UNIVERSITÉ DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématiques appliquées**

Arrêté ministériel : 7 août 2006

Présentée par

Cao Tri DO

Thèse dirigée par **Ahlame DOUZAL-CHOUAKRIA** et codirigée par **Michèle ROMBAUT**et co-encadré par **Sylvain Marié**

préparée au sein du
Laboratoire d'Informatique de Grenoble (LIG)
dans **l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Metric learning for time series analysis

Thèse soutenue publiquement le **date de soutenance**,
devant le jury composé de:

Prénom NOM
Labo de bidule, Rapporteur, Présidente du jury
Prénom NOM
Labo de bidule, Rapporteur
Prénom NOM
Labo de bidule, Examinateur
Prénom NOM
Labo de bidule, Examinateur
Ahlame DOUZAL-CHOUAKRIA
LIG, Directeur de thèse
Michèle ROMBAUT
GIPSA-Lab, Co-Directeur de thèse

UNIVERSITÉ DE GRENOBLE

ÉCOLE DOCTORALE MSTII

Description de complète de l'école doctorale

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble-Alpes

Mention : Informatique et Mathématiques appliquées

Présentée et soutenue par

Cao Tri DO

Metric learning for time series analysis

Thèse dirigée par Ahlame DOUZAL-CHOUAKRIA

préparée au Laboratoire d'Informatique de Grenoble (LIG)

soutenue le date de soutenance

Jury :

Rapporteurs :	Prénom NOM	-	Labo de bidule
	Prénom NOM	-	Labo de bidule
Directeur :	Ahlame DOUZAL-CHOUAKRIA	-	LIG
Co-Directeur :	Michèle ROMBAUT	-	GIPSA-Lab
Encadrant :	Sylvain Marié	-	Schneider Electric
Président :	Prénom NOM	-	Labo de bidule
Examinateur :	Prénom NOM	-	Labo de bidule
	Prénom NOM	-	Labo de bidule

Acknowledgements

I would like to thanks:

- my directors
- my GIPSA collegues
- my AMA collegues
- my Schneider collegues
- my parents

Contents

List of Figures

List of Tables

Table of Acronyms

LIG	*Laboratoire d'Informatique de Grenoble*
AMA	*Apprentissage, Méthode et Algorithme*
GIPSA-Lab	*Grenoble Images Parole Signal Automatique Laboratoire*
AGPiG	*Architecture, Géométrie, Perception, Images, Gestes*
A4S	*Analytic for Solutions*
k-**NN**	*k-nearest neighbors*
SVM	*Support Vector Machines*
SVR	*Support Vector Regression*
d_E	*Euclidean distance*
corr	*Pearson correlation*
cort	*Temporal correlation*
dtw	*Dynamic Time Warping*
IoT	*Internet of Things*

Introduction

Motivation

- Qu'est-ce qu'une série temporelle ? (réponse d'un système dynamique complexe (= pas de modèle du système)
- Motiver l'intérêt des séries temporelles dans les applications aujourd'hui: données de plus en plus présentes dans de nombreux domaines divers et variés
- Les séries temporelles sont impliquées dans des problèmes de classification, régression et clustering
- Pourquoi sont-elles challenging ? (délais, dynamique)
- On fait face à la fois, à un problème de small et big data

Problem statement (with words)

- Dans de nombreux algorithmes de classification ou de régression (kNN, SVM), la comparaison des individus (séries temporelles) reposent sur une notion de distance entre individus (séries temporelles).
- Contrairement aux données statiques, les données temporelles peuvent être comparés sur la base de plusieurs modalités (valeurs, forme, distance entre spectre, etc.) et à différentes échelles. La « métrique idéale », càd, celle qui permettra de résoudre au mieux le problème de classification/régression peut donc impliquer plusieurs modalités.
- Objectif de notre travail : Apprendre une métrique adéquate tenant compte de plusieurs modalités et de plusieurs échelles en vue d'une classification/régression kNN

PhD contributions

- Définition d'un nouvel espace de représentation: la représentation par paires
- Apprentissage d'une métrique multimodale et multi-échelle en vue d'une classification kNN à vaste marge de séries temporelles monovariées.
- Extension/Transposition du problème d'apprentissage de métrique (Metric Learning) dans l'espace des paires

- Comparaison de la méthode proposée avec des métriques classiques sur un vaste jeu de données (30 bases) de la littérature dans le cadre de la classification univariée de séries temporelles
- Extension du framework d'apprentissage de métrique au problème de régression de séries temporelles univariés
- Extension du framework d'apprentissage de métrique au problème de classification/régression de séries temporelles multivariés.

