

Two Sides of Outliers: Optimal Subsequence Bijection and Classification of Imbalanced Datasets

Ph.D. Dissertation Defense

Suzan Köknar-Tezel

Temple University
Computer and Information Sciences

September 10, 2010



Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

The Story

This is the story of outliers

- When outliers corrupt
 - Classifying noisy time series
- When outliers are important
 - The minority class in imbalanced data sets

Outline

1 Introduction and Motivation

- The Story
- **Time Series**
- Imbalanced Data Sets
- Thesis

2 Optimal Subsequence Bijection

- Existing Distance Measures
- OSB
- Experimental Results

3 Ghost Points

- Distance Spaces
- Ghost Points
- Experimental Methodology
- Experimental Results

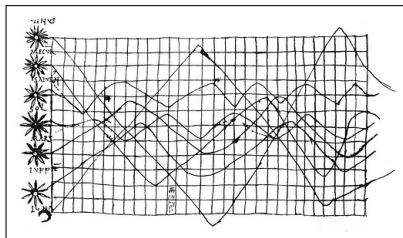
4 Summary and Future Work

5 For Further Reading

Time Series

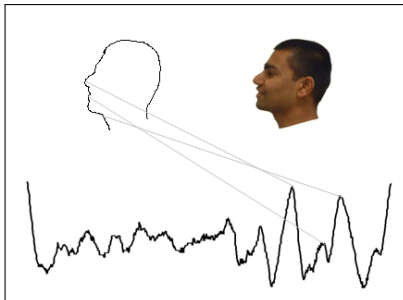
When Outliers Corrupt

- Sequences of real numbers are commonly used in all research fields
- Historically called *Time Series*
 - Even if the natural ordering is imposed from dimension other than time

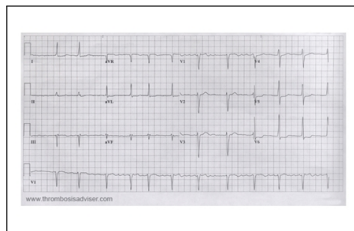


Earliest known time series
plot of planetary orbits
from 10th century monastery
[Funkhouser, 1936]

Example Time Series



[Ratanamahatana and Keogh, 2004]



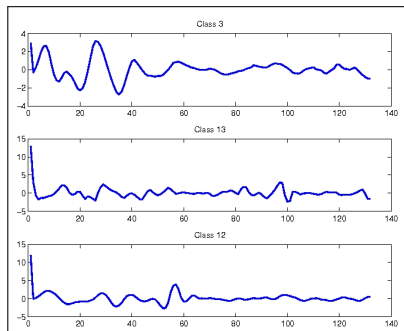
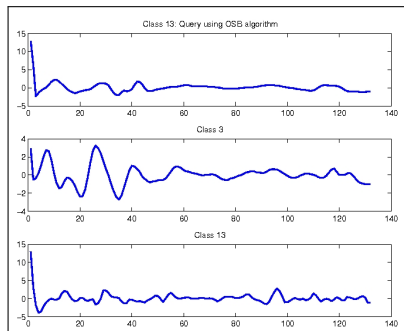
www.thrombosisadvisor.com



finance.yahoo.com

The Problem

Given a data set of known time series, and one unknown time series, can we predict the class label of the unknown time series?



Solutions I

Many data mining algorithms have similarity (distance) measurements of sequences at their core

- Classification [Rafiei, 1999]
- Clustering [Aach and Church, 2001]
- Motif discovery [Chiu et al., 2003]
- Anomaly detection [Keogh et al., 2004, Salvador et al., 2004]

Solutions II

Time series distance measures

- Euclidean Distance
- Dynamic Time Warping [Velichko and Zagoruyko, 1970, Sakoe and Chiba, 1971]
- Longest Common Subsequence [Das et al., 1997, Vlachos et al. 2003]
- Optimal Subsequence Bijection 2007 [Latecki et al., 2007]
- Edit Distance with Real Penalty [Chen and Ng, 2004]
- Time Warp Edit Distance [Marteau, 2009]

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - **Imbalanced Data Sets**
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

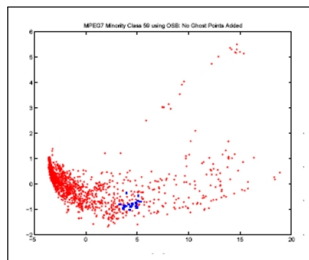
Imbalanced Data Sets

When Outliers are Important

Many real-world data sets are highly imbalanced

- Oil spill detection 896/41 [Kubat et al., 1998]
- Mammography 10,923/260 [Woods et al., 1993]
- Credit card fraud $\approx 400,000/\approx 100,000$ [Chan and Stolfo, 1998]

- The rare event, aka
 - The minority class
 - Positive examples
- The common event, aka
 - The majority class
 - Negative examples



Problems

- Most traditional learning systems are designed to work on balanced data
 - They are biased towards the majority class
 - They focus on improving overall performance
 - They usually perform poorly on the minority class
- There may be uneven costs associated with false negatives and false positives
 - E.g., in cancer diagnosis, a false negative may cost a patient much more than a false positive
 - And often, these costs are difficult to quantify

Solutions

- Undersample the majority class
 - Lose potentially useful data
- Resample the minority class
 - May lead to overfitting since examples are duplicated
- Add synthetic points
 - Until now, this could be done only in feature space

Outline

1 Introduction and Motivation

- The Story
- Time Series
- Imbalanced Data Sets
- **Thesis**

2 Optimal Subsequence Bijection

- Existing Distance Measures
- OSB
- Experimental Results

3 Ghost Points

- Distance Spaces
- Ghost Points
- Experimental Methodology
- Experimental Results

4 Summary and Future Work

5 For Further Reading

Thesis

- Contribution 1: Optimal Subsequence Bijection [Köknar-Tezel and Latecki, 2010b]
 - A sequence matching method
 - Directly optimizes the sum of distances of corresponding elements
 - Allows penalized skipping of outlier elements
 - Defines a bijection on the remaining subsequences
- Contribution 2: Ghost Points [Köknar-Tezel and Latecki, 2010a]
 - A synthetic point that can be added in distance spaces that are metric or non-metric
 - Using geometric analysis, we show that the distances are preserved between the ghost points and the other points in the distance space

Outline

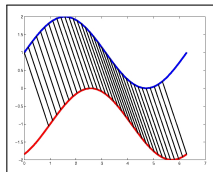
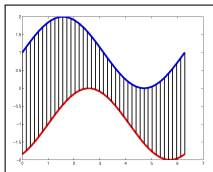
- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 **Optimal Subsequence Bijection**
 - Existing Distance Measures
 - **OSB**
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 **Optimal Subsequence Bijection**
 - **Existing Distance Measures**
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Euclidean Distance

- Given two sequences of real numbers of equal length n , the simplest comparison is to treat them as vectors in \mathbb{R}^n , and compute their squared Euclidean distance (ED)
- This assumes that both sequences are well aligned but this is often not satisfied



- Euclidean distance is very sensitive to distortions in the data
 - Outliers
 - Time phase

