01: Pattern Recognition and Machine Learning **May 2016**

Lecture 1: Sparse Kernel Machines

Author: Christopher M. Bishop *Scribes: scribe-name1,2,3*

Note: *Reading notes for pattern recognition and machine learning.*

In this chapter we shall look at kernel-based algorithms that have sparse solutions, so that predictions for new inputs depend only on the kernel function evaluated at a **subset** of the training data point.

First the detail of SVM, in which the determination of the model parameters corresponds to a convex optimization problem - so that any local solution is also a global optimum.

SVM is a decision machine and so does not provide posterior probabilities. RVM proposed based on a Bayesian formulation and provides posterior probabilistic outputs - as well as having typical much sparser solutions than the SVM.

1.1 Maximum Margin Classifiers

1.1.1 Overlapping class distributions

1.1.2 Relation to logistic regression

1.1.3 Multiclass SVMs

1.1.4 SVMs for regression

1.1.5 Computational learning theory

1.2 Relevance Vector Machines

1.2.1 RVM for regression

1.2.2 Analysis of sparsity

1.2.3 RVM for classification

We now delve right into the proof.

Lemma 1.1 *This is the first lemma of the lecture.*

Proof: The proof is by induction on .... For fun, we throw in a figure.

Figure 1.1: A Fun Figure

This is the end of the proof, which is marked with a little box. ■

1.2.4 A few items of note

Here is an itemized list:

- this is the first item;
- this is the second item.

Here is an enumerated list:

1. this is the first item;
2. this is the second item.

Here is an exercise:

Exercise: Show that P $\neq$ NP.

Here is how to define things in the proper mathematical style. Let f_k be the $AND - OR$ function, defined by

$$f_k(x_1, x_2, \ldots, x_{2^k}) = \begin{cases} x_1 & \text{if } k = 0; \\ AND(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{if } k \text{ is even;} \\ OR(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{otherwise.} \end{cases}$$

Theorem 1.2 *This is the first theorem.*

Proof: This is the proof of the first theorem. We show how to write pseudo-code now.

Consider a comparison between x and y:

if x or y or both are in S **then**
    answer accordingly
else
    Make the element with the larger score (say x) win the comparison
    if $F(x) + F(y) < \frac{n}{t-1}$ **then**
        $F(x) \leftarrow F(x) + F(y)$
        $F(y) \leftarrow 0$
    else
        $S \leftarrow S \cup \{x\}$
        $r \leftarrow r + 1$
    endif
endif

This concludes the proof. ■

1.3 Next topic

Here is a citation, just for fun [CW87].

References

[CW87] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.