Organisation du manuscrit

Présenter les différents chapitres

Notations

$\mathbf{x}_i$	a time series
y_i	a label (discrete or continous)
$\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$	a set of $n \in \mathbb{N}$ labeled time series
d_E	Euclidean distance
L_q	Minkovski q-norm
$\|\|\mathbf{x}\|\|_q$	q-norm of the vector $\mathbf{x}$
d_A	Value-based distance
$corr$	Pearson correlation
$cort$	Temporal correlation
d_F	Euclidean distance between the Fourier spectrum
D	Distance
$\mathbf{x}_{ij}$	a pair of time series $\mathbf{x}_i$ and $\mathbf{x}_j$
y_{ij}	the pairwise label of $\mathbf{x}_{ij}$
t	time stamp/index with $t = 1, ..., T$
T	length of the time series (supposed fixed)
f	frequential index
F	length of the Fourier transform
ξ	Relaxation term
p	number of metric measure considered in the metric learning process
r	order of the temporal correlation
k	number of nearest neighbors
$K(\mathbf{x}_i, \mathbf{x}_j)$	Kernel function between $\mathbf{x}_i$ and $\mathbf{x}_j$
$\phi(\mathbf{x}_i)$	embedding function from the original space to the Hilbert space
C	Hyper-parameter of the SVM (trade-off)
α	
λ	

Part I

Work positionning

The first part of the manuscript aims to position the work context. Our objective is the comparison and the classification or regression of time series. The first chapter considers time series as static vector data and presents classic machine learning algorithms used to classify them. We note that most of these methods relies on the comparison of objects (time series in our case) through a distance measure. In the second chapter, to cope with the characteristics of time series (amplitude, behavior, frequential spectrum, etc.), we recall some basic metrics used to compare time series. We show that time series may be compared by several modalities and at different granularities. We finally cast that learning an adequate distance based on several modalities and several granularities is a key challenge nowadays to well classify time series using classic machine learning algorithms.

CHAPTER 1

Machine Learning for static data: state of the art

Sommaire

In this chapter, we first present what a time series is. Then, we consider time series as static data by vectorizing them to apply machine learning algorithms used for classification and regression. We detail some of these learning algorithms. Finally, we review the protocol used to learn the best fitting of the hyper-parameters of these algorithms and how we evaluate and compare the different algorithms performances.

1.1 Definition of a time series

Let $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iT})$ be a univariate time series of length T. We call time series, a collection of numerical observations made sequentially in the time. It is characterized by a finite number of realized observations made at discrete instants of time $t = 1, ..., T$. Moreover, each observation x_{it} is bounded (i.e. the infinity is not a valid value: $x_{it} \neq \pm\infty$). The time series $\mathbf{x}_i$ is said to be univariate if the collection of observations x_{it} comes from the observations of one variable (i.e. it has been measured by one sensor, the temperature for example). When the observations are made from Q variables (several sensors such as the temperature, the pressure, etc.), the time series is said multivariate and is denoted $\mathbf{x}_i =$

$(\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,Q}) = (x_{i1,1}, ..., x_{iT,1}, x_{i1,2}, ..., x_{iT,2}, ..., x_{i1,Q}, ..., x_{iT,Q})$. We consider in the following for simplification purpose univariate time series.

Fig. 1.1 illustrates a model for time series proposed by Chatfield in [**Chatfield2004a**]. It states that a time series can be decomposed into 3 components: a trend, a cycle (periodic component) and a residual (irregular variations).