Euclidean Distance Equation

Given two sequences a and b of the same length n ,

$$a = (a_1, \dots, a_n) \text{ and } b = (b_1, \dots, b_n)$$

the Euclidean distance between a and b is

$$ED(a, b) = \sum_i^m |a_i - b_i|^2$$

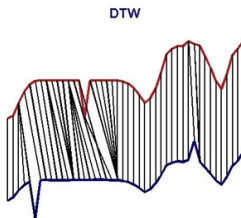
Dynamic Time Warping I

DTW:

- The most well-known elastic measure
- First used for aligning spoken words
- It allows two sequences to be stretched or compressed to optimize local alignments
 - The distance is then computed as the sum of the distances of the corresponding elements
 - Dynamic programming is used to find corresponding elements so that this distance is minimal

Dynamic Time Warping II

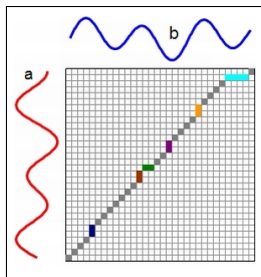
- It is particularly sensitive to outliers, since it is not able to skip any elements of the sequences
 - Each element of the query sequence must correspond to some element of the target sequence and vice versa
 - Thus, the optimal correspondence computed by DTW is a relation on the set of indices of both sequences, i.e., a one-to-many and many-to-one mapping
- The fact that outlier elements must participate in the correspondence optimized by DTW often leads to an incorrect correspondence of other sequence elements



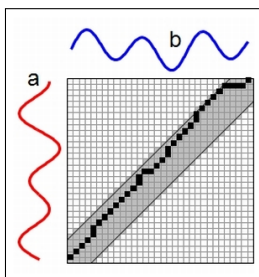
Dynamic Time Warping III

DTW with warping windows

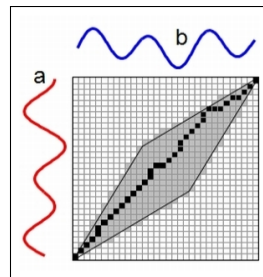
- A global constraint that
 - Prevents pathological warping
 - Slightly speeds up calculation



Dynamic Time Warping



Sakoe-Chiba Band



Itakura Parallelogram

Images taken from [Keogh and Ratanamahatana, 2005]

DTW vs OSB

- The main difference is that OSB can skip outlier elements of the query and target sequences while DTW requires that every element of both sequences participate in the correspondence
 - This makes the performance of OSB robust in the presence of outliers
- OSB defines a bijection on the remaining subsequences, i.e. a one-to-one correspondence of the remaining elements

Dynamic Time Warping Equation

Given two sequences a and b of different lengths m and n ,

$$a = (a_1, \dots, a_m) \text{ and } b = (b_1, \dots, b_n).$$

- A nonnegative, local dissimilarity function d must be defined for every pair of elements a_i and b_j
 - For univariate time series, usually the L_1 -norm is used
 - $d(a_i, b_j) = |a_i - b_j|$

$$DTW(a, b) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \vee n = 0 \\ |a_m - b_n| + \min\{DTW(a_{1:i}, b_{1:j}) & \text{otherwise} \\ \quad |i = m - 1 \vee m, j = n - 1 \vee n, \\ \quad i + j < m + n\} \end{cases}$$

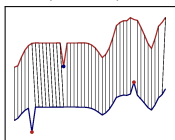
Longest Common Subsequence I

LCSS:

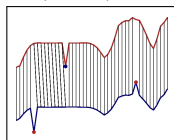
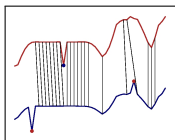
- Used to deal with the alignment and outlier problems
- LCSS determines the longest common subsequence
 - LCSS finds subsequences of the query and target that best correspond to each other
- The subsequence
 - Does not need to consist of consecutive points
 - The order of points is not rearranged
 - Some points can remain unmatched
- The distance is based on the ratio between the length of longest common subsequence and the length of the whole sequence

Longest Common Subsequence II

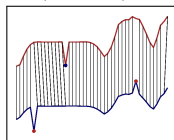
- When LCSS is applied to sequences of numeric values, one needs to set a threshold ϵ that determines when values of corresponding points are treated as equal [Vlachos et al. 2003]
 - The performance of LCSS depends heavily on the correct setting of ϵ

LCSS ($\epsilon = 0.05$)

OSB (C = 0.1)

LCSS ($\epsilon = 0.01$)

OSB (C = 0.002)



LCSS vs OSB

- LCSS optimizes over the length of the longest common subsequence, while OSB directly optimizes the sum of distances of corresponding elements
- OSB includes a penalty (*jumpcost*) for skipping elements in either the query or target sequence
 - The penalty for skipping consecutive elements of a sequence is proportional to the number of elements skipped
 - Thus skipping one outlier costs less than skipping a consecutive subsequence of several elements
- LCSS has no direct penalty for skipping elements which often leads to accidental matches
- In OSB, the equality of two elements is dynamic and it depends on other elements in their neighborhoods in both sequences
- In LCSS, the threshold ϵ is static

Longest Common Subsequence Equation

Given two sequences a and b of different lengths m and n ,

$$a = (a_1, \dots, a_m) \text{ and } b = (b_1, \dots, b_n).$$

- There is no “distance” between two elements
- Instead, if two elements match (according to some threshold ϵ), then the subsequence length is increased by 1

$$LCSS(a, b) = \min \begin{cases} 0 & \text{if } m = 0 \vee n = 0 \\ 1 + LCSS(a_{1:m-1}, b_{1:n-1}) & \text{if } |a_m - b_n| < \epsilon \\ \max\{LCSS(a_{1:m-1}, b), LCSS(a, b_{1:n-1})\} & \text{otherwise} \end{cases}$$

Edit Distance with Real Penalty

- The edit distance was originally used as a distance measure between strings,
- The edit distance between two strings is the smallest number of primitive operations (insertions, deletions, and substitutions) needed to transform one string into the other
- It has been adapted to work with sequences of real numbers
- In ERP, the only edit operation supported is deletion of an element from a sequence, but it is treated as an insertion of a null element into the other sequence
 - The null element is indicated by Λ
- The L_1 distance is used for calculating the distance between two elements, using a constant value for Λ
- Note that ERP can be viewed as a variant of
 - The L_1 -norm except that it handles local time shifting
 - DTW except that it is a metric

Edit Distance with Real Penalty Equation I

Given two sequences a and b of different lengths m and n ,

$$a = (a_1, \dots, a_m) \text{ and } b = (b_1, \dots, b_n)$$

the distance between two elements is defined as

$$d_{erp}(a_i, b_j) = \begin{cases} |a_i - b_j| & \text{if } a_i, b_j \text{ match} \\ |a_i - g| & \text{if } b_j = \Lambda \\ |b_j - g| & \text{if } a_i = \Lambda \end{cases}$$

where g is a constant gap penalty

Edit Distance with Real Penalty Equation II

- The authors of ERP use $g = 0$ and give two justifications:
 - When $g = 0$, the distance between sequences a and b corresponds to the difference between the area under the curve of a and the area under the curve of b
 - Then $\sum_i a_i = \sum_j a'_j$ where a is the original sequence and a' is the transformed sequence

