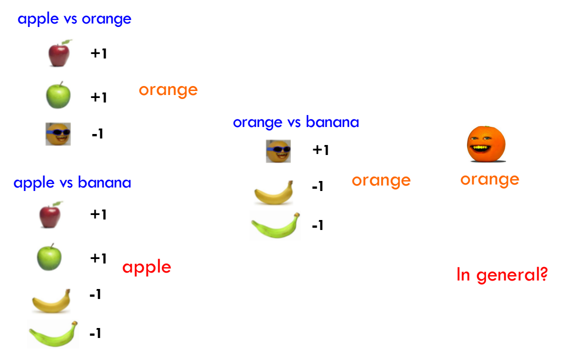
for i = 1 to number of labels:
  for k = i+1 to number of labels:
    train a classifier to distinguish between $label_j$ and $label_k$:
      - create a dataset with all examples *with* $label_j$ labeled positive and all examples with $label_k$ labeled negative
      - train classifier on this subset of the data

AVA training visualized



AVA classify

AVA classify



AVA classify

To classify example e, classify with each classifier f_{jk}

We have a few options to choose the final class:

- Take a majority vote
- Take a weighted vote based on confidence
  - $y = f_{jk}(e)$
  - $score_j$ += y   How does this work?
  - $score_k$ -= y

*Here we're assuming that y encompasses both the prediction (+1,-1) and the confidence, i.e. y = prediction * confidence.*

AVA classify

Take a weighted vote based on confidence

- $y = f_{jk}(e)$
- $score_j$ += y
- $score_k$ -= y

If y is positive, classifier thought it was of type j:
- raise the score for j
- lower the score for k

if y is negative, classifier thought it was of type k:
- lower the score for j
- raise the score for k

OVA vs. AVA

Train/classify runtime?

Error? Assume each binary classifier makes an error with probability ε

OVA vs. AVA

Train time:

AVA learns more classifiers, however, they're trained on much smaller data this tends to make it faster if the labels are equally balanced

Test time:

AVA has more classifiers, so often is slower

Error (see the book for more justification):

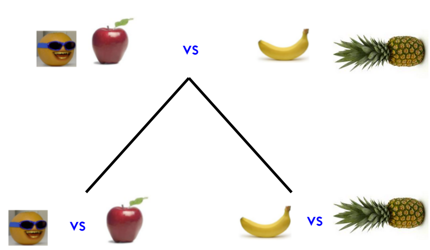
- AVA trains on more balanced data sets
- AVA tests with more classifiers and therefore has more chances for errors

- Theoretically:

-- OVA: ε (number of labels -1)

-- AVA: 2 ε (number of labels -1)

Approach 3: Divide and conquer



Pros/cons vs. AVA?

Multiclass summary

If using a binary classifier, the most common thing to do is OVA

Otherwise, use a classifier that allows for multiple labels:

- DT and k-NN work reasonably well
- We'll see a few more in the coming weeks that will often work better

Multiclass evaluation

label	prediction
apple	orange
orange	orange
apple	apple
banana	pineapple
banana	banana
pineapple	pineapple

How should we evaluate?

Multiclass evaluation

label	prediction
apple	orange
orange	orange
apple	apple
banana	pineapple
banana	banana
pineapple	pineapple

Accuracy: 4/6

Multiclass evaluation imbalanced data

label	prediction
apple	orange
…	
apple	apple
banana	pineapple
banana	banana
pineapple	pineapple

Any problems?

Data imbalance!

Macroaveraging vs. microaveraging

microaveraging: average over examples (this is the "normal" way of calculating)

macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

What effect does this have?
Why include it?

Macroaveraging vs. microaveraging

microaveraging: average over examples (this is the "normal" way of calculating)

macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

- Puts more weight/emphasis on rarer labels
- Allows another dimension of analysis

Macroaveraging vs. microaveraging

label	prediction
apple	orange
orange	orange
apple	apple
banana	pineapple
banana	banana
pineapple	pineapple

microaveraging: average over examples

macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

?

