Combining Similarity in Time and Space for Training Set Formation under Concept Drift

Indrė Žliobaitė

Vilnius University, Naugarduko 24, LT-03225 Vilnius, Lithuania
Eindhoven University of Technology, PO Box 513, NL 5600 MB Eindhoven, the Netherlands

e-mail: zliobaite@gmail.com
telephone: +31 40 247 2733 , fax: +31 40 246 3992

Abstract

Concept drift is a challenge in supervised learning for sequential data. It describes a phenomenon when the data distributions change over time. In such a case accuracy of a classifier benefits from the selective sampling for training. We develop a method for training set selection, particularly relevant when the expected drift is gradual. Training set selection at each time step is based on the distance to the target instance. The distance function combines similarity in space and in time. The method determines an optimal training set size online at every time step using cross validation. It is a wrapper approach, it can be used plugging in different base classifiers. The proposed method shows the best accuracy in the peer group on the real and artificial drifting data. The method complexity is reasonable for the field applications.

Keywords: concept drift; gradual drift; online learning; instance selection

1 Introduction

Concept drift challenges building supervised learning models for sequential data. The data distribution might change over time due to, for example, changes in user interests (recommender systems), external unobserved variables (bankruptcy prediction) or adversary activities (fraud detection). Thus adaptive learning models are required.

In supervised learning adaptivity can be achieved either by designing specific base learners (e.g. [1]) or by manipulating training set over time in instance or feature space, or both. Manipulating training set includes instance selection (e.g. training windows [2], selective sampling [3]), instance weighting [4] and dynamic feature selection [5]. Training set manipulation strategies are wrapper approaches in a sense that they can be used for online learning plugging in different types of base classifiers.

Sequential instance selection (training windows) is typically used at sudden concept drift. Training window strategies select the nearest neighbors in time to form a training set. Selective sampling in space is particularly beneficial when reoccurring concepts are expected. In such a case the closest instances in the feature space to the target instance are selected to form a training set.

In this study we present a concept of combining distances in time and space for training set selection under concept drift. A combined view to

instance selection is required due to complex nature of the real data. Therefore, a unified view to the training sample formation is proposed which is flexible with respect to the actual changes.

Preliminary results were presented in a short conference paper [6]. That study was delimited to the fixed proportion of the distance in time and space.

Using time and space similarity concept we develop a method for classifier training, especially relevant when the expected drift is gradual. Training set selection is based on similarity to the target instance. Distances in space and in time are linearly combined. The method determines an optimal training set size online at every time step using cross validation. It is used as a wrapper approach, that means different base classifiers can be plugged in.

The proposed method shows the best accuracy in the peer group on the real and artificial drifting data. The method complexity is reasonable for the field applications. The method is expected to demonstrate a competitive advantage under gradual drift scenarios in small and moderate size data sequences.

The paper is organized as follows. We start by a motivation for combining time and space similarity for training set seletion in Section 2, followed by fixing the framework and basic assumptions. Next we introduce and illustrate the concept of distance in time and space in Section 3. The following Section 4 outlines particularly related work and maps the proposed concept within the related work. In Section 5 we present the proposed methods. Section 6 gives experimental setup and the results. Sections 8 and 9 discuss the results and conclude.

2 Problem Set-up

In this section we present a motivation for combining similarity in time and space for training set formation under concept drift and fix the set-up and basic assumptions for this study.

2.1 Motivation

Assume an online recommender system where a user reads online news. When she became more interested in real estate, market news were appearing more and more often as the most interesting topic. At the same time she was still interested in meat prices in New Zealand, but the relative interest was declining. Thus the relevance of a given document to the reader's interests

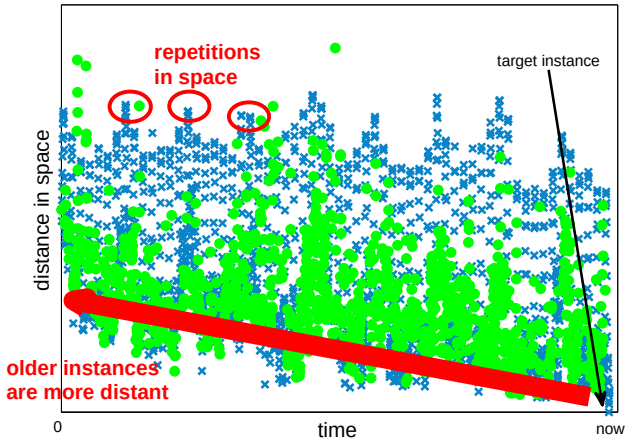

Figure 1: A snapshot of electricity demand data, • and × mark classes.

depended on the age of the document (distance in time) and the content (distance in space).

