MATHEMATICAL MODELS FOR MACHINE LEARNING AND PATTERN RECOGNITION

D. Bouchaffra[(1)], *F. Ykhlef* [(2)]

Design and Implementation of Intelligent Machines Division (DIIM)
Centre de Développement des Technologies Avancées, Algiers, Algeria

ABSTRACT

In this tutorial, we provide an in depth analysis of some important issues within the field of Machine Learning and Pattern Recognition. We intend to reflect recent developments and provide a comprehensive introduction to some fundamental issues pertaining to the field of machine learning and pattern recognition. We target advanced undergraduates or first year Ph.D. students as well as researchers and practitioners. The mathematical models covered during this tutorial include Machine Learning for Pattern Recognition, Hidden Markov Models and feature space Dimensionality Reduction. MATLAB projects are provided as experiments to the theory covered.

1. INTRODUCTION

The mission of this tutorial is to teach undergraduate and graduate students the different steps necessary to build a pattern recognition system. Practitioners, researchers and students should be capable to operate in different areas when classification or regression is needed based on the information learned in this tutorial. This tutorial is divided into two main parts. The first part is a review of some mathematical concepts and models related to the field of pattern recognition (PR) and machine learning (ML). The second part is devoted to some MATLAB demonstrations of the theory covered in the first part. The materials covered in the theoretical part are: (i) Different modules needed to build a complete PR system, (ii) dimensionality reduction techniques (DR), and (iii) classification of time series data sequences using a hidden Markov model (HMM) formalism.

2. THEORY AND MODELS

2.1. A Pattern Recognition System

The purpose of a PR system is to assign a label to a given input data. The basic scheme of such systems is classification, which attempts to assign each input to one of a given set of classes [1], [2].
A PR system is closely related to the field of ML in the sense that models are capable to learn solely from data. The intelligence of these machines is conveyed by their learning power in a data environment.

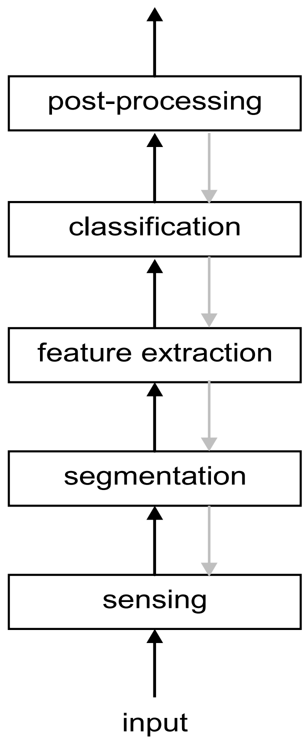

Figure 1: Steps of a PR system [2].

[1] Dr. Bouchaffra is Director of Research and Head of DIIM Division at CDTA.
[2] Ykhlef is a Research Assistant and Lead of the Signal Processing team at DIIM Division.

978-1-4673-5540-7/13/$31.00 ©2013 IEEE

The main steps of a PR system are depicted by Figure 1.

Sensing:

- use of a transducer (camera or microphone),
- PR system depends of the bandwidth, the resolution sensitivity distortion of the transducer.

Segmentation and grouping

- Patterns should be well separated. This step is one of the most difficult, because an object can overlap with another one or can be occluded by another object in the image (or scene). Often algorithms deployed provide several possible segmentations points that can be obtained and only information from the next stages (such as classification and post-processing) can lead to one single correct interpretation.

Feature selection/extraction:

We make a difference between feature selection which consists of reducing the number of features without losing information and feature extraction which requires a transformation (or mapping) from a source feature space to a target low dimensional space.

- There is a need for discriminative features to improve the performance of a PR system,
- Invariant features with respect to translation, rotation and scale are most appealing in many applications such as optical character recognition and face recognition.

Classification:

- Use a feature vector provided by a feature extractor/selector to assign the object to a category among a set of predefined categories.

Post-processing:

- Exploit context input dependent information other than from the target pattern itself to improve performance. For example, linguistic information is often exploited to improve post-processing in speech recognition.

2.2. Dimensionality Reduction

Dimensionality reduction consists of converting data living in a high dimensional space (source) into a lower dimensionality space (target) such that proximity between points in the source space and the target space are respected. The ultimate goal is to minimize the loss of information during this mapping. This is typically done while solving machine learning problems to get uncorrelated and noise free features for a classification task. Features Dimensionality reduction is usually performed in one of these cases [3]:

- Discover hidden correlations/topics,
- Remove redundant and noisy features,
- Interpretation and visualization,
- Easier storage and processing of the data.

Many linear and nonlinear dimensionality reduction techniques have been proposed in the literature. We can mention:

1. Metric Multidimensional Scaling (MDS),
2. Principal Components Analysis (PCA),
3. ISOMAP,
4. Locally Linear Embedding (LLE),
5. Hessian LLE,
6. Laplacian,
7. Diffusion MAP
8. LTSA
9. Semidefinite Embedding (SDE),
10. Laplacian Eigenmaps.
11. Kernel PCA (KPCA),

The DR techniques from 1 through 8 have been covered in the tutorial. We invite the reader to refer to [4] for mathematical details of all these techniques. It is noteworthy that all these techniques have a similar goal, which consists of dealing with small dimensional space; however their performance depends on the topology of the input data.