The ERP between two sequences is

$$ERP(a, b) = \begin{cases} \sum_1^n |b_i - g| & \text{if } m = 0 \\ \sum_1^m |a_i - g| & \text{if } n = 0 \\ \min\{ERP(a_{1:m-1}, b_{1:n-1}) + d_{erp}(a_m, b_n), & \text{otherwise} \\ ERP(a_{1:m-1}, b) + d_{erp}(a_m, \Lambda), \\ ERP(a, b_{1:n-1}) + d_{erp}(\Lambda, b_n)\} \end{cases}$$

Time Warp Edit Distance

- TWED also combines L^p -norms with edit distance like ERP
- But also uses the time stamps of the sequences when calculating the distance between elements
 - This controls the elasticity of the measure
- TWED uses the difference in the time stamps to linearly penalize the matching elements
 - This favors matching elements that have close time stamps

Time Warp Edit Distance Equation I

Given two sequences a and b of different lengths m and n ,

$$a = (a_1, \dots, a_m) \text{ and } b = (b_1, \dots, b_n)$$

the distance between two elements is defined as

$$d_{twed}(a_i, b_j) = \begin{cases} d_{match} & \text{if } a_i, b_j \text{ match} \\ d_{dele} & \text{if } b_j = \Lambda \\ d_{delb} & \text{if } a_i = \Lambda \end{cases}$$

Time Warp Edit Distance Equation II

with

$$d_{match} = dist(a_i, b_j) + dist(a_{i-1}, b_{j-1}) + \nu \cdot (|t_{a_i} - t_{b_j}| + |t_{a_{i-1}} - t_{b_{j-1}}|)$$

$$d_{dela} = dist(a_i, a_{i-1}) + \nu \cdot (t_{a_i} - t_{a_{i-1}}) + \lambda$$

$$d_{delb} = dist(b_j, b_{j-1}) + \nu \cdot (t_{b_j} - t_{b_{j-1}}) + \lambda$$

where

- $dist(a_i, b_j)$ is any L^p -norm
- $\lambda \geq 0$ is a constant penalty for deletion
- $\nu \geq 0$ is a constant that characterizes the stiffness of the elasticity
- $|t_{a_i} - t_{b_j}|$ is the time-stamp difference of elements a_i and b_j respectively

Time Warp Edit Distance Equation III

The TWED between two sequences is

$$TWED(a, b) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \vee n = 0 \\ \min \left\{ \begin{array}{l} TWED(a_{1:m-1}, b_{1:n-1}) + \\ \quad d_{twed}(a_m, b_n), \\ TWED(a_{1:m-1}, b) + d_{twed}(a_m, \Lambda), \\ TWED(a, b_{1:n-1}) + d_{twed}(\Lambda, b_n) \end{array} \right\} & \text{otherwise} \end{cases}$$

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 **Optimal Subsequence Bijection**
 - Existing Distance Measures
 - **OSB**
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Optimal Subsequence Bijection

- The *Optimal Subsequence Bijection (OSB)* works for the elastic matching of two sequences of different lengths m and n :

$$a = (a_1, \dots, a_m) \text{ and } b = (b_1, \dots, b_n).$$

- The goal of OSB is to find subsequences a' of a and b' of b such that a' best matches b'
- Skipping (not matching) some elements of a and b is necessary because both sequences may contain some outlier elements
- However, skipping too many elements of either sequence increases the chance of accidental matches
- To prevent this from happening, we introduce a penalty for skipping which we call *jump cost* and denote it with \mathbf{C} .

Definition of OSB I

- To formally define OSB, we need to first augment the sequences a and b by first and last elements

$$\bar{a} = (a_0, a_1, \dots, a_m, a_{m+1}) \text{ and } \bar{b} = (b_0, b_1, \dots, b_n, b_{n+1}).$$

- The subsequences using \bar{a} and \bar{b} will be denoted \bar{a}' and \bar{b}'
 - \bar{a}' will always contain the elements a_0 and a_{m+1}
 - \bar{b}' will always contain b_0 and b_{n+1}
- These added elements do not contribute to the computed distance between the optimal subsequences \bar{a}' and \bar{b}' .

Definition of OSB II

- We assume that the distance function d used to compute the dissimilarity value between elements is given
- We do not have any restrictions on the distance function d other than non-negativity, i.e., $d(a_i, b_j) \geq 0$
- Usually, for sequences of real numbers, $d(a_i, b_j) = (a_i - b_j)^2$
 - This is the case for all our experiments
- We define

$$d_{osb}(a_i, b_j) = \begin{cases} (a_i - b_j)^2 & \text{if } 1 \leq i \leq m \wedge 1 \leq j \leq n \\ 0 & \text{if } (i = 0 \wedge j = 0) \vee \\ & (i = m + 1 \wedge j = n + 1) \\ \infty & \text{otherwise} \end{cases}$$

Definition of OSB III

- We want to select a subsequence a' of the query sequence a by skipping some outlier elements of a
 - 1 So that each element of a' matches to some element of b
 - 2 In an order preserving manner
 - 3 With possibly skipping some outliers in b as well
- The optimal correspondence is obtained by optimizing the balance between the dissimilarity of a' to its image subsequence of b and the penalties of skipping elements of a and of b

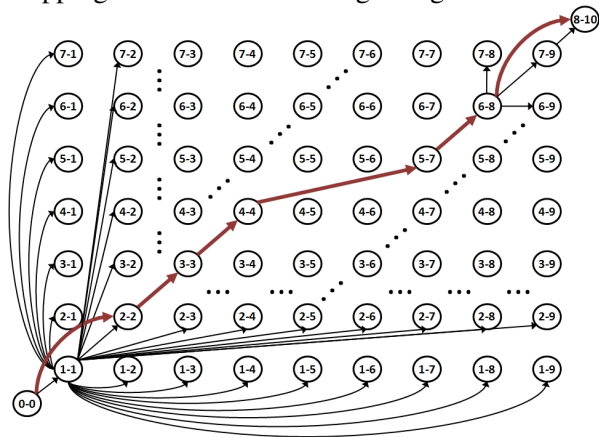
Definition of OSB IV

- The optimal correspondence can be found with a shortest path algorithm on a DAG (directed acyclic graph)
- The nodes of the DAG are all index pairs $(i, j) \in \{0 \dots m + 1\} \times \{0 \dots n + 1\}$
- The edge cost w is defined as

$$w((i, j)(k, l)) = \begin{cases} ((k - i - 1) + (l - j - 1)) \cdot C + d(a_k, b_l) & \text{if } i < k \wedge j < l \\ \infty & \text{otherwise} \end{cases}$$

Definition of OSB V

The purpose of the added nodes $(0, 0)$ and $(m + 1, n + 1)$ is to have distinct *source* and *destination* vertices for the shortest path algorithm and to allow the skipping of elements at the beginning and the end of a and b