In order to build a classifier, which would react to a gradual concept drift, we aim to select the most relevant historical instances to form a training set. In the online news example, for each incoming document (unlabeled) we would look for *similar* documents within the historical stock.

Similarity between two objects in instance based learning [7] is defined as a function of distance in space. If the domain is non stationary distance in time might be relevant as well. For an illustration see a snapshot of Electricity data [8], which is provided in Figure 1.

We plot the distance of the historical instances to the target instance over time against the distance in space (here we use the Euclidean distance in the feature space). The target instance is the very last in time (denoted 'now'), and its distance to itself in space is 0. The older instances are generally further from the target instance (declining slope along with x axis), which indicates the relevance of similarity in time. Recent instances are closer to the target instance. Moreover, there are notable recurrences in space, indicated by circles. This advocates that sequential sampling (window) might miss relevant training instances. Thus both distances in space and in time are to be taken into consideration.

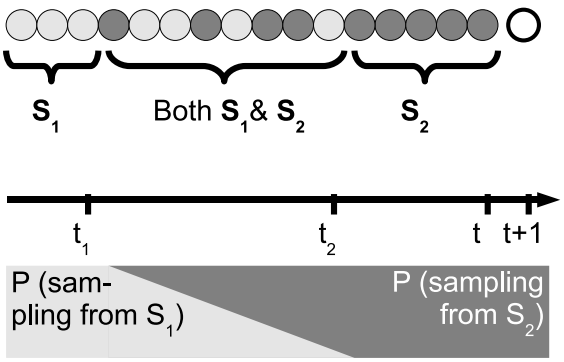

Figure 2: Gradual drift scenario.

2.2 Set-up and Basic Assumptions

Assume online classification task. One data instance $\mathbf{X} \in \Re^p$ is received at a time, the corresponding discrete class label $\mathbf{y}$ is unknown. At time $t+1$ the task is to predict the class label $\mathbf{y}_{t+1}$ for the target instance $\mathbf{X}_{t+1}$. It is allowed to retrain a classifier at every time step if needed.

Any selected or all the historical labeled data $\mathbf{X}_1, \ldots, \mathbf{X}_t$ with corresponding known labels $\mathbf{y}_1, \ldots, \mathbf{y}_t$ can be used as a training set for a classifier at time $t+1$. At time $t+2$ after the classification decision the true label can be received, we can add $\mathbf{X}_{t+1}$ to the training data and proceed with the decision making for a target instance $\mathbf{X}_{t+2}$.

Consider a gradual drift scenario, illustrated in Figure 2. Up to time t_1 data generating source S_I is active. Note, that the source is not the same as class label. Each source can generate an instance from either of the classes. A source can be considered to be a distribution. From time t_2+1 on the source S_I is completely replaced by the source S_{II}. In time interval (t_1+1, t_2) both sources are active and an instance comes from either one or the other source with a prior probability. The probability of sampling from S_{II} increases with time. A designer does not know when the sources switch.

The task is to assign a class label to an instance $\mathbf{X}_{t+1}$. It is expected that a concept drift might have taken place, i.e. several sources were active up to time $t+1$. In order to build an accurate classifier we would like the training data to come from the same or as close as possible source as the target data $\mathbf{X}_{t+1}$. We can find how similar $\mathbf{X}_{t+1}$ is to the historical instances even though the label of $\mathbf{X}_{t+1}$ is not known. We aim to select a training set,

consisting of the instances, which are *similar* to the target instance.

Similarity is a share of commonality. A detailed discussion on similarity concept can be found in [9]. In the next section we define similarity as a function of distances in time and space.

