多分类及复杂预测问题

刘新旺

https://xinwangliu.github.io/

国防科技大学　计算机学院

计算科学系人工智能与大数据教研室

2020 年 11 月 24 日

Content

多分类的应用

Classification tasks with multiclass is common in practical applications. For examples,

- Categorizing documents: economic, political, military, $\cdots$

多分类的应用

Classification tasks with multiclass is common in practical applications. For examples,

- Categorizing documents: economic, political, military, $\cdots$
- Vehicle recognition: bicycle, car, bus, $\cdots$

多分类的应用

Classification tasks with multiclass is common in practical applications. For examples,

- Categorizing documents: economic, political, military, $\cdots$
- Vehicle recognition: bicycle, car, bus, $\cdots$
- Flower recognition: $\cdots$

Content

多分类问题形式化定义

给定训练样本集合 $\mathcal{S} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$，其中 $\{\mathbf{x}_i\}_{i=1}^m \subseteq \mathcal{X}^m$ 独立同分布，$y_i = f(\mathbf{x}_i) \in \mathcal{Y}\,(\forall i = 1, \cdots, m)$。多分类问题的目标是基于数据 $\mathcal{S}$，从假说集合 $\mathcal{H}$ 中选择一个假说 h，以使得 期望误差

$$\mathrm{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathrm{sgn}(h(\mathbf{x})) \neq f(\mathbf{x})\right] \tag{1}$$

最小。

In the multi-class setting, a hypothesis is defined based on a scoring function $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The label associated to point $\mathbf{x}$ is the one resulting in the largest score $h(\mathbf{x}, y)$, which defines the following mapping from $\mathcal{X}$ to $\mathcal{Y}$:

$$\mathbf{x} \mapsto \arg\max_{y\in\mathcal{Y}} \; h(\mathbf{x}, y) \tag{2}$$

The optimization problem defining the multi-class SVM algorithm is:

$$\begin{aligned}
&\min_{\mathbf{W},\boldsymbol{\xi}} \ \frac{1}{2}\sum_{l=1}^{k}\|\mathbf{w}_l\|^2 + C\sum_{i=1}^{m}\xi_i \\
&\text{subject to: } \forall i \in [1,m], \forall l \in \mathcal{Y} - \{y_i\}, \\
&\qquad\qquad \mathbf{w}_{y_i}\cdot\mathbf{\Phi}(x_i) \geq \mathbf{w}_l\cdot\mathbf{\Phi}(x_i) + 1 - \xi_i.
\end{aligned}$$

where $K : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is a PSD kernel and $\mathbf{\Phi} : \mathcal{X}\to\mathcal{H}$ is a feature mapping associated to K.

多分类策略

- One-versus-all:

$$\forall \mathbf{x} \in \mathcal{X},\ h(\mathbf{x}) = \arg\max_{l \in \mathcal{Y}} f_l(\mathbf{x}) \tag{3}$$

- One-versus-one:

$$\forall \mathbf{x} \in \mathcal{X},\ h(\mathbf{x}) = \arg\max_{l' \in \mathcal{Y}} |l :\ h_{ll'}(\mathbf{x}) = 1| \tag{4}$$

where $h_{ll'}(\mathbf{x}) = 1$ indicating l winning over l', then the class predicted by h can be interpreted as the one with the largest number of wins in that tournament.

Content

错误修正编码的基本思想

A more general method for the reduction of multi-class to binary classification is based on the idea of error-correction codes (ECOC).

- It consists of assigning to each class $l \in \mathcal{Y}$ a code word of length $c \geq 1$.
  - In the simplest case, it is a binary vector $\mathbf{M}_l \in \{-1, +1\}^c$.
- $\mathbf{M}_l$ serves as a signature for class l, and together these vectors define a matrix $\mathbf{M} \in \{-1, +1\}^{k \times c}$ whose l-th row is $\mathbf{M}_l$.

错误修正编码：训练和预测

The training and prediction procedure of ECOC.

- The training procedure:
  - For each column $j \in [1, c]$, a binary classifier $h_j : \mathcal{X} \to \{-1, +1\}$ is learned using the full training sample $\mathcal{S}$.
- The prediction procedure
  - For any $\mathbf{x} \in \mathcal{X}$, let $h(\mathbf{x}) = [h_1(\mathbf{x}), \cdots, h_c(\mathbf{x})]$. Then, the multi-class hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is defined by

$$\forall \mathbf{x} \in \mathcal{X},\ h(\mathbf{x}) = \arg\min_{l \in \mathcal{Y}} d_H(\mathbf{M}_l, h(\mathbf{x})) \tag{5}$$

The class predicted is the one whose signatures is the closest to $h(\mathbf{x})$ in Hamming distance.

codes

classes	1	2	3	4	5	6
1	0	0	0	1	0	0
2	1	0	0	0	0	0
3	0	1	1	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	1	0	1	0	0

$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$
0	1	1	0	1	1

new example x

Figure 8.5 Illustration of error-correction codes for multi-class classification. Left: binary code matrix $\mathbf{M}$, with each row representing the code word of length $c = 6$ of a class $l \in [1, 8]$. Right: vector of predictions $\mathbf{h}(x)$ for a test point x. The ECOC classifier assigns label 3 to x, since the binary code for the third class yields the minimal Hamming distance with $\mathbf{h}(x)$ (distance of 1).