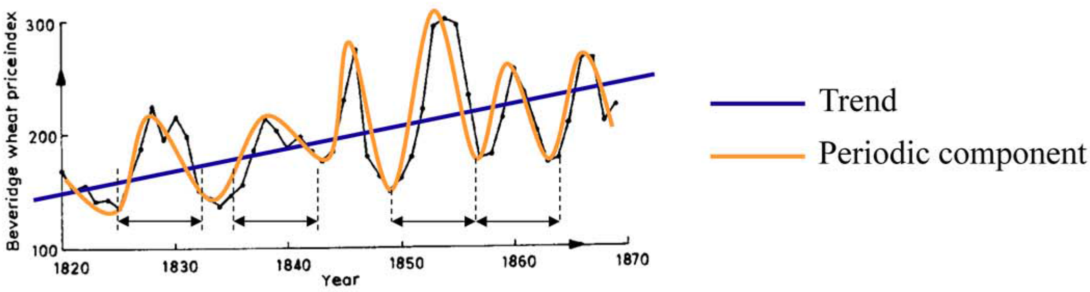

Figure 1.1: The Beveridge wheat price index is the average in nearly 50 places in various countries measured in successive years from 1500 to 1869. [1]

According to Chatfield, most time series exhibit a variation at a fixed period of time (seasonality) such as for example the seasonal variation of temperature. Beyond this cycle, there exist either or both a long term change in the mean (trend) that can be linear, quadratic, and a periodic (cyclic) component. By definition, a signal is periodic of period R if $x_{t+R} = x_t$ for all time instants t. In practice, this condition is not valid for real time series.

Time series can be found in various emerging applications such as sensor networks, smart buildings, social media networks or Internet of Things (IoT) [**Najmeddine2012**; **Nguyen2012**; **Yin2008**]. They are involved in classification, regression and clustering problems. Due to their temporal and structured nature, time series constitute complex data to be analyzed by classic machine learning approaches. One common technics consists in considering time series as a static vector data and then to use classic machine learning algorithms.

1.2 Machine learning algorithms

We are going to detail now a few number of machine learning algorithms used classically to solve classification or regression problems (k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM)). Other algorithms are popular nowadays (Deep neural network, Decision tree, Relevance vector machine, etc.) but we will focus on k-NN and SVM because these learning algorithms are based on the comparison of objects (time series in our case) through a distance measure, notion that we will develop in the next chapter and that will be at the base of our proposition.

[1] This time series can be downloaded from `http://www.york.ac.uk/depts/maths/data/ts/ts04.dat`

1.2.1 Classification, regression and clustering

Let $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be a set of n time series $\mathbf{x}_i$ and y_i their corresponding labels. The aim of Machine learning is to learn a relation (model) f between the time series $\mathbf{x}_i$ and their labels y_i based on examples. This relationship can include static relationships, correlations, dynamic relationship, etc. [**Dreyfus2006**].

The idea of Machine learning (also refer as Pattern learning or Pattern recognition) is to imitate with algorithms executed on computers, the ability of living beings to learn from examples. For example, to teach a child how to read letters, we show him during the learning phase labeled examples of letters ('A', 'B', 'C', etc.) written in different styles and fonts. We don't give him a complete and analytic description of the topology of the characters but labeled examples. After the learning phase (testing phase), we want the child to be able to recognize and to label correctly the letters that have been seen during the training, and also to generalize to new instances. In the same spirit, after the training phase based on labeled examples, the model f has to be able to generalize on the testing phase, i.e. to give a correct prediction for new instances that haven't been seen during the training.

When y_i are class labels (i.e. class 'A', 'B', 'C'), learning the model f is a classification problem; when y_i is a continuous value (i.e. the energy consumption in a building), learning f is a regression problem. Both problems corresponds to supervised learning as $\mathbf{x}_i$ and y_i are known during the training phase [biblio]. When a part of the labels y_i are known and an other part of y_i is unknown during training, learning f is a semi-supervised problem [biblio]. Finally, when the labels y_i are unknown, learning f refers to a clustering problem (unsupervised learning) [biblio]. Our work will focus on supervised learning (classification, regression).

