Multiclass and Multi-label Classification

INFO-4604, Applied Machine Learning
University of Colorado Boulder

September 25, 2018
Prof. Michael Paul

Today

Beyond binary classification

- All classifiers we've looked at so far have predicted one of two classes

- We'll learn two main ways of predicting one of many classes:
  - Repurposing binary classifiers
  - Extending logistic regression

Outputting multiple labels

- Sometimes straightforward, but sometimes not

- Tricks for better results

Multiclass Classification

What color is the cat in this photo?

Calico

Orange Tabby

Tuxedo

Multiclass Classification

Multiclass classification refers to the setting when there are > 2 possible class labels.

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

- It's possible to create multiclass classifiers out of binary classifiers.

One versus Rest

One-vs-rest (or **one-vs-all**) classification involves training a binary classifier for each class

- Each classifier predicts whether the instance belongs to the target class or not

One versus Rest

One-vs-rest (or **one-vs-all**) classification involves training a binary classifier for each class

- Each classifier predicts whether the instance belongs to the target class or not

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

One versus Rest

One-vs-rest (or **one-vs-all**) classification involves training a binary classifier for each class

- Each classifier predicts whether the instance belongs to the target class or not

"Calico" classifier

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Yes
2.50	1.00	4.87	5.95	No
-2.34	-1.24	-0.88	-1.31	No
0.55	0.59	-3.08	1.27	No
2.08	-3.46	4.62	-1.13	No
…	…	…	…	…

One versus Rest

One-vs-rest (or **one-vs-all**) classification involves training a binary classifier for each class

- Each classifier predicts whether the instance belongs to the target class or not

"Orange Tabby" classifier

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	No
2.50	1.00	4.87	5.95	Yes
-2.34	-1.24	-0.88	-1.31	No
0.55	0.59	-3.08	1.27	Yes
2.08	-3.46	4.62	-1.13	No
…	…	…	…	…

One versus Rest

What color is the cat in this photo?

Classifier	Prediction
Calico	No
Orange Tabby	Yes
Tuxedo	No
Gray Tabby	No
...	...

One versus Rest

What color is the cat in this photo?

Classifier	Prediction
Calico	No
Orange Tabby	**Yes**
Tuxedo	No
Gray Tabby	No
...	...

We'll go with *Orange Tabby* as the best prediction.

One versus Rest

What color is the cat in this photo?

Classifier	Prediction
Calico	No
Orange Tabby	Yes
Tuxedo	No
Gray Tabby	Yes
...	...

What if multiple classifiers said yes?

One versus Rest

What color is the cat in this photo?

Classifier	Prediction
Calico	No
Orange Tabby	No
Tuxedo	No
Gray Tabby	No
...	...

What if none of the classifiers said yes?

One versus Rest

Instead of only using the final binary prediction of each classifier, consider the **score** associated with the prediction.

Recall:

We defined a classification score for the linear classifiers we've seen as the dot product $\mathbf{w}^\top \mathbf{x_i}$

- Other kinds of classifiers usually have some sort of score, but it might look different

Go with whichever one-vs-rest classifier has the highest score (highest confidence in prediction)

One versus Rest

What color is the cat in this photo?

Classifier	Score
Calico	-4.59
Orange Tabby	2.18
Tuxedo	-1.80
Gray Tabby	0.73
...	...

One versus Rest

What color is the cat in this photo?

Classifier	Score
Calico	-4.59
Orange Tabby	**2.18**
Tuxedo	-1.80
Gray Tabby	0.73
...	...

We'll go with *Orange Tabby* as the best prediction.

All Pairs

The **all pairs** approach to multiclass classification trains a binary classifier for every pair of classes

- Whichever class "wins" more pairwise classifications will be the final prediction

All Pairs

The **all pairs** approach to multiclass classification trains a binary classifier for every pair of classes

- Whichever class "wins" more pairwise classifications will be the final prediction

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

All Pairs

The **all pairs** approach to multiclass classification trains a binary classifier for every pair of classes

- Whichever class "wins" more pairwise classifications will be the final prediction

"Calico vs Tuxedo" classifier

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

All Pairs

The **all pairs** approach to multiclass classification trains a binary classifier for every pair of classes

- Whichever class "wins" more pairwise classifications will be the final prediction

"Calico vs Orange Tabby" classifier

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

All Pairs

The **all pairs** approach to multiclass classification trains a binary classifier for every pair of classes

- Whichever class "wins" more pairwise classifications will be the final prediction

"Tuxedo vs Orange Tabby" classifier

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
…	…	…	…	…

All Pairs

What color is the cat in this photo?

Classifier	Prediction
Calico vs Orange	Orange
Calico vs Tuxedo	Tuxedo
Calico vs Gray	Gray
Orange vs Tuxedo	Orange
Orange vs Gray	Orange
...	...

All Pairs

What color is the cat in this photo?

Classifier	Prediction
Calico vs Orange	**Orange**
Calico vs Tuxedo	Tuxedo
Calico vs Gray	Gray
Orange vs Tuxedo	**Orange**
Orange vs Gray	**Orange**
…	…

We'll go with *Orange Tabby* as the best prediction.