3 Similarity in Time and Space for Training Set Selection

In this section we introduce and explore the concept of combining distances in time and space for training set selection to achieve adaptive learning. First we define how to measure similarity and then look how to use it for training set selection.

3.1 The Concept of Similarity in Time and Space

Let *the similarity in time and space* between the target instance $\mathbf{X}_j$ and a historical instance $\mathbf{X}_i$ be a distance function

$$\mathcal{D}(\mathbf{X}_i, \mathbf{X}_j) = f(d_{ij}^{(S)}, d_{ij}^{(\mathcal{T})}), \tag{1}$$

where $d_{ij}^{(S)}$ is the distance between the two instances in space and $d_{ij}^{(\mathcal{T})}$ is the distance between the two instances in time. The smaller the distance, the more similar are the instances.

Distance in time between the instances $\mathbf{X}_i$ and $\mathbf{X}_j$ in case of equally spaced time intervals is defined as a function

$$d_{ij}^{(\mathcal{T})} = f(|i - j|). \tag{2}$$

Different distance function can be chosen based on the domain knowledge and visual inspection of the data. For instance, exponential function can be used aiming to emphasize the recent times, $d_{ij}^{(\mathcal{T})} = e^{|i-j|}$. In this study we use a linear distance, which is the least complex option $d_{ij}^{(\mathcal{T})} = |i - j|$.

Time intervals can be unequally spaced, e.g. stock prices are recorded only on weekdays, thus there is a three days gap between the Friday value and the Monday value. In that case $d_{ij}^{(\mathcal{T})} = |\mathcal{T}(i) - \mathcal{T}(j)|$, where $\mathcal{T}$ is the function mapping indexes to actual time values.

Distance in space can have a number of alternative metrics (e.g. City-block, Euclidean distances), a discussion of the most common metrics can be found in [10, 11]. A distance metric of designer's choice can be used.

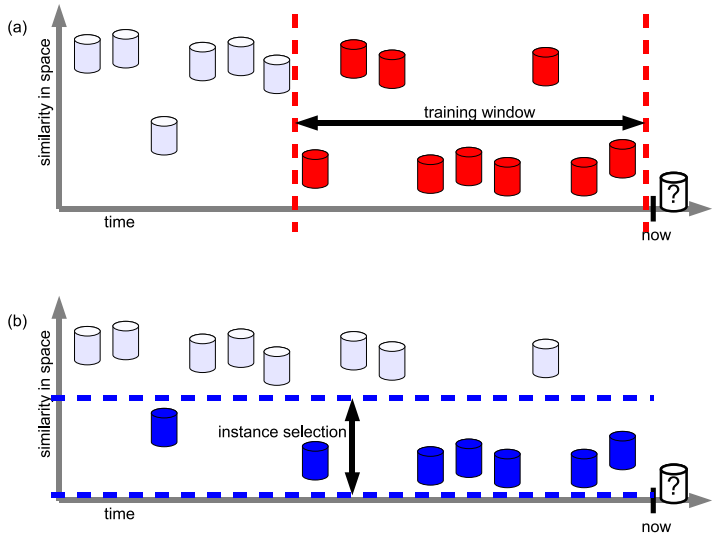

Figure 3: Training set selection boundary cases: (a) selection based only on the distance in time (training window), (b) only distance in space (instance selection).

We use two terms, *similarity* and *distance*, which are inversely related. The larger is distance, the smaller is similarity. We use the term similarity when referring to a general concept and the term distance when referring to an actual metric.

3.2 Combining Distances in Time and Space

The form of a combination function $\mathcal{D}$ depends on the expectations of a designer related to the data at hand. The choice of time and space proportions directly depends on the observed change types and the future expectations. The goal is to select the training set in a way that it would represent the current target instances well.

Let us look at the two boundary cases. If a designer selects training set only based on the distance in time, that is a training window strategy (see Figure 3(a)). The most recent instances are selected as a training set in a sequential order. Another boundary case is to disregard time and select training set only based on the distance in space (see Figure 3(b)).