与OvA及OvO的联系

Both OVA and OVO become special instances of the ECOC technique by applying ternary codes $\{-1, 0, +1\}$. The examples in classes labeled with 0 are disregarded when training a binary classifier for each column.

- The matrix $\mathbf{M}$ for OVA is a square matrix, that is $c = k$, with all terms equal to -1 except from the diagonal ones which are all equal to $+1$.
- The matrix $\mathbf{M}$ for OVO has $c = k(k-1)/2$ columns. Each column corresponds to a pair of distinct classes (l, l'), $l \neq l'$, with all entries equal to 0 except from the one with row l, which is -1, and the one with row l', which is $+1$.

A interesting extension of ECOC consists of extending discrete codes to continuous ones by letting the matrix entries take arbitrary real values and by using the training samples to learn $\mathbf{M}$. Specifically,

1. Starting with a discrete version of $\mathbf{M}$, c binary classifiers with scoring functions $f_l\,(1 \leq l \leq c)$ are first learned as described previously.
2. The entries of M are relaxed to take real values and learned from the training sample with the objective of making the row of $\mathbf{M}$ corresponding to the class of any point $\mathbf{x} \in \mathcal{X}$ more similar to $F(\mathbf{x})$ than other rows, where $F(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_c(\mathbf{x}))^\top$.

An algorithm for learning $\mathbf{M}$ is similar to the idea just discussed in multi-class SVMs, which can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{M},\boldsymbol{\xi}} \ \|\mathbf{M}\|_{\mathbf{F}}^2 + C\sum_{i=1}^{m}\xi_i \\ &s.t.\ \forall i \in [1:m]\ \forall l \in \mathcal{Y} - \{y_i\} \\ &\qquad K(\mathbf{M}_{y_i}, F(\mathbf{x}_i)) - K(\mathbf{M}_l, F(\mathbf{x}_i)) \geq 1 - \xi_i,\ \xi_i \geq 0, \end{aligned} \tag{6}$$

where the similarity between $\mathbf{M}_l\ (1 \leq l \leq k)$ and $F(\mathbf{x}_i)$ can be measured by any PSD kernel K.

An algorithm for learning $\mathbf{M}$ is similar to the idea just discussed in multi-class SVMs, which can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{M},\boldsymbol{\xi}} \ \|\mathbf{M}\|_{\mathbf{F}}^2 + C\sum_{i=1}^{m}\xi_i \\ &s.t.\ \forall i \in [1:m]\ \forall l \in \mathcal{Y} - \{y_i\} \\ &\qquad K(\mathbf{M}_{y_i}, F(\mathbf{x}_i)) - K(\mathbf{M}_l, F(\mathbf{x}_i)) \geq 1 - \xi_i,\ \xi_i \geq 0, \end{aligned} \tag{6}$$

where the similarity between $\mathbf{M}_l\ (1 \leq l \leq k)$ and $F(\mathbf{x}_i)$ can be measured by any PSD kernel K.

The resulting multi-class classification decision function is:

$$h : \mathbf{x} \mapsto \arg\max_{1 \leq l \leq k} K(\mathbf{M}_l, F(\mathbf{x})). \tag{7}$$

Content

"Normal" Machine Learning:

$$f : \mathcal{X} \to \mathbb{R}.$$

- inputs $\mathcal{X}$ can be any kind of objects
- output y is a real number
  - classification, regression, density estimation, ...

"Normal" Machine Learning:

$$f : \mathcal{X} \to \mathbb{R}.$$

- inputs $\mathcal{X}$ can be any kind of objects
- output y is a real number
  - classification, regression, density estimation, ...

