Multi-class Classification, Maximum Entropy and Structured Classification

Outline

- Multi-class Classification
- Structured Learning

Outline

- Multi-class Classification
- Structured Learning

One-against-the rest

- Assume data in k classes: $\{1, \ldots, k\}$
- Train k binary SVMs:

$$\begin{array}{lll} \text{1st class} & \text{vs.} & (2, \cdots, k)\text{th class} \\ \text{2nd class} & \text{vs.} & (1, 3, \ldots, k)\text{th class} \\ & \vdots & \end{array}$$

- k decision functions

$$\begin{gathered} (\boldsymbol{w}_1)^T \boldsymbol{x} \\ \vdots \\ (\boldsymbol{w}_k)^T \boldsymbol{x} \end{gathered}$$

One-against-the rest (Cont'd)

- Prediction:

$$\arg\max_{j} \ (\boldsymbol{w}_j)^T \boldsymbol{x}$$

- Reason: If $\boldsymbol{x} \in$ 1st class, then we should have

$$(\boldsymbol{w}_1)^T \boldsymbol{x} \geq +1$$
$$(\boldsymbol{w}_2)^T \boldsymbol{x} \leq -1$$
$$\vdots$$
$$(\boldsymbol{w}_k)^T \boldsymbol{x} + b_k \leq -1$$

Multi-class Classification (Cont'd)

- One-against-one: train $k(k-1)/2$ binary SVMs
  $(1,2),(1,3),\ldots,(1,k),(2,3),(2,4),\ldots,(k-1,k)$
- If 4 classes $\Rightarrow$ 6 binary SVMs

$y_i = 1$	$y_i = -1$	Decision functions
class 1	class 2	$f_{12}(\boldsymbol{x}) = (\boldsymbol{w}_{12})^T\boldsymbol{x} + b_{12}$
class 1	class 3	$f_{13}(\boldsymbol{x}) = (\boldsymbol{w}_{13})^T\boldsymbol{x} + b_{13}$
class 1	class 4	$f_{14}(\boldsymbol{x}) = (\boldsymbol{w}_{14})^T\boldsymbol{x} + b_{14}$
class 2	class 3	$f_{23}(\boldsymbol{x}) = (\boldsymbol{w}_{23})^T\boldsymbol{x} + b_{23}$
class 2	class 4	$f_{24}(\boldsymbol{x}) = (\boldsymbol{w}_{24})^T\boldsymbol{x} + b_{24}$
class 3	class 4	$f_{34}(\boldsymbol{x}) = (\boldsymbol{w}_{34})^T\boldsymbol{x} + b_{34}$

- For a testing data, predicting all binary SVMs

Classes		winner
1	2	1
1	3	1
1	4	1
2	3	2
2	4	4
3	4	3

- Select the one with the largest vote

class	1	2	3	4
# votes	3	1	1	1

- May use decision values as well

Solving a Single Problem

- Example (Crammer and Singer, 2002)

$$\min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_k} \quad \frac{1}{2}\sum_{m=1}^{k}\|\boldsymbol{w}_m\|_2^2 + C\sum_{i=1}^{l}\xi(\{\boldsymbol{w}_m\}_{m=1}^{k};\boldsymbol{x}_i,y_i),$$

where

$$\xi(\{\boldsymbol{w}_m\}_{m=1}^{k};\boldsymbol{x},y) \equiv \max_{m\neq y}\max(0,1-(\boldsymbol{w}_y-\boldsymbol{w}_m)^T\boldsymbol{x}).$$

- We hope the decision value of $\boldsymbol{x}_i$ by the model $\boldsymbol{w}_{y_i}$ is larger than others
- Prediction: same as one-against

$$\arg\max_j \; (\boldsymbol{w}_j)^T\boldsymbol{x}$$

Maximum Entropy

- Maximum Entropy: a generalization of logistic regression for multi-class problems
- It is widely applied by NLP applications.
- Conditional probability of label y given data $\boldsymbol{x}$.

$$P(y|\boldsymbol{x}) \equiv \frac{\exp(\boldsymbol{w}_y^T \boldsymbol{x})}{\sum_{m=1}^{k} \exp(\boldsymbol{w}_m^T \boldsymbol{x})},$$

Maximum Entropy (Cont'd)

- We then minimizes regularized negative log-likelihood.

$$\min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_m} \quad \frac{1}{2}\sum_{m=1}^{k} \|\boldsymbol{w}_k\|^2 + C\sum_{i=1}^{l} \xi(\{\boldsymbol{w}_m\}_{m=1}^{k}; \boldsymbol{x}_i, y_i),$$

where

$$\xi(\{\boldsymbol{w}_m\}_{m=1}^{k}; \boldsymbol{x}, y) \equiv -\log P(y|\boldsymbol{x}).$$