2.3. Hidden Markov Models

Hidden Markov models [5], [6], [7] have been used in several research areas. The goal in this tutorial is to highlight the importance of the three main problems that are embedded within a hidden Markov model.

Evaluation: Given the observation sequence $O = o_1, o_2, \ldots o_T$ and a model $\lambda = [\pi, A, B]$, determine the probability that this observation sequence was generated by the model λ.

Decoding: Suppose we have an HMM λ as well as a sequence of observation O. Determine the most likely sequence of hidden states $q_1, q_2, \ldots q_T$ that generated the sequence of observation O.

Learning: Suppose we are given a coarse structure of a model (the number of hidden states and the number of observations symbols) but not the probabilities a_{ij} nor b_{jk}. Given a set of training observation sequences, determine these parameters.

$$a_{ij} = \frac{\#< i, j >}{\#< i, + >} \quad (1)$$

$$b_{jk} = \frac{\#< o_k, j >}{\#< o_k, any\, state >}. \quad (2)$$

In order to understand the next sections which represent the main contributions in this paper, let's focus on the evaluation problem. Let $O = (o_1\ o_2\ \ldots o_T)$ be the observation sequence of length T and $q = (q_1 q_2 \ldots q_T)$ be the state sequence where q_1 is the initial state.
The evaluation problem is mathematically expressed as follows: Given a model λ, and the observation sequence O, evaluate the match between λ and the observation sequence O by computing $P(O \mid \lambda)$:

$$P(O|\lambda) = \sum_{all\ q} P(O,q|\lambda) \qquad (3)$$

$$P(O,q|\lambda) = \quad P(O,q|\lambda) \times P(q|\lambda), \qquad (4)$$

and using state conditional independence, we obtain:

$$P(O,q|\lambda) = \prod_{t=1}^{T} P(O_t, q_t|\lambda). \qquad (5)$$

The evaluation problem is based on the state conditional independence of the observation symbols. However, there are several scenarios where the conditional independence assumption doesn't hold. For example, while standard HMM's perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences, they are inadequate for predicting the secondary structure of a protein.

3. PRACTICAL DEMOS

In this session, we provide some MATLAB examples and demonstrations as experiments to the theory covered in section 2. We focus on two main issues:

- Dimensionality reduction,
- Hidden Markov models,

3.1. Dimensionality reduction

We use the Manifold Learning MATLAB Graphical User Interface (MANI GUI) [8] made publically available by the University of California to explore different methods of DR (Figure 2).

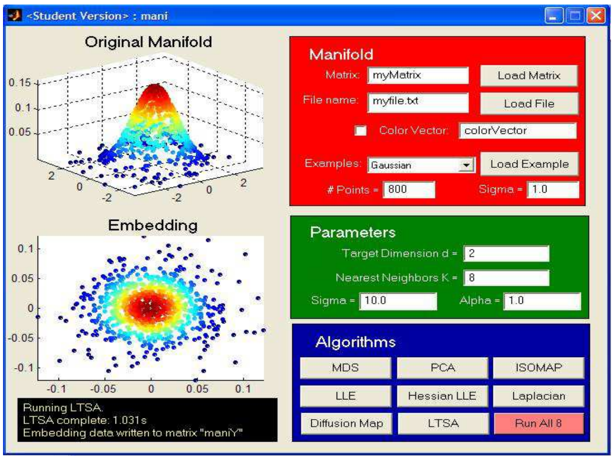

Figure 2: Manifold Learning MATLAB.

The MANI GUI is pretty straightforward to use. It puts together several manifold learning techniques and basic examples into one MATLAB GUI. The incorporated learning techniques are: MDS, PCA, ISOMAP, LLE, Hessian LEE, Laplacian, Diffusion MAP and LTSA.

During this tutorial, three main examples are given. In each example, we try to reduce the dimension of an input data using different algorithms as depicted by Figure 1.

Example 1:

- source dimension: 2D,
- target dimension: 1D,
- type of data: 10×2 matrix which contains a set of random [x,y] coordinates,
- MATLAB functions: MANI GUI (PCA and ISOMAP).

Example 2:

- source dimension: 3D,
- target dimension: 2D,
- type of data: Predefined Gaussian and Swiss Roll data,
- MATLAB functions: MANI GUI (PCA, ISOMAP, LEE, and Diffusion MAP).

Example 3:

- source dimension: ND,
- target dimension: KD,
- type of data: Normally distributed random matrixes of high dimension (N>>K),
- MATLAB functions: (ISOMAP).

