

Richer output spaces

Multiclass classification

We have mostly discussed binary classification problems, with $|Y| = 2$.
Do the methods we've studied generalize to cases with $k > 2$ labels?

- Nearest neighbor?
- Generative models?
- Linear classifiers?

Linear classifiers seem inherently binary: there are just two sides of the boundary! How can they be extended to multiple classes?

Multiclass logistic regression

p :

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^p$, the classifier is given by $w \in \mathbb{R}^p$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x}}{1 + e^{w \cdot x}}.$$

When $\mathcal{Y} = \{1, 2, \dots, k\}$, specify a classifier by $w_1, \dots, w_k \in \mathbb{R}^p$:

$$\Pr(y = j|x) = \frac{e^{w_j \cdot x}}{e^{w_1 \cdot x} + \dots + e^{w_k \cdot x}}.$$

Multiclass logistic regression

p :

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^p$, the classifier is given by $w \in \mathbb{R}^p$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x}}{1 + e^{w \cdot x}}.$$

When $\mathcal{Y} = \{1, 2, \dots, k\}$, specify a classifier by $w_1, \dots, w_k \in \mathbb{R}^p$:

$$\Pr(y = j|x) = \frac{e^{w_j \cdot x}}{e^{w_1 \cdot x} + \dots + e^{w_k \cdot x}}.$$

Prediction: given a point x , predict label

$$\arg \max_j \Pr(y = j|x) = \arg \max_j w_j \cdot x$$

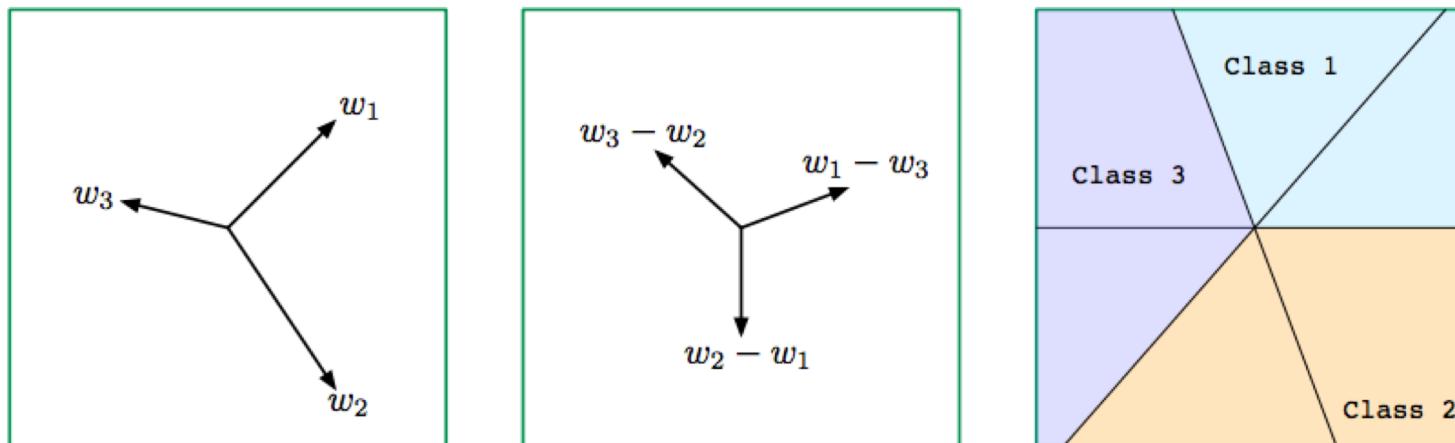
Learning: given data $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^p \times \mathcal{Y}$, find vectors $w_1, \dots, w_k \in \mathbb{R}^p$ that maximize the likelihood

$$\prod_{i=1}^n \Pr(y^{(i)}|x^{(i)}).$$

Taking negative log gives a convex minimization problem.

Multiclass prediction with linear functions

- $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \dots, k\}$.
- **Model:** $w_1, \dots, w_k \in \mathbb{R}^p$, one per class.
- **Prediction:** On instance x , predict label $\arg \max_j w_j \cdot x$.



Each class is the intersection of half-spaces through the origin.

A half-space is a set of points that satisfy a single inequality constraint $Ax \leq b$.

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \dots, k\}$

Model: $w_1, \dots, w_k \in \mathbb{R}^p$, one per class.

Prediction: On instance x , predict label $\arg \max_j w_j \cdot x$.

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \dots, k\}$

Model: $w_1, \dots, w_k \in \mathbb{R}^p$, one per class.

Prediction: On instance x , predict label $\arg \max_j w_j \cdot x$.