Structured Output Learning:

$$f : \mathcal{X} \to \mathcal{Y}.$$

- inputs $\mathcal{X}$ can be any kind of objects
- outputs $y \in \mathcal{Y}$ are complex (structured) objects
  - images, text, audio, folds of a protein

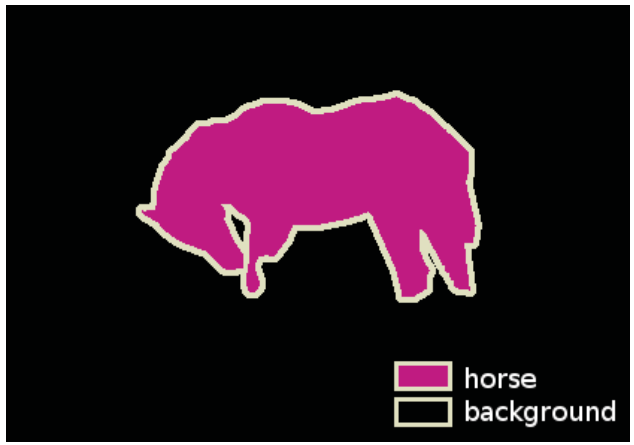
结构化数据的定义

predicting structured outputs from input data.

- Natural Language Processing:
  - Automatic Translation (output: sentences)
  - Sentence Parsing (output: parse trees)
- Speech Processing
  - Text-to-Speech (output: audio signal)
- Robotics:
  - Planning (output: sequence of actions)

计算机视觉样例：语义图像分割

$\longmapsto$



input: images $\longmapsto$ output: segmentation masks

- input space $\mathcal{X} = \{\text{images}\} \triangleq [0, 255]^{3 \cdot M \cdot N}$
- output space $\mathcal{Y} = \{\text{segmentationmasks}\} \triangleq \{0, 1\}^{M \cdot N}$

计算机视觉样例：目标定位

input:
image

output:
object position
$(left, top$
$right, bottom)$

- input space $\mathcal{X} = \{\text{images}\}$
- output space $\mathcal{Y} = \mathbb{R}^4$ bounding box coordinates

The relevant features in structured output problems often depend on both the input and the output. Thus, we denote by $\Phi(\mathbf{x}, y)$ the feature vector associated to a pair $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$.

Unified formulation

The hypothesis set used by most structured prediction algorithms is then defined as the set of functions $h : \mathcal{X} \to \mathcal{Y}$ such that

$$\forall \mathbf{x} \in \mathcal{X},\ h(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \mathbf{w}^\top \Phi(\mathbf{x}, y) \tag{8}$$

for some vector $\mathbf{w}$.

结构化预测的形式化问题

Let $\mathcal{S} = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ be an i.i.d. labeled sample. Since the hypothesis set is linear, we can seek to define an algorithm similar to multi-class SVMs.

Structured SVM

$$\min_{\mathbf{w}, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \max_{y \neq y_i} \max \left\{0, 1 - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right\} \tag{9}$$

However, the loss function used is typically not a zero-one loss but one that depends on the substructures.

结构化预测的形式化问题

Let $\mathcal{S} = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ be an i.i.d. labeled sample. Since the hypothesis set is linear, we can seek to define an algorithm similar to multi-class SVMs.

Structured SVM

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} \max \left\{0, 1 - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right\} \tag{9}$$

However, the loss function used is typically not a zero-one loss but one that depends on the substructures.

The relationship between structured SVMs with conventional SVMs?

结构化预测的形式化问题（续）

Let $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denote a loss function such that $L(y', y)$ measures the penalty of predicting the label $y' \in \mathcal{Y}$ instead of the correct label $y \in \mathcal{Y}$.

- In part-of-speech tagging, $L(y', y)$ could be for example the Hamming distance between y' and y.

We need to take into account the loss function L, that is $L(y, y_i)$ for each $i \in [1, m]$ and $y \in \mathcal{Y}$.

结构化SVM—加性惩罚

The additive penalization leads to the following algorithm known as Maximum Margin Markov Networks (M^3N):

Additive penalization

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} \max\left\{0, L(y_i, y) - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right\} \tag{10}$$

The additive penalization leads to the following algorithm known as Maximum Margin Markov Networks (M^3N):

Additive penalization

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} \max\left\{0, L(y_i, y) - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right\} \tag{10}$$

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} \exp\left(L(y_i, y) - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right) \tag{11}$$

Note: $f : (x_1, \cdots, x_k) \mapsto \log(\sum_{j=1}^{k} \exp(x_i))$ provides a smooth approximation of $(x_1, \cdots, x_k) \mapsto \max\{x_1, \cdots, x_k\}$.

结构化SVM—乘性惩罚

The multiplicative penalization leads to the following algorithm known as Maximum Margin Markov Networks (M^3N):

Multiplicative penalization

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \max_{y \neq y_i} L(y_i, y) \max\left\{0, 1 - \left(\mathbf{w}^\top [\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)]\right)\right\} \tag{12}$$

This problem can be equivalently written as a QP with an infinite number of constraints. In practice, it is solved iteratively by augmenting at each round the finite set of constraints of the previous round with the most violating constraint.