Definition of OSB VI

- The output of OSB yields a *correspondence* defined as a monotonic injection

$$f : \{i_0, \dots, i_{m'}\} \rightarrow \{0, 1 \dots n + 1\}$$

such that

- $(i_0, \dots, i_{m'}) \subseteq (0, 1 \dots m + 1)$ is a subsequence with
 - $i_0 = 0$
 - $i_{m'} = m + 1$
 - $f(i_0) = f(0) = 0$
 - $f(i_{m'}) = f(m + 1) = n + 1$
- The sets of indices $\{i_0, \dots, i_{m'}\}$ and $\{f(i_0), \dots, f(i_{m'})\}$ define subsequences \bar{a}' of \bar{a} and \bar{b}' of \bar{b} , such that f restricted to these sequences is a bijection
 - The phrase “subsequence bijection” in *OSB*

Definition of OSB VII

- Our goal is to find a subsequence bijection f that minimizes the function

$$\frac{1}{m'} \sum_{k=0}^{m'} w((i_k, f(i_k)), (i_{k+1}, f(i_{k+1}))).$$

- We need to find subsequences
 - $\bar{a}' = (a_{i_0}, \dots, a_{i_{m'}})$ of \bar{a}
 - $\bar{b}' = (b_{f(i_0)}, \dots, b_{f(i_{m'})})$ of \bar{b}
- With a minimal total weight for w
 - The word “optimal” in *OSB*

Definition of OSB VIII

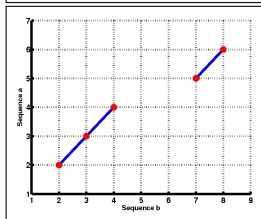
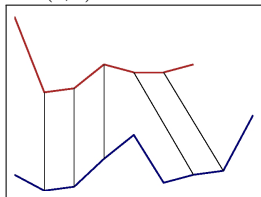
The OSB distance between two sequences is

$$OSB(\bar{a}, \bar{b}) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \vee n = 0 \\ d_{osb}(a_{m+1}, b_{n+1}) + \min\{OSB(a_{0:i}, b_{0:j}) + \\ \quad (|m - i| + |n - j|) \cdot C \mid i = 0 : m + 1, \\ \quad j = 0 : n + 1, i + j < m + n + 2\} & \text{otherwise} \end{cases}$$

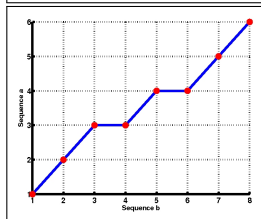
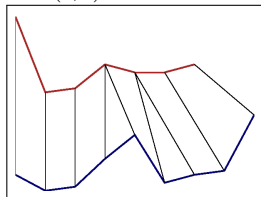
A Simple Example

Given two sequences $a = (20, 1, 2, 8, 6, 6, 8)$ and $b = (5, 1, 2, 9, 15, 3, 5, 6, 20)$ with the jump cost $C = 1$

$$OSB(a, b) = 8$$



$$DTW(a, b) = 14.28$$



Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 **Optimal Subsequence Bijection**
 - Existing Distance Measures
 - OSB
 - **Experimental Results**
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Experimental Results

- The UCR data sets (20 data sets) [Keogh et al., 2006]
 - We had best accuracy on 8 data sets
 - Uniquely best on 5
- The MPEG-7 Core Experiment CE-Shape-1 data set [Latecki et al., 2000]
 - Full sequences
 - We had 96.3% accuracy for 1NN
 - We had 72.4% accuracy for bulls-eye
 - Partial sequence matching
 - We had 100% accuracy for 1NN
 - We had 79% accuracy for bulls-eye

UCR Results

DATASET	Number of Classes	Size of Training Set	Size of Testing Set	Time Series Length	ED	DTW WW	DTW	LCSS	ERP	OTWED	OSB
Synthetic Control	6	300	300	60	0.120	0.017	0.007*	0.047	0.036	0.023	0.020
Gun-Point	2	50	150	150	0.087	0.087	0.093	0.013+	0.040	0.013+	0.020
CBF	3	30	900	128	0.148	0.004	0.003+	0.009	0.003+	0.007	0.004
Face (all)	14	560	1690	131	0.286	0.192	0.192	0.201	0.202	0.189*	0.190
OSU Leaf	6	200	242	427	0.483	0.384	0.409	0.202 *	0.397	0.248	0.409
Swedish Leaf	15	500	625	128	0.213	0.157	0.210	0.117	0.120	0.102	0.085*
50 Words	50	450	455	270	0.369	0.242	0.310	0.213	0.281	0.187*	0.257
Trace	4	100	100	275	0.240	0.010	0.000*	0.020	0.170	0.050	0.030
Two Patterns	4	1000	4000	128	0.090	0.002	0.000+	0.000+	0.000+	0.001	0.000+
Wafer	2	1000	6174	152	0.005	0.005	0.020	0.000*	0.009	0.004	0.001
Face (four)	4	24	88	350	0.216	0.114	0.170	0.068	0.102	0.034*	0.045
Lightning2	2	60	61	637	0.246	0.131+	0.131+	0.180	0.148	0.213	0.131+
Lightning7	7	70	73	319	0.425	0.288	0.274	0.452	0.301	0.247	0.192*
ECG	2	100	100	96	0.120	0.120	0.230	0.100+	0.130	0.100+	0.100+
Adiac	37	390	391	176	0.389	0.391	0.396	0.425	0.378	0.376	0.358*
Yoga	2	300	3000	426	0.170	0.155	0.164	0.137	0.147	0.130*	0.142
Fish	7	175	175	463	0.267	0.233	0.267	0.091	0.120	0.051*	0.103
Beef	5	30	30	470	0.467	0.467	0.500	0.533	0.500	0.533	0.433*
Coffee	2	28	28	286	0.250	0.179+	0.179+	0.214	0.250	0.214	0.286
OliveOil	4	30	30	570	0.133	0.167	0.133	0.800	0.167	0.167	0.100*
Total Number of Best Scores per Method					0	2	6	5	2	7	8
Total Number of UNIQUELY Best Scores per Method					0	0	2	2	0	5	5

Table: + indicates a best score; * indicates a uniquely best score.

MPEG-7 Results (Full Sequences)

	OSB C = 0.03	LCSS $\epsilon = 0.45$	DTW $r = 3$
1NN	0.963 ✓	0.955	0.912
5NN	0.872 ✓	0.847	0.780
10NN	0.779 ✓	0.752	0.678
20NN	0.651 ✓	0.627	0.557
Bulls-eye	0.724 ✓	0.719	0.624

Table: The retrieval results on the MPEG-7 data set for various distance measures. Bolded, checked results indicate best scores.

MPEG-7 Results (Partial Sequences)

	Full-length Targets			Targets using Corresp. Window			
	OSB	LCSS	DTW	OSB	LCSS	DTW	ED
1NN	1.00 ✓	0.40	0.10	1.00 ✓	0.80	0.50	0.70
5NN	0.86 ✓	0.26	0.06	0.86 ✓	0.70	0.38	0.66
10NN	0.82 ✓	0.28	0.05	0.82 ✓	0.59	0.33	0.46
20NN	0.69 ✓	0.23	0.04	0.69 ✓	0.47	0.28	0.31
Bulls-eye	0.79 ✓	0.33	0.09	0.79 ✓	0.57	0.37	0.40

Table: The retrieval results on the MPEG-7 data set for ten partial query sequences. Bolded, checked results indicate best scores.