Maximum Entropy (Cont'd)

- Is this loss function reasonable?
- If

$$\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i \gg \boldsymbol{w}_m^T \boldsymbol{x}_i, \forall m \neq y_i,$$

  then

$$\xi(\{\boldsymbol{w}_m\}_{m=1}^k; \boldsymbol{x}_i, y_i) \approx 0$$

  That is, no loss
- In contrast, if

$$\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i \ll \boldsymbol{w}_m^T \boldsymbol{x}_i, m \neq y_i,$$

  then $P(y_i|\boldsymbol{x}_i) \ll 1$ and the loss is large.

Features as Functions

- NLP applications often use a function $\boldsymbol{f}(\boldsymbol{x}, y)$ to generate the feature vector

$$P(y|\boldsymbol{x}) \equiv \frac{\exp(\boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{x}, y))}{\sum_{y'} \exp(\boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{x}, y'))}. \tag{1}$$

- The earlier probability model is a special case by

$$\boldsymbol{f}(\boldsymbol{x}_i, y) = \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} y-1 \\ \\ \\ \\ \\ \end{array} \hspace{-9em} \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \boldsymbol{x}_i \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \hspace{6em} \in \mathbf{R}^{nk} \text{ and } \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_k \end{bmatrix}.$$

Outline

- Multi-class Classification
- Structured Learning

Structured Data

- So far we assume that the label y_i is a single value
- In some applications, the label may be a more sophisticated object.
- For example, in part-of-speech (POS) tagging, a training instance is a sentence and a label is a sequence of POS tags of words.
- For l sentences, training instances are

$$(\mathbf{y}_i, \boldsymbol{x}_i) \in Y^{n_i} \times X^{n_i}, \forall i = 1, \ldots, l,$$

where $\boldsymbol{x}_i$ is the ith sentence, $\mathbf{y}_i$ is a sequence of tags,

Structured Data (Cont'd)

- X: set of unique words in the context
- Y: set of candidate tags for each word
- n_i: number of words in the ith sentence.

Conditional Random Fields

- CRF solves

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{l}\xi(\boldsymbol{w};\boldsymbol{x}_i,\mathbf{y}_i),$$

  where

$$\xi(\boldsymbol{w};\boldsymbol{x}_i,\mathbf{y}_i) \equiv -\log P(\mathbf{y}_i|\boldsymbol{x}_i), \text{ and}$$

$$P(\mathbf{y}|\boldsymbol{x}) \equiv \frac{\exp(\boldsymbol{w}^T\boldsymbol{f}(\boldsymbol{x},\mathbf{y}))}{\sum_{\mathbf{y}'}\exp(\boldsymbol{w}^T\boldsymbol{f}(\boldsymbol{x},\mathbf{y}'))}. \qquad (2)$$

- Dynamic programming used to handle exponentially many $\mathbf{y}$

Structured SVM

- Structured SVM solves

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{l}\xi(\boldsymbol{w};\boldsymbol{x}_i,\mathbf{y}_i),$$

  where

$$\begin{aligned}&\xi(\boldsymbol{w};\boldsymbol{x}_i,\mathbf{y}_i)\\ \equiv\ &\max_{\mathbf{y}\neq\mathbf{y}_i}\left(\max\left(0,\Delta(\mathbf{y}_i,\mathbf{y})-\boldsymbol{w}^T(\boldsymbol{f}(\boldsymbol{x}_i,\mathbf{y}_i)-\boldsymbol{f}(\boldsymbol{x}_i,\mathbf{y}))\right)\right)\end{aligned}$$

- $\Delta(\cdot)$ is a distance function

Structured SVM (Cont'd)

- $\Delta(\cdot)$ should satisfy

$$\Delta(\mathbf{y}_i, \mathbf{y}_i) = 0 \text{ and } \Delta(\mathbf{y}_i, \mathbf{y}_j) = \Delta(\mathbf{y}_j, \mathbf{y}_i)$$

.

- If

$$\Delta(\mathbf{y}_i, \mathbf{y}_j) = \begin{cases} 0 & \text{if } \mathbf{y}_i = \mathbf{y}_j \\ 1 & \text{otherwise,} \end{cases}$$

and

$$\mathbf{y}_i \in \{1, \ldots, k\}, \forall i,$$

then structured SVM becomes the multi-class formulation discussed earlier

More Information

- See Sections V and VIII in Yuan et al. (2012) and references therein

References I

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, (2–3):201–233, 2002.

G.-X. Yuan, C.-H. Ho, and C.-J. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/survey-linear.pdf`.