1.2.2 k-Nearest Neighbors (k-NN)

- Donner l'intuition
- Formaliser le problème en tant que problème d'optimisation
- Présenter les extensions (kNN pondéré), extension à la régression
- Soulever le fait que la résolution du problème fait intervenir une notion de distance entre les individus (time series)
- Donner les arguments qui permettent de dire pourquoi utiliser un 1-NN permet de faire une évaluation de métriques (Ding)

A simple approach to classify objects is to consider that "close" objects have a great probability to belong to the same class. Given a test sample (in our case, a time series) $\mathbf{x}_i$, one can decide that this object belong to the same class of its nearest neighbor in the training set. More generally, we can consider the k nearest neighbors of an unknown object $\mathbf{x}_i$, and

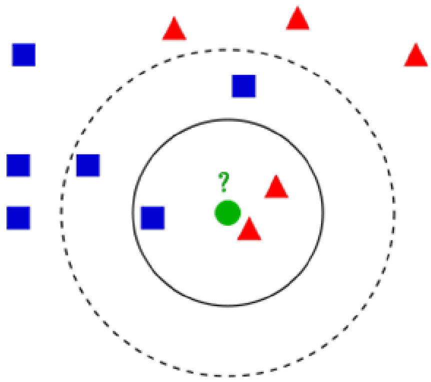

Figure 1.2: Example of k-NN classification. The test sample (green circle) is classified either to the first class (blue squares) or to the second class (red triangles). If $k = 3$ (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

we assign its label y_i to the majority class of its k nearest neighbors (vote). This algorithm is refer as the k nearest neighbors algorithm (k-NN) [**Dreyfus2006**].

The key point in the k-NN algorithm is that the notion of "closeness" between objects is based on the computation of a distance measure. Let D be a distance measure. Usally, for static data, standard metrics are the Euclidean Distance, the Minkowski Distance or the Mahalanobis Distance. Solving the 1-NN classification problem is equivalent to solve the optimization problem:
For a new instance $\mathbf{x}_j$, $\forall i \in \{1...n\}$,

$$y_j = y_i \tag{1.1}$$

where $i = \underset{i \in \{1...n\},}{argmin} D(\mathbf{x}_i, \mathbf{x}_j)$.

The k-NN algorithm can be extended to estimate continous labels (regression problems). The procedure is similar. The label y_j is defined as :

$$y_j = \sum_{i=1}^{k} y_i \tag{1.2}$$

where i corresponds to the index of the k-nearest neighbors. **[biblio]**

Variant kNN (kNN weighed, fuzzy kNN)

Pros of the kNN

Cons of the kNN

DUDA The computational complexity of the nearest-neighbor algorithm - both in space (storage of prototypes) and time (search) - has received a great deal of analysis. Suppose we have n labelled training samples in d dimensions, and seek to find the closest to a test point x (k = 1). In the most naive approach we inspect each stored point in turn, calculate its Euclidean distance to x, retaining the identity only of the current closest one. Each distance calculation is O(d), and thus this search is 0(dn2). An alternative but straightforward parallel implementation is shown in Fig. 4.17, which is 0(1) in time and O(n) in space.

1.2.3 Support Vector Machine (SVM)

- Présenter le principe général des SVM en commençant par l'intuition.
- Forme primale: donner la formulation primale du problème d'optimisation
- Forme duale: montrer commencer passer de la forme primal à la forme dual.
- A partir de la forme duale, passer au kernel
- Finir par la complexité du SVM en apprentissage ($o(N^3p)$) et en test (limités aux nombres de support vectors) et les interprétations des supports vectors.
- Expliquer la modification du problème initiale avec une régularisation L1 sur le terme de régularisation et nous permet d'avoir une solution sparse (donner 1 cas où la solution sparse est meilleur).
- Expliquer que dans le cadre de données non-balancées, il faut ajouter des termes pour rebalancer l'équation du SVM

1.2.4 Other classification algorithms

- S'intéresser au Deep neural network (à la mode en ce moment)
- RVM, Decision Tree,
- Ne pas trop développer
- Dans notre cas, on ne s'intéressera pas à ce type d'algorithmes (type deep learning) car il ne repose pas sur une notion de distance et les features qui sont trouvés ne sont pas interprétables

1.3 Learning protocol

1.3.1 Learning framework

- Learning framework (train, validation, test): la plupart des algorithmes requiert l'optimisation d'un hyper-paramètre. Le jeu de train permet d'apprendre les meilleurs hyper-paramètres.