Learning. Given training set $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:

- Initialize $w_1 = \dots = w_k = 0$
- Repeat while some training point (x, y) is misclassified:

$$\text{for correct label } y: \quad w_y = w_y + x$$

$$\text{for predicted label } \hat{y}: \quad w_{\hat{y}} = w_{\hat{y}} - x$$

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \dots, k\}$

Model: $w_1, \dots, w_k \in \mathbb{R}^p$, one per class.

Prediction: On instance x , predict label $\arg \max_j w_j \cdot x$.

Learning. Given training set $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:

- Initialize $w_1 = \dots = w_k = 0$
- Repeat while some training point (x, y) is misclassified:

$$\text{for correct label } y: \quad w_y = w_y + x$$

$$\text{for predicted label } \hat{y}: \quad w_{\hat{y}} = w_{\hat{y}} - x$$

Guarantee: Suppose all $\|x^{(i)}\| \leq R$ and that there exist unit-length $u_1, \dots, u_k \in \mathbb{R}^p$ and “margin” $\gamma > 0$ such that for all i and all $y \neq y^{(i)}$,

$$u_{y^{(i)}} \cdot x^{(i)} - u_y \cdot x^{(i)} \geq \gamma.$$

Then the multiclass perceptron algorithm makes at most $2kR^2/\gamma^2$ updates.

Multiclass SVM

Model: $w_1, \dots, w_k \in \mathbb{R}^P$, one per class.

Prediction: On instance x , predict label $\arg \max_j w_j \cdot x$.

Learning. Given $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^P \times \{1, \dots, k\}$:

$$\begin{aligned} & \min_{w_1, \dots, w_k \in \mathbb{R}^P, \xi \in \mathbb{R}^n} \quad \frac{1}{2} \sum_{j=1}^k \|w_j\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.: } & w_{y^{(i)}} \cdot x^{(i)} - w_y \cdot x^{(i)} \geq 1 - \xi_i \quad \text{for all } i \text{ and all } y \neq y^{(i)} \\ & \xi \geq 0 \end{aligned}$$

Once again, a convex optimization problem.

Quick quiz

Suppose we have input space $X = \mathbf{R}^p$ and label space $Y = \{1, 2, \dots, k\}$, and we have a training set of size n .

- ① If we use multiclass SVM, how many variables does the primal program have?

Quick quiz

Suppose we have input space $X = \mathbb{R}^p$ and label space $Y = \{1, 2, \dots, k\}$, and we have a training set of size n .

- ① If we use multiclass SVM, how many variables does the primal program have? $k.p$

Quick quiz

Suppose we have input space $X = \mathbb{R}^p$ and label space $Y = \{1, 2, \dots, k\}$, and we have a training set of size n .

- ① If we use multiclass SVM, how many variables does the primal program have? $k.p$
- ② How many constraints does it have?

Quick quiz

Suppose we have input space $X = \mathbb{R}^p$ and label space $Y = \{1, 2, \dots, k\}$, and we have a training set of size n .

- ① If we use multiclass SVM, how many variables does the primal program have? $k.p$
- ② How many constraints does it have? $n.(k-1)$

Structured output spaces: examples

Part-of-speech tagging.

the/D cat/N bit/V the/D dog/N

Inaccurate to treat each tag as a separate prediction problem.

E.g. bit (N or V?)

To score a candidate tagging y of a sentence x , add up:

- Score for each (word, tag)
- Score for each trigram (tag1, tag2, tag3)
- Other such component scores