Macroaveraging vs. microaveraging

label	prediction
apple	orange
orange	orange
apple	apple
banana	pineapple
banana	banana
pineapple	pineapple

microaveraging: 4/6

macroaveraging:

apple = 1/2
orange = 1/1
banana = 1/2
pineapple = 1/1
total = (1/2 + 1 + 1/2 + 1)/4
= 3/4

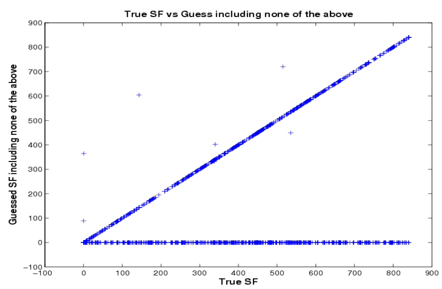
Confusion matrix

entry *(i, j)* represents the number of examples with label *i* that were predicted to have label *j*

another way to understand both the data and the classifier

	Classic	Country	Disco	Hiphop	Jazz	Rock
Classic	86	2	0	4	18	1
Country	1	57	5	1	12	13
Disco	0	6	55	4	0	5
Hiphop	0	15	28	90	4	18
Jazz	7	1	0	0	37	12
Rock	6	19	11	0	27	48

Confusion matrix



BLAST classification of proteins in 850 superfamilies

Multilabel vs. multiclass classification

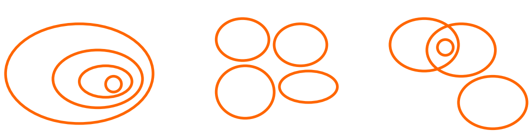
• Is it edible? • Is it sweet? • Is it a fruit? • Is it a banana?	Is it a banana? Is it an apple? Is it an orange? Is it a pineapple?	Is it a banana? Is it yellow? Is it sweet? Is it round?

Any difference in these labels/categories?

Multilabel vs. multiclass classification

• Is it edible? • Is it sweet? • Is it a fruit? • Is it a banana?	Is it a banana? Is it an apple? Is it an orange? Is it a pineapple?	Is it a banana? Is it yellow? Is it sweet? Is it round?

Different structures



Nested/ Hierarchical   Exclusive/ Multiclass   General/Structured

Multiclass vs. multilabel

Multiclass: each example has one label and exactly one label

Multilabel: each example has ***zero or more*** labels. Also called annotation

Multilabel applications?

Multilabel

Image annotation

Document topics

Labeling people in a picture

Medical diagnosis

Ranking problems

Suggest a simpler word for the word below:

vital

Suggest a simpler word

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

Suggest a simpler word for the word below:

acquired

Suggest a simpler word

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

Suggest a simpler word

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

…

training data

train

list of synonyms → ranker → list ranked by simplicity

Ranking problems in general

ranking1 ranking2 ranking3 …

$f_1, f_2, \ldots, f_n$

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

train

$f_1, f_2, \ldots, f_n$ → ranker → ranking/ordering or examples

Ranking problems in general

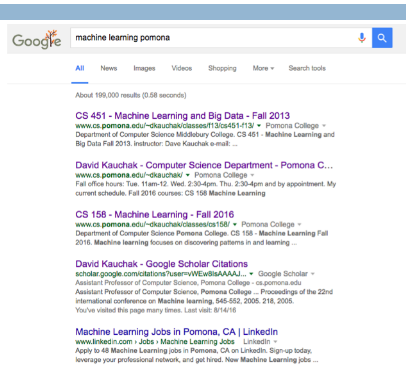
ranking1 ranking2 ranking3 …

$f_1, f_2, \ldots, f_n$

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

Real-world ranking problems?