- Cross-validation (pourquoi on fait de la cross-validation, comment on la fait (Faire un schéma))

1.3.2 Model evaluation

- Classification Error et test de significativité
- Regression Error

1.3.3 Data pre-processing

- Normalization
- PCA

1.4 Limits of classical technics

- The concept of order of temporal data in not take into account (dynamic, frequence, etc.)
- Introduction to next chapter: all of these algorithms (kNN, SVM, etc.) are based on a notion of distance or similarity between objects to compare/classify. Let consider now the object time series and let recall the concept of distance between time series

Chapter 2

Time series basic metrics

Sommaire

Chapeau introductif

- Rappel : notion de similaire : dans le cadre de la classification, on a un comportement « similaire » pour une même classe. La notion de « similaire » est lié à une notion de distance ou (dis)similarité.
- Donner les hypothèses de travail :
  - Considérons la série temporelle comme étant un objet ordonné.
  - les séries temporelles sont de même taille
  - les séries temporelles ont la même période d'échantillonnage
  - les séries temporelles peuvent être comparés sur l'ensemble des valeurs, sur une partie des valeurs, sur un ensemble de fenêtre (fréquences, etc.)
- On va définir dans la suite la notion de métrique, d'alignement, de localité pour des séries temporelles.
- Mettre un graphique (dit GRAPHIQUE GENERAL) qui prend 5 séries temporelles que l'on va utiliser pour la suite : proche en valeur, proche en forme, proche en fréquence, proche en valeur avec un délai, proche en forme avec un délai

2.1 Properties of a distance measure

- Rappeler les propriétés d'une mesure de distance (positivité, symétrique, distinguabilité, inégalité triangulaire)
- Donner les différences entre métriques, distance, dissimilarités, similarités, pseudo-métrique, etc.
- Dans la suite du travail, on va tout assimiler au mot métrique pour une meilleure simplicité

2.2 Basic metrics for time series

2.2.1 Value-based metrics

- Distance classiquement utilisée dans la littérature
- Distance de Minkowski (norm Lp)
- Distance de Mahalanobis (norm pondéré)
- D_E qui est une forme particulière de Minkowski
- Prendre le GRAPHIQUE GENERAL et faire le calcul des distances entre les courbes et montrer que pour 2 courbes qui ont des "amplitudes proches", on obtient une valeur de distance faible.

2.2.2 Behavior-based metrics

- Intuition : expliquer ce que signifie "2 séries temporelles sont proches en forme".
- Dans la littérature classique, on trouve la corrélation de Pearson
- Récemment, Douzal & al. propose une généralisation: cort
- Transformer la cort en mesure de dissimilarité
- Prendre le GRAPHIQUE GENERAL et faire le calcul des distances entre les courbes et montrer que pour 2 courbes qui ont des "formes proches", on obtient une valeur de distance faible.

2.2.3 Frequential-based metrics

- Dans le cadre du traitement de signal, les gens utilisent des représentations fréquentielles (Fourier, etc.)
- Rappeler la transformée de Fourier (TF) + spectre (module de la TF)
- On peut définir une distance dans la représentation de Fourier.
- Prendre le GRAPHIQUE GENERAL et faire le calcul des distances entre les courbes et montrer que pour 2 courbes qui ont des "spectres proches", on obtient une valeur de distance faible.

2.2.4 Other metrics

- Il existe dans la littérature de nombreuses autres métriques pour les séries temporelles (laisser la porte ouverte).
- Certaines métriques sont utilisées dans le domaine temporelle
- D'autres métriques sont utilisés dans d'autres représentations (Wavelet, etc.)
- Certaines combinent la représentation temporelles et fréquentielles (Représentation spectrogramme en temps-fréquence)
- Se baser sur l'article "TSclust : An R Package for Time Series Clustering".
- Fermer le cadre : dans la suite de notre travail, on ne va pas les utiliser mais elles pourront être intégrées dans le framework qui suivra au chapitre suivant

2.3 Kernels for time series

(à compléter)

2.4 Time series alignment

- Les données réelles peuvent présenter des délais, des changements de dynamique de l'échelle de temps : extension, compression (dans la limite du raisonnable).
- Il existe des techniques qui permettent de ré-aligner les séries temporelles comme la DTW
- Définir la notion d'alignement