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points**
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points**
 - Distance Spaces**
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Metric Distance Spaces

Definition

A *metric* on a set X is a distance function $\rho : X \times X \rightarrow \mathbb{R}$, such that the following axioms hold:

- 1 $\rho(x, y) \geq 0$ (non-negativity)
- 2 $\rho(x, y) = \rho(y, x)$ (symmetry)
- 3 $\rho(x, y) = 0 \Leftrightarrow x = y$ (positive definiteness)
- 4 $\rho(x, y) + \rho(y, z) \geq \rho(x, z)$ (triangle inequality)

for any $x, y, z \in X$.

Definition

A *metric space* is an ordered pair (X, ρ) , where X is a set of points, and ρ is metric on X , that is, a distance function $\rho : X \times X \rightarrow \mathbb{R}$.

Embeddings

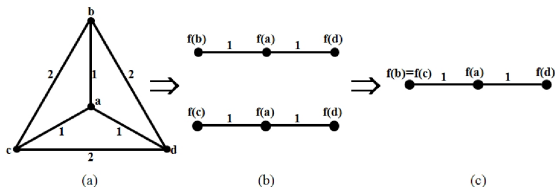
- When non-Euclidean distance measures are used, embeddings to low dimensional Euclidean spaces are often utilized
- However, embedding implies distance distortion
- In addition, not every four point metric space can be isometrically embedded into a Euclidean space \mathbb{R}^k
 - E.g., see [Matousek, 2002]

Definition

Let Y and Z be two metric spaces. We say that a mapping f of the space Y into Z is an *isometric embedding* if $dist_Z(f(y_1), f(y_2)) = dist_Y(y_1, y_2)$.

4-Point Embedding Example [Georgiou and Hatami, 2008]

- Given the metric space (X, ρ) defined in the figure below and the mapping $f : X = \{a, b, c, d\} \rightarrow \mathbb{R}^k$ for some k where f preserves the distances
- The triangle inequality holds for a, b, d
 - In fact $\rho(b, d) = \rho(b, a) + \rho(a, d)$ and because of the equality, the mapped points $f(b), f(a)$, and $f(d)$ are collinear in the space \mathbb{R}^k
- Same holds for elements a, c, d
- But then $f(b) = f(c)$ contradicting the fact that the original distance between b and c is 2



Non-metric Distance Spaces I

- Many applications have non-metric distances at their core, such as distances between
 - Images
 - Shapes
 - Text documents
 - Time series
- The data are often represented as a matrix of pairwise comparisons
- This matrix represents the distance space of the data and is often non-metric

Non-metric Distance Spaces II

- Many well-established machine learning methods require the data to be metric
- Non-metric distance spaces are forced to be metric by embedding them into Euclidean spaces
- The distortion of the data that occurs with this embedding is assumed to be noise
 - But little is known about the real information loss
- Working directly with non-metric distance spaces may better represent the real distances between objects [Jacobs et al., 2000, Laub and Müller, 2004]

Non-metric Distance Spaces III

Definition

For our purposes, a *distance space* is an ordered pair (X, ρ) , where X is a set of points and $\rho : X \times X \rightarrow \mathbb{R}$ is a distance function that satisfies


- 1 $\rho(x, y) \geq 0$ (non-negativity)
- 2 $\rho(x, y) = \rho(y, x)$ (symmetry)
- 3 $x = y \Rightarrow \rho(x, y) = 0$

We would like ρ to be as close as possible to a metric, but this is not always possible

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 **Ghost Points**
 - Distance Spaces
 - **Ghost Points**
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

3-Point Metric Embedding

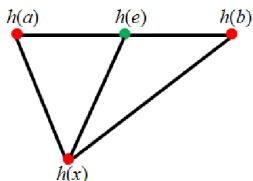
- Although not every four point metric space can be embedded into a Euclidean space, every three point metric space can be isometrically embedded into the plane \mathbb{R}^2
- Let (Δ, ρ) , where $\Delta = \{x, a, b\} \subseteq X$, be a metric space with three distinct points. Then it is easy to map Δ to the vertices of a triangle on the plane.
- For example, we can construct an isometric embedding $h : \Delta \rightarrow \mathbb{R}^2$ by setting $h(a) = (0, 0)$ and $h(b) = (\rho(a, b), 0)$.
- Then $h(x)$ is uniquely defined as a point with nonnegative coordinates such that its Euclidean distance to $h(a)$ is $\rho(x, a)$ and its Euclidean distance to $h(b)$ is $\rho(x, b)$.
- This construction does not require that (X, ρ) be a metric space, but it does require that the three point space (Δ, ρ) be a metric space. 

Definition of Ghost Points I

Given any two points a, b in a distance space X , we define a *ghost point* e induced by a and b using the construction $e = \mu(a, b) = h^{-1}(\frac{1}{2}(h(a) + h(b)))$. For every $x \in X$, the distance from x to e , $\rho(x, \mu(a, b))$, is computed as follows:

Case 1: If the three point subspace $\Delta = \{x, a, b\}$ is a metric, then

$$\rho(x, \mu(a, b))^2 = \frac{1}{2}\rho(x, a)^2 + \frac{1}{2}\rho(x, b)^2 - \frac{1}{4}\rho(a, b)^2 \quad (1)$$

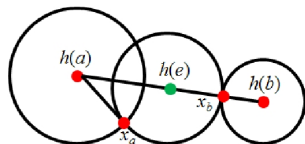
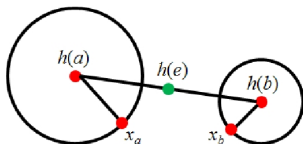


Definition of Ghost Points II

Cases 2 and 3 in this definition apply when Δ is not a metric space

Case 2: If $\rho(a, b) > \rho(x, a) + \rho(x, b)$, then

$$\rho(x, \mu(a, b)) = \frac{1}{2}\rho(a, b) - \rho(x, b) \quad (2)$$



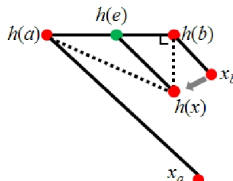
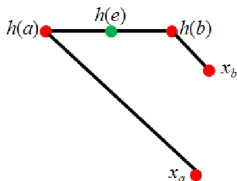
Definition of Ghost Points III

Case 3a: If $\rho(x, a) > \rho(x, b) + \rho(a, b)$, then

$$\rho(x, \mu(a, b))^2 = \rho(x, b)^2 + \frac{1}{4}\rho(a, b)^2 \quad (3)$$

Case 3b: If $\rho(x, b) > \rho(x, a) + \rho(a, b)$, then

$$\rho(x, \mu(a, b))^2 = \rho(x, a)^2 + \frac{1}{4}\rho(a, b)^2 \quad (4)$$



The Augmented Distance Space

- Ghost points, as defined, are guaranteed to be nonnegative and symmetric by their construction
 - Hence the space augmented by ghost points remains a distance space
- If the space X is finite, i.e., $X = \{x_1, \dots, x_n\}$, then the distance function $\rho : X \times X \rightarrow \mathbb{R}_{\geq 0}$ is represented by a square matrix $M_\rho(X)$
 - Each row of the square distance matrix $M_\rho(X)$ is the distance of one data point x to all data points in the data set
 - I.e., for all $y \in X$, $M_\rho(x, y) = \rho(x, y)$.
- The matrix for $X \cup \{\mu(a, b)\}$ is obtained by simply adding one row and one column to $M_\rho(X)$, with each entry computed using the equations in the definition
- **Thus, the proposed approach can be applied to metric and non-metric distance spaces**

SMOTE I

- Synthetic Minority Oversampling Technique (SMOTE) [Chawla et al., 2002] has shown that it can improve overall classification accuracy and also improve the learning of the rare event
 - The synthetic points are generated from existing minority class examples
 - It takes the difference between the corresponding feature values of a minority class example x and one of its nearest neighbors in the minority class
 - Multiplies each feature difference by a random number between 0 and 1
 - Adds these amounts to the feature vector of x .