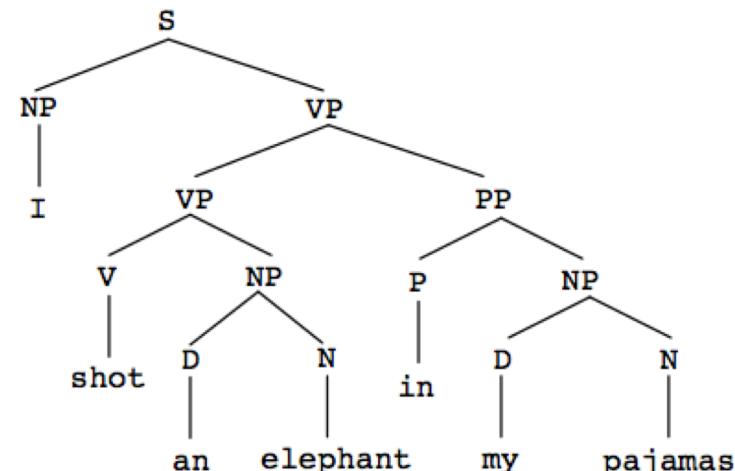
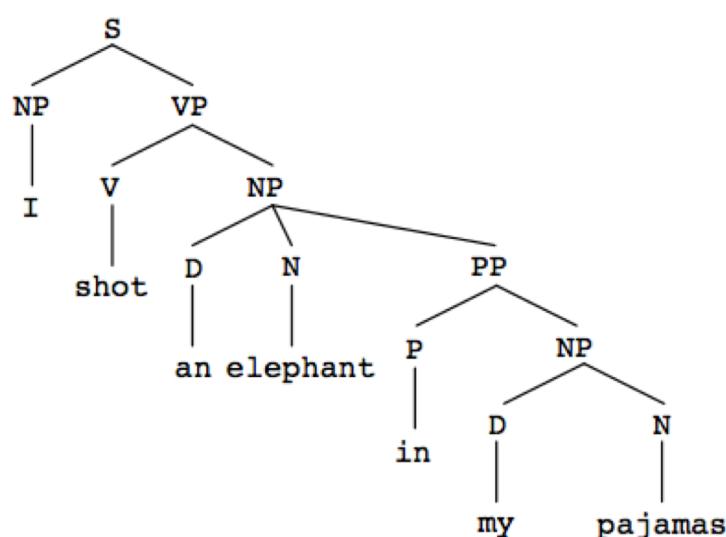
To tag a given sentence x : find the tagging y with maximum score. Can be done efficiently by dynamic programming.

Structured output spaces: examples

Parsing.

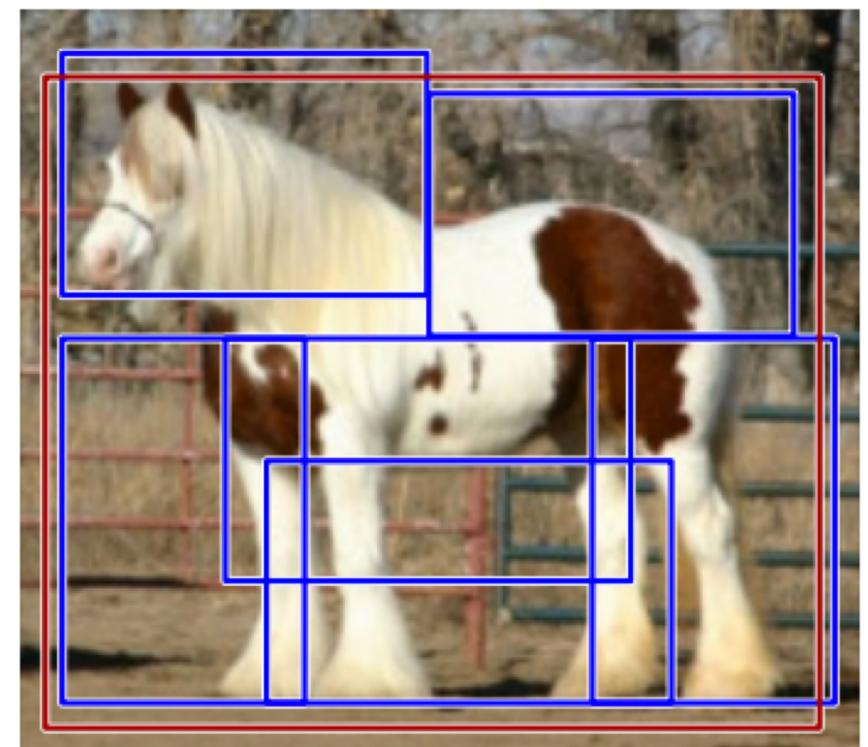
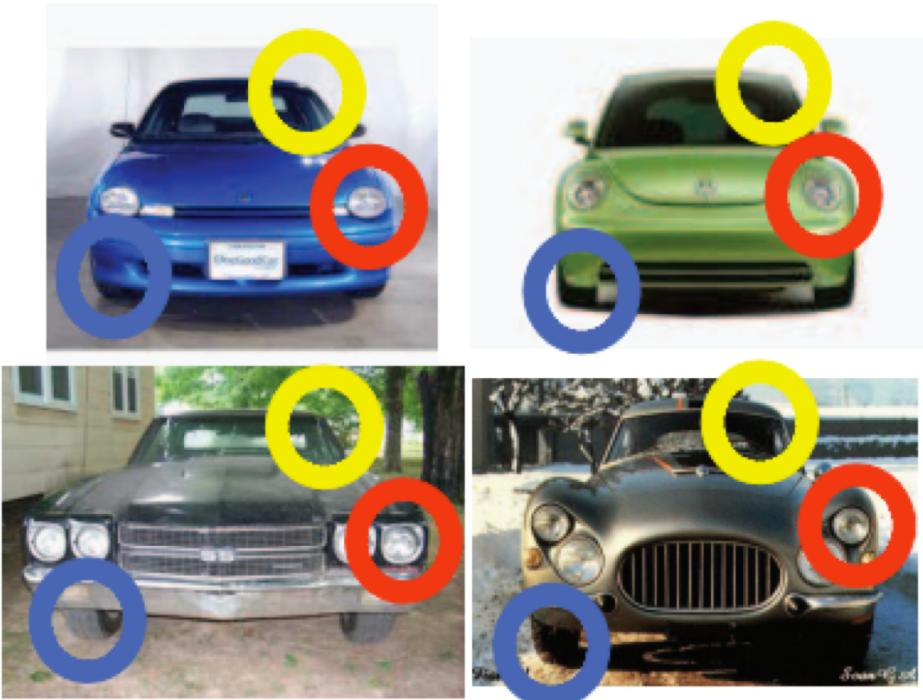
Groucho Marx (1930): While hunting in Africa, I shot an elephant in my pajamas. How an elephant got into my pajamas I'll never know.