- Présenter la DTW (+ algorithme)
- Présenter les variantes de la DTW
- Dans la suite du travail, on suppose que les séries temporelles sont ré-alignées.
- Prendre le GRAPHIQUE GENERAL et faire le calcul des distances entre les courbes et montrer que pour 2 courbes qui ont des "valeurs proches" mais décalés, on obtient une valeur de distance faible. (prendre DTW standard avec une fonction de coût D_E par exemple)

2.5 Multi-scale aspect

- Dans le cadre de la classification, on peut avoir des données où l'information qui permet de discriminer une classe d'une autre n'est pas globale mais est localisé sur une partie du signal
- Limite des métriques basiques présentées précédemment (valeur, forme, fréquence) considère la comparaison sur l'intégralité du signal
- On propose la définition de métriques locales. Pour cela, on va découper notre signal. Il existe plusieurs manières de réaliser ce découpage. On va utiliser la dichotomie proposée par Douzal & al.

Chapter 3

Time series advanced metrics

Sommaire

Chapeau introductif

- Objectif : Trouver une distance, combinaison des distances basiques qui donne une bonne classification k-NN sur une base de données.
- Pourquoi une distance combinée? Dans le cadre de données réelles, plusieurs modalités peuvent être impliquées (forme, valeur, fréquence), de manière globale ou locale.
- Dans le cadre des données réelles, plusieurs composantes/modalités peuvent être impliqués (forme, valeur, fréquence). = attribut (feature) en traitement du signal. Hypothèse : valeur sur une série complète, sur un intervalle ou sur une fenêtre (dans le cadre des métriques à base fréquentielle).

3.1 Combined metrics for time series

- Certains travaux dans la littérature propose des combinaisons : linéaire, exponentielle, sigmoïde.
- Limites:
  - Implique que 2 modalités et au niveau global. Pour intégrer d'autres modalités et à d'autres échelles, il faut changer la formule et ajouter de nouveaux hyper-paramètres à optimiser → l'apprentissage de ces paramètres est plus long.
  - La combinaison est définie a priori
  - La combinaison est indépendante de la tâche d'analyse.
  - Pour répondre à ces problèmes, certains auteurs proposent d'apprendre une métrique en vue de la tâche d'analyse considérée (classification, régression, clustering).

3.2 Metric Learning: state of the art

- Placer le contexte : travaux réalisés dans le cadre de la classification de données statiques.
- Présenter l'intuition du Metric Learning sur la base des travaux de Weinberger.
- Donner la terminologie (target, imposter, push, pull)
- Objectif : push des imposters et pull des targets
- Formalisation du problème (optimisation)
- Limites:
  - On apprend les poids d'une distance de Mahalanobis
  - L'apprentissage ne prend pas en compte l'aspect multi-modal dans les données

Conclusion of Part I

This first part of the manuscrit present a state of the art of classic machine learning framework and algorithms to make the classification or the regression of time series. We note that the considered algorithms (k-NN, SVM) are based on the comparison of time series through distance measure. Considering time series as static data lead to compare time series only the comparison of their value, on the same instant between the two considered time series. To take into account over characteristics of time series, other metrics (cort, d_F, etc.) and other methods (DTW, dichomotie) have been proposed to cope with these characteristics. Learning an adequate metrics is a key challenge to well classify time series. Inspired by the metric learning work, we propose in the following a framework to learn a multi-modal and multi-scale metric for a nearest neighbors classification.

Part II

Metric learning for time series from multiple modalities and multiple scales

The first part has enlightened the importance of combining several modalities and several scales to compare time series in order to make a better classification. We propose, in this part, a framework to learn this metric. For that, the idea is to define a new space representation, the pairwise space, where a vector is a pair of time series which is described by the basic metrics. Then, we formalize the problem of learning the metric as an optimization problem and show its equivalence by solving an adequate SVM problem. In the first chapter, we present this pairwise representation and gives interpretation in this new space. In the second chapter, we formalize the optimization problem and its adapted SVM equivalence. In the third chapter, we present the details of the algorithm. In the final chapter, we experiment our proposed approach on datasets, discuss and interpret the results.