SMOTE II

- But SMOTE works only in feature space
- Feature space - n -dimensional space where n is the number of features of each example
- Variations of SMOTE
 - SMOTEBoost [Chawla et al., 2003]
 - SMOTE with Different Costs [Akbana et al., 2004]
 - Borderline-smote [Han et al., 2005]

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 **Ghost Points**
 - Distance Spaces
 - Ghost Points
 - **Experimental Methodology**
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

Evaluating Performance on Imbalanced Data Sets

- The Mammography data set [Woods et al., 1993] has 10,923 examples of non-cancerous tumors and 260 examples of cancerous tumors
- A trivial classifier will be 97.68% accurate
- But with a misclassification rate of 100% on the minority examples
- When the performance on the minority class is as important or more important than overall accuracy, other performance measures must be used

Confusion Matrix

- Metrics borrowed from the information retrieval community

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table: Confusion Matrix

Performance Metrics

- *Accuracy*
 - Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- *Precision* measures the exactness of a classifier
 - Higher precision means less false positives
 - Precision = $\frac{TP}{TP+FP}$
- *Recall* measures the completeness or sensitivity of a classifier
 - Higher recall means less false negatives
 - Recall = $\frac{TP}{TP+FN}$
- *F_β-Measure* is the weighted harmonic mean of precision and recall
 - F₁-Measure weights precision and recall equally
 - F₂-Measure weights recall twice as heavily as precision
 - F_β-Measure = $(1 + \beta^2) \frac{\text{Recall} \times \text{Precision}}{\beta^2 \times \text{Precision} + \text{Recall}}$

Data Sets and Distance Measures Used

- Data sets:
 - The UCR data sets [Keogh et al., 2006]
 - We use 17 of the data sets
 - Three already have two classes with a minority class
 - For the other fourteen, we create minority classes using one-against-all then average the results
 - The MPEG-7 Core Experiment CE-Shape-1 data set [Latecki et al., 2000]
 - Seventy classes with twenty examples per class
 - Use one-against-all for all classes then average the results
- Distance functions
 - Optimal Subsequence Bijection (OSB) [Köknar-Tezel and Latecki, 2010b]
 - Dynamic Time Warping (DTW) [Velichko and Zagoruyko, 1970, Sakoe and Chiba, 1971]

Methodology: Training Set

- 1 Given a training set consisting of m time series, create the $m \times m$ distance matrix by calculating the OSB or DTW distance between each pair of examples.
- 2 For each minority class example x , add k -many ghost points by inserting one ghost point between x and each of its k nn. This gives us a total of p new points.
- 3 Calculate the distance from the p ghost points to every other point in the training set; we now have an $(m + p) \times (m + p)$ matrix.
- 4 Convert both the original and augmented OSB or DTW score matrix to affinity matrices using the approach in [Yang et al., 2008].
- 5 Use these affinity matrices as the *user-defined* or *precomputed* kernels for the SVM to get two models: one that includes ghost points and one that does not.
- 6 Run SVM to train.

Methodology: Testing Set

- 1 Given a testing set consisting of n time series, and a training set consisting of m time series, create the $n \times m$ OSB or DTW distance score matrix.
- 2 Calculate the distance from each test data point to each of the p ghost points; we now have an $n \times (m + p)$ distance matrix.
- 3 Convert both the original and augmented OSB or DTW score matrix to an affinity matrix as for training set.
- 4 Use these affinity matrices as the *user-defined* or *precomputed* kernels for the SVM as in step 1e above.
- 5 Run SVM to test.

Parameters I

- Param 1 & 2: Converting distance matrix to affinity matrix [Yang et al., 2008]
 - The affinity between a pair of points

$$k(x_i, x_j) = \exp\left(\frac{-d(x_i, x_j)^2}{\sigma_{ij}}\right) \quad (5)$$

where

- $\sigma_{ij} = A \cdot \text{mean}\{\text{knn } d(x_i), \text{knn } d(x_j)\}$
 - $\text{mean}\{\text{knn } d(x_i), \text{knn } d(x_j)\}$ is the the mean distance of the K -nearest neighbors of points x_i, x_j
 - A is an extra scaling parameter
- Param 3: SVM
 - The cost parameter C

Parameters II

- Param 4: The number of ghost points per minority example
 - The final results can be sensitive to this
 - Two good heuristics but neither always give the best results
 - 1 Balance the classes
 - 2 Add one ghost point per minority example
 - How to choose the optimal number of ghost points is an open question
- For all UCR experiments we used $A = 0.5$, $K = 5$, and $C = 0.5$
- For all MPEG-7 experiments we used $A = 0.36$, $K = 25$, and $C = 0.5$

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 **Ghost Points**
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - **Experimental Results**
- 4 Summary and Future Work
- 5 For Further Reading

UCR Data Sets and OSB

Shaded results indicate best performers; the darker the shade, the larger the difference between the results with and without ghost points

Data Set	#GP Added Per Minority Example	Overall Accuracy		F ₁ -Measure: Minority Class		F ₂ -Measure: Minority Class	
		SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
SyntheticControl	2	98.83%	99.78%	0.967	0.993	0.984	0.991
CBF	1	96.89%	98.56%	0.950	0.978	0.928	0.966
FaceAll	1	98.83%	99.26%	0.906	0.940	0.931	0.939
OSULeaf	2	86.16%	87.05%	0.369	0.532	0.329	0.492
SwedishLeaf	8	98.27%	99.11%	0.855	0.938	0.814	0.940
50Words	1	98.78%	98.95%	0.324	0.466	0.278	0.416
Trace	2	91.50%	96.75%	0.792	0.934	0.748	0.930
TwoPatterns	2	99.78%	99.96%	0.995	0.999	0.993	0.999
Wafer	5	96.25%	99.81%	0.791	0.991	0.706	0.994
FaceFour	1	91.19%	96.88%	0.790	0.939	0.736	0.923
Lightning2	1	73.77%	83.61%	0.619	0.800	0.516	0.746
Lightning7	1	89.63%	93.54%	0.452	0.723	0.397	0.692
ECG	1	87.00%	93.00%	0.787	0.896	0.710	0.857
Adiac	3	98.07%	98.29%	0.442	0.625	0.377	0.576
Fish	3	94.86%	97.39%	0.755	0.907	0.686	0.889
Beef	1	82.67%	81.33%	0.167	0.342	0.167	0.310
OliveOil	1	91.11%	94.44%	0.571	0.745	0.543	0.702