In Figure 4(a) we illustrate a linear combination of the distances in time and space, which we present in this study. Note that a designer is not limited to linear combination. For instance, an example in Figure 4(b)

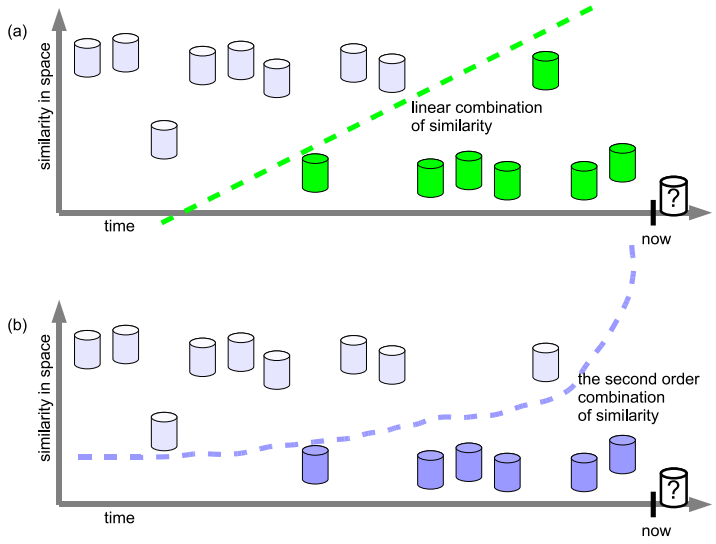

Figure 4: Time and space based training set selection: (a) linear combination, (b) the second order combination.

might be considered if an emphasis of the boundary instances is to be made.

The linearly combined distance between the instances $\mathbf{X}_i$ and $\mathbf{X}_j$ is

$$\mathcal{D}(\mathbf{X}_i, \mathbf{X}_j) = \alpha_1 d_{ij}^{(S)} + \alpha_2 d_{ij}^{(\mathcal{T})}, \tag{3}$$

where α_1 and α_2 are the weight coefficients. If $\alpha_1 = 0$, it is a training window, as in Figure 3(a). If $\alpha_2 = 0$, it is an instance selection, as in Figure 3(b). As a design choice, the weights α_1, α_2 can be fixed based on the domain knowledge or visual inspection of the data or they can be trainable on a validation set or online.

For interpretation we normalize the proportions of time and space in the distance function $d^{(S)}$ and $d^{(\mathcal{T})}$. We scale the values of each feature in $\mathbf{X}$ to the interval $[0, \frac{1}{p}]$ in order to get $d^{(S)} \in [0, 1]$. We scale the time distances to get $d_{ij}^{(\mathcal{T})} \in [0, 1]$ as well. For a single dataset scaling is not essential, since the proportion can be regulated by the weights α_1 and α_2. However, this way time and space distances become comparable across different datasets.

For training set selection under concept drift we are interested in relative distances (ranking). Thus, for simplicity, α_1 and α_2 can be replaced by $A = \frac{\alpha_2}{\alpha_1}$, assuming that $\alpha_1 \neq 0$. We are interested in the ranks of distances between the historical instances and the target instance $\mathbf{X}_{t+1}$. Thus, we can

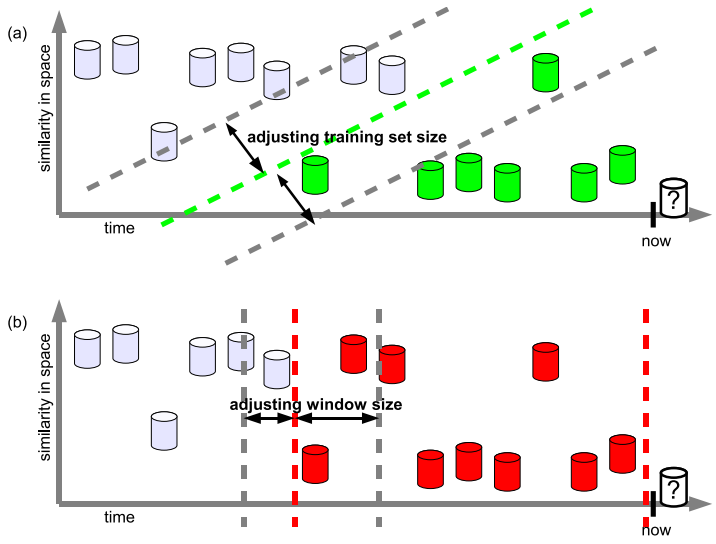

Figure 5: Training set size: (a) selection based on combined distance in time and space, (b) training window.