Chapter 4

Projection in the pairwise space

Sommaire

Chapeau introductif

- Le calcul d'une métrique implique toujours 2 individus. On va proposer un changement d'espace, un nouvel espace : la représentation par paire.
- Le cadre : on suppose que l'on a p métriques.

4.1 Pairwise embedding

- Changement de l'espace
- Normalisation de l'espace des paires
- Label des pairwise

4.2 Interpretation of the pairwise space

- Proximity to the origin (les individus sont identiques)
- Proximity of 2 pairwise points in the pairwise space
- Norm in the pairwise space
- Representation of combined metric in the pairwise space

4.3 Pros & Cons

- perte de la classe initiale des individus. L'information qui nous reste est : les 2 individus sont de la même classe ou sont de classes différentes.

Chapter 5

M^2TML: formalization

Sommaire

Chapeau introductif

- Rappeler : quel problème on résout : pull des targets et push des individus de classes différentes
- Formaliser le problème général avec D

5.1 LP optimization problem

- Formaliser le problème sous forme d'un problème d'optimisation sous contraintes

5.2 QP optimization problem

- Passer de la forme LP (forme primale) et par transformation, arriver à la forme duale
- Montrer les similitudes avec la résolution SVM
- Montrer que l'on peut kerneliser la méthode

5.3 SVM approximation

- Faire remarquer que le problème LP ressemble à un problème SVM
- Faire la démonstration de l'équivalence (ou mettre la démonstration en annexe).

- Expliquer les différences entre la résolution LP/QP et la résolution SVM. (ajout de sur-contraintes dans le problème SVM)
- Expliquer pourquoi on va préférer le cadre SVM. Expliquer mathématiquement et avec des interprétations géométriques.
- Cadre connu
- Utilisation de librairie standard de Machine Learning
- Extension directe à l'apprentissage de métrique non linéaire grâce au kernel trick

Chapter 6

M^2TML: implementation

Sommaire

Chapeau introductif :

- Quel problème on résout?
- Donner les étapes principales de résolution (sous forme de puces). Cela doit rester général, clair et concis.
- Développer dans chaque section les puces énumérés précédemment.

6.1 Projection in the pairwise space

- Projection
- Log normalization

6.2 M-NN M-diff strategy

- Expliquer les différentes stratégies (k-NN VS All / M-NN VS M-diff / k-NN VS Imposters)
- Expliquer pourquoi on va choisir une stratégie M-NN VS M-diff

6.3 Radius normalization

- Expliquer le problème de la non-homogénéité des radius.
- Expliquer comment on résout ce problème par une normalisation des radius de chaque voisinage.

6.4 Solving the SVM problem

- Expliquer l'apprentissage avec le SVM.
- Utilisation de la version L1 du SVM pour avoir une solution sparse.

6.5 Definition of the dissimilarity measure

- Produit scalaire
- Papier PR : norme pondérée x fonction exponentielle
- Version Sylvain : norme x fonction exponentielle?

6.6 Extension to regression problem

(To do)

6.7 Extension to multivariate problem

(To do)

Chapter 7

M^2TML: Experiments

Sommaire

Chapeau introductif

- Application sur des bases de séries temporelles univariés de la littérature (Keogh)
- Données Schneider? ou Expliquer les problématiques de Schneider

7.1 Dataset presentation

7.2 Experimental protocol

7.3 Results

7.4 Discussion

Conclusion of Part II

Conclusion and perspectives

- Bilan des apports de la thèse
- Perspectives
  - Multi-pass learning
  - Kernel pour la résolution du problème QP
  - Utilisation de la distance apprise dans d'autres algorithmes de machine learning (Arbre de décision) pour obtenir une interprétabilité?
  - Utilisation d'autres distances (wavelets, etc.)
  - Apprentissage locale de la métrique

Appendix A

Detailed presentation of the datasets

Appendix B

Solver library

Appendix C

SVM librairy

Résumé — Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit. Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis.

Mots clés : Série temporelle, Apprentissage de métrique, k-NN, SVM, classification, régression.

Abstract — Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit. Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis.

Keywords: Time series, Metric Learning, k-NN, SVM, classification, regression.

Schneider Electric
Université Grenoble Alpes, LIG
Université Grenoble Alpes, GIPSA-Lab