UCR Data Sets and DTW

Shaded results indicate best performers; the darker the shade, the larger the difference between the results with and without ghost points

Data Set	# GP Added Per Minority Example	Overall Accuracy		F ₁ -Measure: Minority Class		F ₂ -Measure: Minority Class	
		SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
SyntheticControl	1	97.44%	99.28%	0.929	0.979	0.968	0.981
CBF	1	95.83%	97.72%	0.934	0.964	0.917	0.945
FaceAll	8	96.08%	97.56%	0.731	0.844	0.792	0.837
OSULeaf	2	85.12%	86.98%	0.345	0.478	0.309	0.432
SwedishLeaf	9	97.94%	98.71%	0.829	0.907	0.791	0.911
50Words	1	98.76%	98.97%	0.311	0.472	0.272	0.417
Trace	2	90.25%	95.50%	0.769	0.909	0.717	0.899
TwoPatterns	2	98.45%	99.08%	0.968	0.981	0.953	0.970
Wafer	5	96.82%	99.69%	0.830	0.986	0.759	0.988
FaceFour	1	83.52%	92.33%	0.515	0.835	0.464	0.790
Lightning2	1	77.05%	83.61%	0.682	0.792	0.586	0.720
Lightning7	2	90.02%	90.80%	0.441	0.588	0.421	0.581
ECG	2	82.00%	84.00%	0.710	0.742	0.647	0.676
Adiac	3	97.74%	98.05%	0.419	0.626	0.389	0.622
Fish	5	93.55%	95.76%	0.708	0.845	0.650	0.824
Beef	1	82.00%	81.33%	0.167	0.308	0.167	0.300
OliveOil	1	85.56%	91.11%	0.400	0.726	0.371	0.725

MPEG-7 Data Set

Shaded results indicate best performers; the darker the shade, the larger the difference between the results with and without ghost points

Distance Measure	Characteristics			Overall Accuracy		F ₁ -Measure: Minority Class		F ₂ -Measure: Minority Class	
	#GP Added Per Minority Example	Number of Minority Examples	Number of Majority Examples	SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
OSB	4	10	1390	99.43%	99.74%	0.710	0.897	0.662	0.868
DTW	5	10	1390	99.11%	99.20%	0.603	0.767	0.581	0.794

Types of Ghost Point Distance Calculations

- Computing the distance of a ghost point to other points can take one of three forms (see *Definition of Ghost Points*)
- We compute the number of each type for OSB and DTW on all data sets
 - See tables on next two slides
- Interesting note: most of the distance spaces induced by DTW contain very few Type 2 and Type 3 computations
 - This indicates that the distance space induced by DTW is very close to a metric space
- For distance spaces induced by OSB, the numbers are much more variable
 - The number of non-Type 1 computations ranges from 6% to 85%
 - Thus OSB is more likely to induce non-metric distance spaces
- Though we stress, and our experimental results show, that ghost points may be used to densify non-metric distance spaces

Types of Ghost Points and OSB

Data Set	#GP Added Per Minority Example	Number of Type 1 Dist. Comp.	Number of Type 2 Dist. Comp.	Number of Type 3 Dist. Comp.	Total Number of Distance Computations per Minority Class	Percentage of Type 1	Percentage of Type 2	Percentage of Type 3
SyntheticControl	1	29,389	10	1,826	31,225	94%	0%	6%
CBF	1	7,796	4	607	8,407	93%	0%	7%
FaceAll	1	67,809	48	22,923	90,780	75%	0%	25%
OSULeaf	2	26,738	42	5,126	31,905	84%	0%	16%
SwedishLeaf	7	182,556	261	107,177	289,993	63%	0%	37%
50Words	1	5,014	7	3,360	8,381	60%	0%	40%
Trace	2	1,871	34	9,352	11,256	17%	0%	83%
TwoPatterns	2	1,500,368	270	1,124,450	2,625,087	57%	0%	43%
Wafer	6	1,840,370	2,577	2,495,572	4,338,519	42%	0%	58%
FaceFour	1	551	4	134	689	80%	1%	19%
Lightning2	1	1,828	16	766	2,610	70%	1%	29%
Lightning7	1	741	9	734	1,484	50%	1%	49%
ECG	1	4,229	7	2,429	6,665	63%	0%	36%
Adiac	3	15,615	126	9,467	25,208	62%	1%	38%
Fish	3	13,706	65	15,282	29,053	47%	0%	53%
Beef	1	56	17	301	375	15%	5%	80%
OliveOil	1	235	2	118	355	66%	1%	33%
MPEG-7	4	51,195	23	5,561	56,780	90%	0%	10%

Types of Ghost Points and DTW

Data Set	#GP Added Per Minority Example	Number of Type 1 Dist. Comp.	Number of Type 2 Dist. Comp.	Number of Type 3 Dist. Comp.	Total Number of Distance Computations per Minority Class	Percentage of Type 1	Percentage of Type 2	Percentage of Type 3
SyntheticControl	1	30,982	0	243	31,225	99%	0%	1%
CBF	1	8,337	0	70	8,407	99%	0%	1%
FaceAll	6	549,721	13	18,946	568,680	97%	0%	3%
OSULeaf	2	31,609	1	295	31,905	99%	0%	1%
SwedishLeaf	7	267,539	43	22,411	289,993	92%	0%	8%
50Words	1	7,784	0	597	8,381	93%	0%	7%
Trace	2	7,504	15	3,738	11,256	67%	0%	33%
TwoPatterns	2	2,544,129	13	80,946	2,625,087	97%	0%	3%
Wafer	5	3,566,072	274	25,564	3,591,910	99%	0%	1%
FaceFour	1	688	0	2	689	100%	0%	0%
Lightning2	1	2,591	0	19	2,610	99%	0%	1%
Lightning7	2	3,055	0	29	3,085	99%	0%	1%
ECG	2	14,003	0	288	14,291	98%	0%	2%
Adiac	3	22,581	101	2,526	25,208	90%	0%	10%
Fish	2	15,353	3	3,381	18,738	82%	0%	18%
Beef	1	209	0	166	375	56%	0%	44%
OliveOil	1	355	0	0	355	100%	0%	0%
MPEG-7	4	71,101	22	103	71,225	100%	0%	0%

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 **Summary and Future Work**
- 5 For Further Reading

When Outliers Corrupt

- OSB
 - OSB directly optimizes the sum of distances of corresponding elements
 - Allows penalized skipping of outlier elements
 - Defines a bijection on the remaining subsequences
 - The penalty for skipping outliers is part of the edge weights of the DAG built from two matched sequences
 - This results in skipping decisions being made with a dynamic threshold whose optimization is directly included in the dynamic programming optimization

When Outliers Are Important

- Ghost Points

- An innovative method for over-sampling the minority class of imbalanced data sets
- Unlike other feature based methods ghost points are added in distance space
- In addition, ghost points can be added to distance spaces that are not metric

Future Work

- Trying a non-linear jumpcost penalty
- Exploring optimal strategies for inserting ghost points.
- Choosing the optimal number of ghost points