We have assisted the students during the tutorial practical session and gave them all necessary materials. It has been shown from these examples that the choice of a suitable DR technique depends on the geometrical aspect of the input data. No technique is inherently superior to any other technique independently of the data distribution.

3.2. Hidden Markov Models

In this application, we focus on the practical use of an HMM through two main examples.

In the first example, we present the classical *weather problem*. In the second example, we show how to classify a test sequence of observations into one of two predefined classes.

Example 1: The weather problem

The hidden states of the model are the three possible types of weather:

- sunny,
- rainy,
- cloudy.

At any given day, we assume that the weather can be only one of these states. Further, in this example, we assume that the person who is observing is locked in the basement, so he/she is not capable to monitor the weather directly. The only evidence the observer has, is whether the guest who checks on her/him every day is carrying an umbrella or not (Figure 3). In HMM terminology, "carrying an umbrella" or "no carrying" represents two visible observations.

The MATLAB toolboxes used in this example are:

- *Statistical MATLAB Toolbox 4.1*: This toolbox is included in MATLAB V7 or higher [9].

Figure 3: Weather Scenario.

- *Kevin Murphy HMM toolbox*: It is an external toolbox which can be downloaded from the University of British Columbia [10].

In this tutorial, we use some functions from these toolboxes to:

- Generate the observations and states of a known model (dhmm_sample),
- Estimate the transition and emission matrices of an unknown model based only on a set of observations and states (dhmm_em and hmmestimate),
- Evaluate the log-likelihood of the sequence (dhmm_logprob).

Example 2: Sequence classification example
The purpose of this example is to classify a test sequence (observation) into one of two classes (Figure 4):

- The first class represents the weather in town 1 (first model),
- The second class represents the weather in town 2 (second model).

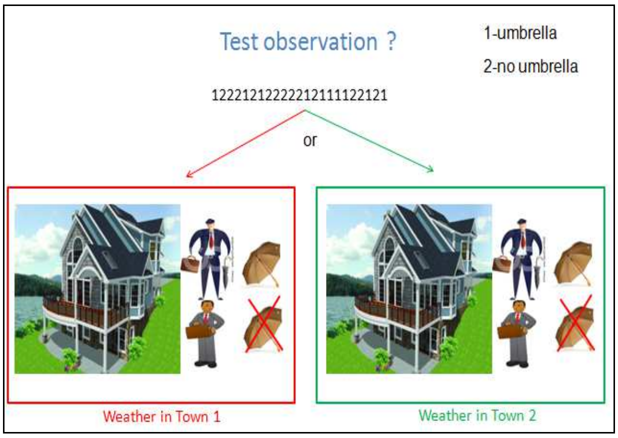

Figure 4: Sequence classification.

To resolve this problem, we have to perform the following three steps:

1. Train up 2 HMMs (models of the weather in town 1 and 2), one per class,
2. Compute the log-likelihood that each model gives to the test sequence,
3. Decide if the i^{th} model is the most likely, then assign class I (i=1 or 2) to the sequence.

The log-likelihood of the sequence is computed using dhmm_logprob function.

4. CONCLUSION

We have presented in this tutorial a complete statistical framework that allows students, practitioners and researchers build a full classification system using MATLAB. Because dimensionality reduction is an important phase in selecting discriminative features, we have demonstrated the importance of choosing the correct dimensionality reduction technique according to the data geometry. We have also emphasized the role played by data context-dependent classifier such as HMMs under a MATLAB platform.

REFERENCES

[1] C. Bishop, *Pattern Recognition and Machine Learning,* Springer, 2006.

[2] Duda, Hart and Stork, *Pattern Classification*, Wiley, 2001.

[3] J. Leskovec, *Dimensionality Reduction: SVD and CUR,* CS 246: Mining Massive Datasets, Stanford University, USA, 2013.

[4] A. Ghodsi, *Dimensionality Reduction: A Short Tutorial,* Department of Statistics and Actuarial Science, University of Waterloo, Canada, 2006.

[5] L.R. Rabiner, A Tutorial on HMM and Selected Applications in Speech Recognition, *In Proc. IEEE*, Vol. 77, No. 2, pp. 257-286, 1989.

[6] D. Bouchaffra and J. Tan, Structural Hidden Markov Models using a Relation of Equivalence: Application to Automotive Designs", in*: Data Mining and Knowledge Discovery Journal*, Vol. 12, No. 1, Springer-V, 2006.

[7] D. Bouchaffra and J. Tan, "Structural Hidden Markov Models: Application to Handwritten Numeral Recognition", *in: Intelligent Data Analysis Journal*, IDA, Vol. 10, No. 1, IOS Press, 2006.

[8] T. Wittman, *MANI fold Learning MATLAB Demo*, Depart-ment of Mathematics, University of California, Los Angeles, USA, 2005.

[9] MATLAB and Statistics Toolbox Release 2007b, The MathWorks, Inc., Natick, Massachusetts, United States.

[10] Kevin Murphy, Hidden Markov Model (HMM) Toolbox for MATLAB, 1998.