Here are two possible parse trees y for the sentence $x = \text{"I shot an elephant in my pajamas"}$.



Structured output spaces: examples

Parts-based object recognition.



Structured output prediction

How to handle such output spaces Y ?

- Features based on both the input and output.

For any instance (e.g. sentence) x and candidate output (e.g. part-of-speech tagging) y , let

$$\phi_1(x, y), \phi_2(x, y), \dots, \phi_k(x, y)$$

be features that give a sense of whether y is a desirable output for x . For instance: all word-tag pairs and tag trigrams.

Package these features into a vector:

$$\Phi(x, y) = (\phi_1(x, y), \phi_2(x, y), \dots, \phi_k(x, y))$$

- Score outputs based on a linear function of the features.

The score for output $y \in \mathcal{Y}$ is $w \cdot \Phi(x, y)$, where $w \in \mathbb{R}^k$.

- Predict the highest-scoring output.

For instance x , return $\arg \max_y w \cdot \Phi(x, y)$. This can often be done efficiently with dynamic programming.

Learning task: given data, find a suitable weight vector w .

Structured-output Perceptron

Given training data $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y}$:

- Initialize $w = 0$
- Repeat until satisfied:
 - For $i = 1$ to n :

$$\text{Prediction: } \hat{y} = \arg \max_y w \cdot \Phi(x^{(i)}, y)$$

$$\text{If } y^{(i)} \neq \hat{y}: \quad w = w + \Phi(x^{(i)}, y^{(i)}) - \Phi(x^{(i)}, \hat{y})$$

Convergence guarantee under a margin condition, as before.

Quick quiz

How does structured-output perceptron generalize multiclass perceptron?

Multiclass perceptron

- Initialize $w_1 = \dots = w_k = 0$
- Repeat while some (x, y) is misclassified:
(Prediction is $\hat{y} = \arg \max_y w_y \cdot x$.)

$$\text{for correct label } y: \quad w_y = w_y + x$$

$$\text{for predicted label } \hat{y}: \quad w_{\hat{y}} = w_{\hat{y}} - x$$

Structured-output perceptron

- Initialize $w = 0$
- Repeat while some (x, y) is misclassified:
(Prediction is $\hat{y} = \arg \max_y w \cdot \Phi(x, y)$.)

$$w = w + \Phi(x, y) - \Phi(x, \hat{y})$$

Quick quiz

$$W = \begin{matrix} w_1 \\ \vdots \\ w_k \end{matrix} \quad \phi(x,y) = \begin{matrix} x \\ \vdots \\ 0 \end{matrix}$$

$\phi(x,y)$ has value ‘x’ for row ‘y’
and 0 everywhere else.

Structured-output SVM

Loss function.

Not all errors are equal, especially when the outputs have many parts.

Let $\Delta(y, \hat{y})$ be the loss when predicting \hat{y} instead of y .

Learning. Given $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y}$:

$$\min_{w \in \mathbb{R}^k, \xi \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

$$w \cdot \Phi(x^{(i)}, y^{(i)}) - w \cdot \Phi(x^{(i)}, y) \geq \Delta(y^{(i)}, y) - \xi_i \quad \text{for all } i \text{ and all } y \neq y^{(i)}$$
$$\xi \geq 0$$

Clever optimization tricks are needed to solve this efficiently.