Outline

- 1 Introduction and Motivation
 - The Story
 - Time Series
 - Imbalanced Data Sets
 - Thesis
- 2 Optimal Subsequence Bijection
 - Existing Distance Measures
 - OSB
 - Experimental Results
- 3 Ghost Points
 - Distance Spaces
 - Ghost Points
 - Experimental Methodology
 - Experimental Results
- 4 Summary and Future Work
- 5 For Further Reading

My Publications

● Journals

- **Suzan Köknar-Tezel** and Longin Jan Latecki. Improving SVM classification on imbalanced time series data sets with ghost points. Knowledge and Information Systems, June 2010 (OnlineFirst)

● Conference Proceedings

- **Suzan Köknar-Tezel** and Longin Jan Latecki. Improving SVM classification on imbalanced data sets in distance spaces. Proceedings of the IEEE International Conf. on Data Mining (ICDM), pages 259-267, 2009. (Blind peer reviewed; Acceptance rate = 9%).
- Xingwei Yang, **Suzan Köknar-Tezel**, and Longin Jan Latecki. Locally constrained diffusion process on locally densified distance spaces with application to shape retrieval. Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 357-364, 2009. (Blind peer reviewed; Acceptance rate = 26.2%)
- Longin Jan Latecki, Qiang Wang, **Suzan Köknar-Tezel**, and Vasileios Megalooikonomou. Optimal subsequence bijection. IEEE International Conference on Data Mining, pages 565-570, 2007. (Blind peer reviewed; Acceptance rate = 19%).
- Longin Jan Latecki, **Suzan Köknar-Tezel**, Qiang Wang, and Vasileios Megalooikonomou. Sequence matching capable of excluding outliers. Proceedings of the Workshop on Time Series Classification at the Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2007. (Peer reviewed).

● Submitted for review to Journals

- **Suzan Köknar-Tezel** and Longin Jan Latecki. Sequence matching as subsequence bijection. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI).
- Xingwei Yang, Xiang Bai, **Suzan Köknar-Tezel**, and Longin Jan Latecki. Densifying Distance Spaces. Pattern Recognition (PR).

References I



John Aach and George M. Church.

Aligning gene expression time series with time warping algorithms.

Bioinformatics, 17:495–508, 2001.



R. Akbani, S. Kwek, and N. Japkowicz.

Applying support vector machines to imbalanced datasets.

In *Proceedings of ECML'04*, pages 39–50, 2004.



N. V. Chawla, K. W. Bowyer, and W. P. Kegelmeyer.

Smote: Synthetic minority over-sampling technique.

Journal of Artificial Intelligence Research, 16:321–357, 2002.



Chiu, Keogh, and Lonardi.

Probabilistic discovery of time series motifs.

In *Proceedings ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Washington*, 2003.



Nitesh V. Chawla, Ar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer.

Smoteboost: improving prediction of the minority class in boosting.

In *Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*, pages 107–119, 2003.

References II



Lei Chen and Raymond Ng.

On the marriage of l_p -norms and edit distance.

In *VLDB '04: Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 792–803. VLDB Endowment, 2004.



Philip Chan and Salvatore J. Stolfo.

Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection.

In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 164–168. AAAI Press, 1998.



Das, Gunopulos, and Mannila.

Finding similar time series.

In *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.



H. Gray Funkhouser.

A note on a tenth century graph.

Osiris, 1:260–262, 1936.



Costis Georgiou and Hamed Hatami.

CSC2414- Metric embeddings. Lecture 1: A brief introduction to metric embeddings, examples and motivation.

2008.

References III



Hui Han, Wenyuan Wang, and Binghuan Mao.
Borderline-smote: A new over-sampling method in imbalanced data sets learning.
volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer, 2005.



David W. Jacobs, Daphna Weinshall, and Yoram Gdalyahu.
Classification with nonmetric distances: Image retrieval and class representation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:583–600, 2000.



Miroslav Kubat, Robert C. Holte, and Stan Matwin.
Machine learning for the detection of oil spills in satellite radar images.
Machine Learning, 30(2-3):195–215, 1998.



Keogh, Lonardi, and Ratanamahatana.
Towards parameter-free data mining.
In *Proceedings ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Seattle*, 2004.



Eamonn Keogh and Chotirat Ann Ratanamahatana.
Exact indexing of dynamic time warping.
Knowl. Inf. Syst., 7(3):358–386, 2005.



Suzan Köknar-Tezel and Longin Jan Latecki.
Improving SVM classification on imbalanced time series data sets with ghost points.
Knowledge and Information Systems, June 2010.

References IV



Suzan Köknar-Tezel and Longin Jan Latecki.
Sequence matching as subsequence bijection.
Submitted, 2010.



E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana.
UCR time series classification/clustering page.
Website, 2006.
http://www.cs.ucr.edu/~eamonn/time_series_data/.



Longin Jan Latecki, Rolf Lakämper, and Ulrich Eckhardt.
Shape descriptors for non-rigid shapes with a single closed contour.
In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 424–429, 2000.



Julian Laub and Klaus-Robert Müller.
Feature discovery in non-metric pairwise data.
Journal of Machine Learning Research, 5:801–818, 2004.



Longin Jan Latecki, Qiang Wang, Suzan Köknar-Tezel, and Vasileios Megalooikonomou.
Optimal subsequence bijection.
IEEE International Conference on Data Mining, pages 565–570, 2007.



Pierre-François Marteau.
Time warp edit distance with stiffness adjustment for time series matching.
IEEE Trans. Pattern Anal. Mach. Intell., 31(2):306–318, 2009.

References V



Jiri Matousek.

Lectures on Discrete Geometry.

Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.



Rafiei.

On similarity-based queries for time series data.

In Proceedings of the Int. Conf. on Data Engineering, Sydney, pages 410–417, 1999.



Chotirat Ann Ratanamahatana and Eamonn Keogh.

Everything you know about dynamic time warping is wrong, 2004.



Hiroaki Sakoe and Seibi Chiba.

A dynamic programming approach to continuous speech recognition.

In Proceedings of the Seventh International Congress on Acoustics, Budapest, volume 3, pages 65–69, Budapest, 1971. Akadémiai Kiadó.



Salvador, Chan, and Brodie.

Learning states and rules for time series anomaly detection.

In Proceedings of the 17th Intl. Florida Artificial Intelligence Research Society Conference, Florida, pages 306–311, 2004.

References VI



Vlachos, Hadjieleftheriou, Gunopulos, and Keogh.

Indexing multi-dimensional time-series with support for multiple distance measures.

In *Proceedings of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Washington*, pages 216–225, 2003.



V. M. Velichko and N. G. Zagoruyko.

Automatic recognition of 200 words.

International Journal of Man-Machine Studies, 2:223–234, 1970.



K. Woods, C. Doss, K. Bowyer, J. Solka, C. Priebe, and P. Kegelmeyer.

Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography.

International Journal of Pattern Recognition and Artificial Intelligence, 7:1417–1436, 1993.



Xingwei Yang, Xiang Bai, Longin Jan Latecki, and Zhuowen Tu.

Improving shape retrieval by learning graph transduction.

In *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 788–801. Springer, 2008.