Search



Ranking Applications

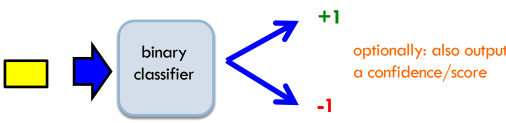
reranking N-best output lists
- machine translation
- computational biology
- parsing
- …

flight search

…

Black box approach to ranking

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



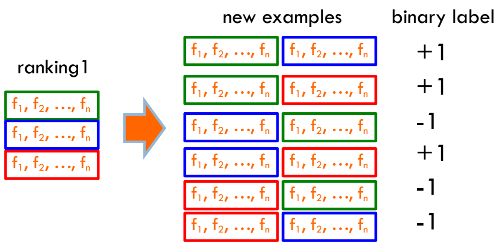
Can we solve our ranking problem with this?

Predict better vs. worse

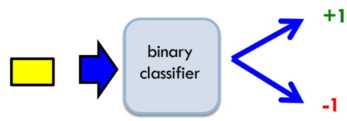
Train a classifier to decide if the first input is better than second:
- Consider all possible pairings of the examples in a ranking
- Label as positive if the first example is higher ranked, negative otherwise

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

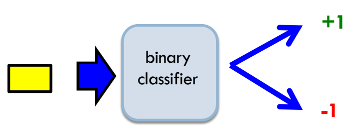


Predict better vs. worse

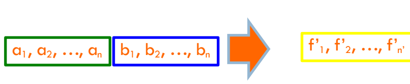


Our binary classifier only takes one example as input

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.

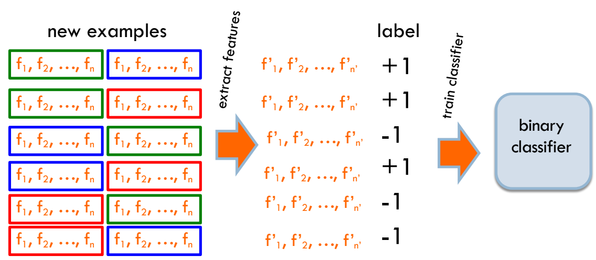
Combined feature vector

Many approaches! Will depend on domain and classifier

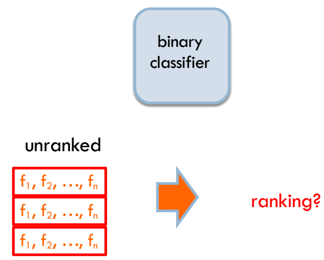
Two common approaches:

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

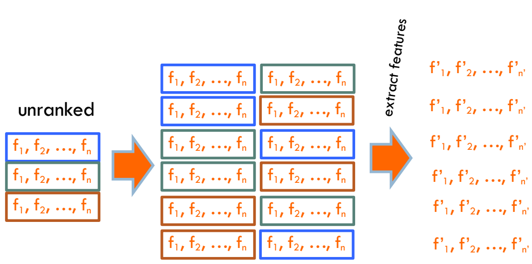
Training



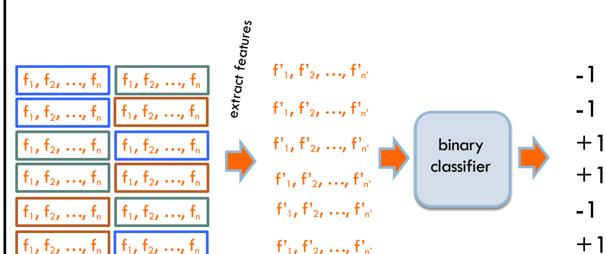
Testing



Testing



Testing



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] -= $f_{jk}(e_{jk})$

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

rank according to label scores

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

An improvement?

ranking1

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

new examples		binary label
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	+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

Are these two examples the same?

Weighted binary classification

ranking1

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

new examples		weighted label
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	+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

Weight based on ***distance*** in ranking

Weighted binary classification

ranking1

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

new examples		weighted label
$f_1, f_2, \ldots, f_n$	$f_1, f_2, \ldots, f_n$	+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

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
$f_1, f_2, \ldots, f_n$	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

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