Content

The learning problem of ranking arises in many modern applications, including the design of search engines and movie recommendation systems.

- A standard user of a search engine is not willing to consult all the documents returned in response to a query, but only the top ten or so.
- ...

Basic Notations

Let $\mathcal{X}$ denote the input space. We denote by $\mathcal{D}$ an unknown distribution over $\mathcal{X} \times \mathcal{X}$ according to which pairs of points are drawn and by $f : \mathcal{X} \times \mathcal{X} \to \{-1, 0, +1\}$ a target labeling function or preference function. The three values assigned by f are interpreted as follows:

- $f(\mathbf{x}, \mathbf{x}') = +1$ if $\mathbf{x}'$ is preferred to $\mathbf{x}$ or ranked higher than $\mathbf{x}$,
- $f(\mathbf{x}, \mathbf{x}') = -1$ if $\mathbf{x}$ is preferred to $\mathbf{x}'$,
- $f(\mathbf{x}, \mathbf{x}') = 0$ if both $\mathbf{x}$ and $\mathbf{x}'$ have the same preference or ranking, or if there is no information about their respective ranking.

Formulation

The learner receives a labeled sample $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{x}'_1, y_1), \cdots (\mathbf{x}_m, \mathbf{x}'_m, y_m)\} \in \mathcal{X} \times \mathcal{X} \times \{-1, 0, +1\}$ with $(\mathbf{x}_1, \mathbf{x}'_1), \cdot, (\mathbf{x}_m, \mathbf{x}'_m)$ drawn i.i.d. according to $\mathcal{D}$ and $y_i = f(\mathbf{x}_i, \mathbf{x}'_i)$ for all $i \in [1, m]$. Given a hypothesis set $\mathcal{H}$ of functions mapping $\mathcal{X}$ to $\mathbb{R}$, the ranking problem consists of selecting a hypothesis $h \in \mathcal{H}$ with small expected pairwise mis-ranking or generalization error $R(h)$ with respect to the target f:

$$R(h) = \Pr_{(x,x') \sim D} \left[\big(f(x, x') \neq 0\big) \wedge \big(f(x, x')(h(x') - h(x)) \leq 0\big) \right]$$

Formulation

The learner receives a labeled sample $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{x}'_1, y_1), \cdots (\mathbf{x}_m, \mathbf{x}'_m, y_m)\} \in \mathcal{X} \times \mathcal{X} \times \{-1, 0, +1\}$ with $(\mathbf{x}_1, \mathbf{x}'_1), \cdot, (\mathbf{x}_m, \mathbf{x}'_m)$ drawn i.i.d. according to $\mathcal{D}$ and $y_i = f(\mathbf{x}_i, \mathbf{x}'_i)$ for all $i \in [1, m]$. Given a hypothesis set $\mathcal{H}$ of functions mapping $\mathcal{X}$ to $\mathbb{R}$, the ranking problem consists of selecting a hypothesis $h \in \mathcal{H}$ with small expected pairwise mis-ranking or generalization error $R(h)$ with respect to the target f:

$$R(h) = \Pr_{(x,x')\sim D}\left[\big(f(x,x') \neq 0\big) \wedge \big(f(x,x')(h(x') - h(x)) \leq 0\big)\right]$$

How to formulate the ranking problem in large margin (SVM) framework?

The optimization problem for ranking with SVMs can be reformulated as:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to: } y_i\left[\mathbf{w}\cdot\left(\mathbf{\Phi}(x_i') - \mathbf{\Phi}(x_i)\right)\right] \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad \forall i \in [1, m]\,.$$

This coincides exactly with the primal optimization problem of SVMs, with a feature mapping $\Psi : \mathcal{X} \times \mathcal{X} \to \mathcal{H}$ defined by $\Psi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}') - \Phi(\mathbf{x})$ for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$, and with a hypothesis set of functions of the form $(\mathbf{x}, \mathbf{x}') \to \mathbf{w} \cdot \Psi(\mathbf{x}, \mathbf{x}')$.

The optimization problem for ranking with SVMs can be reformulated as:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to: } y_i\left[\mathbf{w}\cdot\left(\boldsymbol{\Phi}(x_i') - \boldsymbol{\Phi}(x_i)\right)\right] \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad \forall i \in [1, m].$$

This coincides exactly with the primal optimization problem of SVMs, with a feature mapping $\Psi : \mathcal{X} \times \mathcal{X} \to \mathcal{H}$ defined by $\Psi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}') - \Phi(\mathbf{x})$ for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$, and with a hypothesis set of functions of the form $(\mathbf{x}, \mathbf{x}') \to \mathbf{w} \cdot \Psi(\mathbf{x}, \mathbf{x}')$.

Relation with conventional SVMs?