Multiclass Classification

- These approaches can work reasonably well
- All pairs is faster to train; one-vs-rest is faster at making predictions
- sklearn implements one-vs-rest by default when you give more than two classes to a binary classifier

Next we'll see how logistic regression can handle multiple classes without having to combine different binary classifiers

Logistic Regression

Before:

Binary logistic regression used the logistic function to give the probability that an instance belonged to the positive class.

$$P(y_i = 1 \mid x_i) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x}_i)}$$

Logistic Regression

Multinomial (or **multivariate**) logistic regression uses a similar but more general function (the *softmax* function) for the probability of *K* classes:

$$P(y_i = k \mid x_i) = \frac{\exp(\mathbf{w_k}^\top \mathbf{x_i})}{\displaystyle\sum_{k'=1}^{K} \exp(\mathbf{w_{k'}}^\top \mathbf{x_i})}$$

Logistic Regression

Binary	Multinomial
• One weight vector **w**	• K weight vectors, $\mathbf{w_k}$
• Score plugged into logistic function to get value between [0, 1]	• Vector of K scores plugged into softmax function get to vector of K values, each between [0,1] and all values sum to 1
• Probability of negative class is just 1 minus probability of positive class	• Each class probability depends on its own score from its own weight vector

Logistic Regression

What color is the cat in this photo?

Class	Probability
Calico	0.03
Orange Tabby	0.62
Tuxedo	0.04
Gray Tabby	0.11
…	…

Logistic Regression

What color is the cat in this photo?

Class	Probability
Calico	0.03
Orange Tabby	**0.62**
Tuxedo	0.04
Gray Tabby	0.11
...	...

Orange Tabby has the highest probability.

Logistic Regression

The weights can be learned with gradient descent, just like in the binary version.

The loss function is the negative log-likelihood of the training data, as before.

Won't go into the details in this class, but updates look similar to what you've seen.

Logistic Regression

Other names for multinomial logistic regression that you might encounter:

- Multiclass logistic regression

- Maximum entropy (MaxEnt) classifier

- Softmax regression

Multi-label Classification

What color and sex is the cat in this photo?

Calico
Female

Orange Tabby
Male

Tuxedo
Male

Multi-label Classification

Multi-label classification refers to the setting when there > 1 label you want to predict.

x_1	x_2	x_3	x_4	y_1	y_2
1.01	-4.26	7.99	-0.03	Calico	Female
2.50	1.00	4.87	5.95	Orange Tabby	Male
-2.34	-1.24	-0.88	-1.31	Tuxedo	Male
0.55	0.59	-3.08	1.27	Orange Tabby	Male
2.08	-3.46	4.62	-1.13	Gray Tabby	Female
…	…	…	…	…	…

Multi-label Classification

Starting point: train two separate classifiers

- One predicts sex

- One predicts color

This might work fine, but there are some things to think about when doing this.

Multi-label Classification

Two independent classifiers might output combinations of labels that don't make sense

- Calico cats are almost always female

- If your classifiers predict male and calico, this is probably wrong

There might be correlations between the classes that you could help classification if you had a way to combine the two classifiers

- Orange cats are more often male (~80% of the time)

- If your classifier(s) believed the cat was orange, this would increase the belief that it is male (or vice versa)

Multi-label Classification

One idea: train one classifier first, use its output as a feature in the other.

Example:
First train a classifier to predict color:

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico
2.50	1.00	4.87	5.95	Orange Tabby
-2.34	-1.24	-0.88	-1.31	Tuxedo
0.55	0.59	-3.08	1.27	Orange Tabby
2.08	-3.46	4.62	-1.13	Gray Tabby
...	...	...	...	...

Then train a classifier to predict sex, using the predicted color as an additional feature.

x_1	x_2	x_3	x_4	x_5	y
1.01	-4.26	7.99	-0.03	Calico?	Female
2.50	1.00	4.87	5.95	Orange Tabby?	Male
-2.34	-1.24	-0.88	-1.31	Tuxedo?	Male
0.55	0.59	-3.08	1.27	Orange Tabby?	Male
2.08	-3.46	4.62	-1.13	Gray Tabby?	Female
...	...	...	...	...	...

Multi-label Classification

One idea: train one classifier first, use its output as a feature in the other.

Limitations:

- If the first classifier is wrong, you'll have an incorrect feature value.

- This is a "pipeline" approach where one classifier informs the other, rather than both informing each other simultaneously

Multi-label Classification

Another idea: treat combinations of classes as their own "classes", then do single-label classification

x_1	x_2	x_3	x_4	y
1.01	-4.26	7.99	-0.03	Calico + Female
2.50	1.00	4.87	5.95	Orange Tabby + Male
-2.34	-1.24	-0.88	-1.31	Tuxedo + Male
0.55	0.59	-3.08	1.27	Orange Tabby + Male
2.08	-3.46	4.62	-1.13	Gray Tabby + Female
…	…	…	…	…

Multi-label Classification

Another idea: treat combinations of classes as their own "classes", then do single-label classification

This way you can learn that "Calico + Male" is very unlikely, etc.

Limitations:

- All classes are learned independently:
the classifier has no idea that "Tuxedo+Male" and "Tuxedo+Female" are both the same color and therefore probably have similar feature weights

Summary

Multiclass and multi-label situations arise often.

- Some simple solutions exist that are often effective.
- More sophisticated solutions exist; some we will see later in the semester.

Don't confuse "multiclass" and "multi-label"!

- They are independent concepts.
- Something can be multiclass but not multi-label, or vice versa, or both, or neither.