simplify Equation 3 to

$$\mathcal{D}^*(\mathbf{X}_i, \mathbf{X}_{t+1}) = d^{(S)}_{i,t+1} + Ad^{(T)}_{i,t+1} = \mathcal{D}^*_i. \tag{4}$$

3.3 Training Set Size

We defined the distance in time and space $\mathcal{D}^*$, intended to be used for ranking the historical data $(\mathbf{X}_1, \ldots, \mathbf{X}_t)$ according to the distance to the target instance $\mathbf{X}_{t+1}$. Another important choice in constructing a training set is *how many* from the most similar instances to include into the training set.

The training set size is specified applying a threshold to the distance measure. After the distance measure $\mathcal{D}^*$ is fixed, the training set size can be decided by moving the decision threshold, as shown in Figure 5(a). Note, that here the slope is fixed. It indicates the proportions of time and space distances in the final distance measure, as described in Equation (3).

The threshold principle is the same as in variable window size selection, the case is illustrated in Figure 5(b).

Thus, having an unlabeled target instance $\mathbf{X}_{t+1}$, for $i = 1, \ldots, t$ the instance $\mathbf{X}_i$ is selected into a training set if $\mathcal{D}^*(\mathbf{X}_i, \mathbf{X}_{t+1}) < h^D$, where h^D

is the training set threshold. The threshold can be fixed by a designer or trainable based on a validation set.

4 Positioning within the Related Work

In this section we present the related work and its relation to our approach.

We contribute to the field by generalizing training set selection using time and space similarity. To our best knowledge the representation *unifying* windowing and instance selection under concept drift has not been formulated before. There are related techniques which are implicitly using instance forgetting when employing instance selection strategy, e.g. [12], which is mainly to overcome computational challenges in data streams.

Our approach integrates windowing techniques and instance selection techniques under unified framework of systematic training set selection. Moreover, it extends the existing approaches to a combination of both windowing and instance selection.

The issue of systematic training set selection in space under concept drift has been brought up in [3, 13–17]. Ganti et al. [13] give a generic interpretation of systematic training data selection without a real plug-and-play algorithm. The blocks (intervals) of training data can be picked using moving window based templates.

The following two approaches use space based training set selection techniques. Tsymbal et al. [15] use an ensemble, where the competence of the base classifiers is determined by cross validation on the nearest neighbors of the target instance. However, they use training windows to *build* the individual base classifiers. Katakis et al [17] organize the training data into clusters, derive prototypes for each cluster and then select the clusters for training based on the distances between the target instance and the prototypes. Since the main focus is on reoccurring concepts, time similarity is not integrated there.

Valizadegan and Tan [16] use intelligent training set selection procedure after a change is detected. They aim to acquire more samples from the regions where classification is unreliable. They call the strategy differed-boosting and deferred active approach, where deferred means that resampling is triggered by a change detection.

The above approaches limit the history in time from which the instances can be selected. That is an implicit assumption in data stream mining, where the data streams in principle are endless. The approaches overviewed here have a clear cut in history, without incorporating time features into

instance selection procedure.

Beringer and Hullermeier [3] organize training data into prototype clusters, referred to as case bases. In contrast to the peer works, they exclude the instances which are too similar to the ones already present in the training set. They explicitly address relevance in time and space, as well as consistency. The major difference in our and their approach is in the future assumption. They assume continuous concept (and call it consistency). Track the concept itself and drop out inconsistent data. The approach would be unfavorable to reoccurring concepts and robust to noise. In our approach we determine the concept for a target instance without tracking the change. This way relevant training set might be found as well in case of reoccurring concepts and even in case of noise.

Lazarescu and Verkantesh [14] use time and space dimensions to determine the relevance of a given historical instance. The idea is closely related to the work by Beringer and Hullermeier [3] and has the same limitations regarding following the current concept. The former work was presented four years later than the latter.

