Centre for
Data Analytics

Insight

Machine Learning for Sequence Learning

Learning in an All-Subsequence Space

Severin Gsponer, Georgiana Ifrim, Barry Smyth

January 20, 2016

Outline

- Background
- Linear Classifiers for Sequences
- SEQL Approach
- Contribution
- Future Work

Background for Sequence Learning

Definition of a sequence

A sequence consists of symbols of a given finite alphabet Σ in a given order: $s_0, s_1, \ldots, s_n$

Examples

- Genetic sequence: AGCTGTTCGT , $|\Sigma| = 4,\ \Sigma = \{A, C, G, T\}$
- Protein sequence: KVKTGCKATLR , $|\Sigma| = 20$
- Text: The house is blue , $|\Sigma| = 4$, (# distinct words in corpus)

Sequence Classification

Class	Data points
+1	C70124045**F0*EE*AD**C00E9D64A000C**6689**CCF1C70
+1	7413BAEF01000**6689**51488B7000**F0*EE*AD**00081CA
-1	08F9C81A80**C18B484**000895110B8040000C20C00CCC
-1	CCCFF8CC84C8B5C8B**C18B484**C8B505C8340240481

Find subsequences that can be used to identify the class.

?? CC8CC84C8BC8B458B4CC0F82B505FB4C83B4B0481

Related Work

Bag of Words

- Loss of structural order (e.g., Mary is faster than John)
- Often not accurate enough

Kernel SVM

- Lift into implicit high-dimensional feature space through kernel trick
- Restrict features for scale (e.g., max 5-gram)
- Not easily interpretable (Blackbox)

SEQL (Our Approach)

- Works in explicit high-dimensional feature space
- Unrestricted features (i.e. all-length subsequences)
- Interpretable classifier (Whitebox)

All-Subsequence Feature Space

Sample sequence:	...F09EE1AD ...	
Uni-gram (all):	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	(16 possible)
Bi-gram:	F0, 09, 9E, EE, 1A,...	($16^2 = 256$ possible)
Tri-gram:	F09, 09E, EE1, E1A, 1AD,...	($16^3 = 4096$ possible)
⋮	⋮	
8-gram:	F09EE1AD,...	($16^8 = 4294967296$ possible)

Representation of sequence in explicit vectorspace of all subsequences:

$$
\begin{array}{rl}
 & 0, 1, 2, 3, 4, \ldots, F, 00, 01, 02, 03, \ldots, FF, 000, 0001, \ldots \\
\boldsymbol{x_i} = & (1, 1, 0, 0, 0, \ldots, 1, \;\; 1, \;\; 0, \;\; 0, \;\; 1, \ldots, \;\; 1, \;\; 0, \;\; 0, \ldots)
\end{array}
$$

Linear Sequence Classifier

Given:

Training set of labeled examples:

$$\{x_i, y_i\} \text{ for } i = 1, \ldots, N \quad \text{where } y_i \in \{-1, 1\}$$

$$\boldsymbol{x_i} \in \mathbb{R}^d \quad \text{with } d = \text{number of features}$$

Goal:

Find $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_d)$, $\beta_i \in \mathbb{R}$ by optimizing:

$$\beta^* = \underset{\beta \in \mathbb{R}^d}{\arg\min}\, L(\beta) = \underset{\beta \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{N} \xi(y_i, x_i, \beta) + CR(\beta)$$

Classical gradient descent is computationally infeasible for a large feature space

$$\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \eta_t \boldsymbol{\nabla} \boldsymbol{L}(\boldsymbol{\beta}^{(t-1)})$$

SEQL

Algorithm 1 SEQL worflow

Set $\beta^{(0)} = 0$

while !termination condition **do**

Calculate objective function $L(\beta^{(t)})$

Find feature with maximum gradient value

Find step length η_t by line search

Update $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \eta_t \frac{\partial L}{\partial \beta_{j_t}}(\boldsymbol{\beta}^{(t-1)})$

Add corresponding feature to feature set

end while

Contribution

1. Study influence of problem characteristics on classification performance (simulation)

2. Extend SEQL approach to regression (gradient bound for squared error loss)

3. Real-World Applications

Contribution 1: Simulation Dimensions

- **Alphabet size** $|\Sigma|$
- **Sequence length** L
- **Data set size** N
- Motif length m
- Sparsity of the feature space
- Noise in the motifs

Contribution 1: Analysis

Accuracy

- Classification performance (ACC, AUC, F1, ...)

Speed

- Number of iterations
- Quality of gradient bound (pruning ration)
- Run time

Interpretability

- Number of produced features

Contribution 1: Simulation Framework

Systematic experiments on generated sequences:

Generation of N sequences of length L

$$l_1, l_2, \ldots, l_L$$
$$\text{where} \quad l_i \sim U(\text{Alphabet})$$

Insert **motifs** of length m in positive sequences.
Ratio of positive to negative sequences is 1:10

Contribution 1: Data Generation

1. Random generation of a motif

2. Determine motif insertion position randomly for each sequence

3. Random generation of sequence and insertion of motif at position

Contribution 1: Data Generation

Algorithm 2 Positive sequences generation

```
Generate motif by drawing m symbols from ~ U(Alphabet)
for i < N · 0.1 do
    pos ~ U(L − m)
    for l < (L − m) do
        if l = pos then
            add motif to sequence
        else
            add symbol l ~ U(Alphabet) to sequence
        end if
    end for
    add sequence to data set
end for
```

Contribution 2: Extension to Regression

Value	Data points
+0.2	C70124045C00E9D64A000CCCF1C70
+1.4	7413BAEF0100051488B700000081CA
-3.2	08F9C81A80000895110B8040000C20
-0.1	CCF8CC84C8B5C8BC8B505C834024

Implementation of squared error loss and new gradient bound

$$\xi(y_i, x_i, \beta) = \sum_{i=1}^{N} (y_i - \beta^t x_i)^2$$

With L1 regularization known as LASSO.

Questions Influence of loss function and quality of the bound

Contribution 3: Real World Application

Classification Task

Microsoft Malware Challenge (BIG 2015)
Kaggle Competition in early 2015

Goal Classification of Malware into 9 families

Data $\sim$500GB of hexadecimal sequences

Regression Task

We are still looking for problem domains for sequence regression?

Future Work

Regression applications

Test on real world application.

Rescaling of features

TF-IDF style rescaling of feature instead of binary indicator [1] and analysis of influence for the gradient bound quality.

References

Bibliography

L. Miratrix and R. Ackerman.
Conducting sparse feature selection on arbitrarily long phrases in text corpora with a focus on interpretability.
pages 1--41, 2015.