Adaptive nearest neighbor classification [12, 18] is related to our approach. Ueno et al [18] focus on the computational complexity issues in streaming kNN application, not on training set selection directly. Their idea is to introduce an order in which the comparison of the distances between the instances is processed, that is likely to give more accurate results than random order, if the comparison is stopped before the end of the historical data is reached. Law and Zaniolo [12] use exponential weighting of the instances in time. They use the grid to divide the space into neighborhood region and adapt only the grids, where the newly arrived instances belong. Building the neighborhood can be viewed as instance selection in space, but their approach is kNN classifier specific. The griding mechanism is explicitly oriented towards forming a single class cells, while generalization to different base classifiers would require an opposite strategy.

Finally, Black and Hickey [19] use the idea of augmenting the feature space by adding a time stamp feature. Then they use training window approaches, thus they do not employ space based selection. In principle the time feature can be integrated with space based instance selection. Augmenting the feature space and then measuring distances in the new space can be more flexible with respect to base classifier related adaptivity, than the combination we are taking. For instance, time related splits in a decision tree can be organized. We choose the explicit combination of distances in time and space for training set selection mainly because it can be easier to control and interpret.

We propose an approach for training set selection based on similarity to target instance. There are multiple classifier methods (e.g. [15]) employing similarity aspect but only in the classifier selection phase. However, the needed classifier might not be present among the ensemble members. From similarity in space perspective our approach is related to a lazy learning [20], but the main difference is that the latter does not construct explicit generalization and makes classification based on direct comparison of the target and training instances. Our approach can be used as a wrapper with different base classifiers. The closest approaches [3, 14] try to follow the concept changes, thus are not straightforward to generalize to reoccurring concepts.

5 FISH Method Family

To support our approach of combining distance in time and space, we propose a family of methods called FISH (uni**F**ied **I**nstance **S**election algorit**H**m), which incorporates the ideas presented above. Training instances are systematically selected at each time step. The methods can be used with different base classifiers.

The family includes three modifications: FISH1, FISH2 and FISH3. In FISH1 the size of a training set is fixed and set in advance, the extension FISH2 operates using variable training set size. In FISH2 the proportion of time and space distances (α_1 and α_2 in Equation 3) in the final distance measure are fixed in advance as a design choice. We present a modification of FISH3, where this proportion is trainable online.

We consider FISH2 to be the central in the family. We believe that in many cases the optimal proportions of time and space in the distance function are domain dependent and can be fixed offline (for instance, using an offline validation set). On the other hand, the drifts might take non uniform speeds, thus online adjustable training set size is relevant.

Next we give pseudo code and explain the details and intuition for each of the three FISH methods.

5.1 FISH1

We start with presenting FISH1 in Figure 6. The pseudo code includes the steps for training set selection for decision making at time $t + 1$.

The method ranks the historical instances (without their labels) according to their distance to the target instance and picks N the most similar instances to form a training set. Since the size of the training set is fixed, if

THE TRAINING SET SELECTION METHOD (FISH2)

INPUT
Data: historical instances $\mathbf{X^H} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ with labels $\mathbf{y^H}$, target instance $\mathbf{X}_{t+1}$ without a label.
Parameters: training set size N, time/space proportion A (Equation (4)).
Base learner type: $\mathcal{L}$.
ALGORITHM

1. Calculate distances in time and space $\mathcal{D}_i^*$ (Equation (4)) *for* $i = 1 : t$.
2. Sort the distances from minimum to maximum $\mathcal{D}_{z1}^* < \mathcal{D}_{z2}^* < \ldots < \mathcal{D}_{zt}^*$. Indexes $z1, \ldots, zt$ define the permutation $(\mathbf{X}_1, \ldots, \mathbf{X}_t) \rightarrow (\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zt})$.
3. Pick N instances having the smallest distances $\mathcal{D}$.
4. Output the indexes $\{z1, \ldots, zN\}$.

OUTPUT
The indexes $\mathcal{I}_t = \{z1, \ldots, zN\}$ to form a training set $\mathbf{X_t^T} = (\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zN})$.

Figure 6: FISH1 method for fixed size training set.

it happens to be to small, the instances from a single class might end up in a training set. To insure from this, we suggest selecting a stratified training set. It means that $\frac{N}{c}$ most similar instances are selected from each class, altogether forming a training set of size N.

5.2 FISH2

FISH1 uses fixed training set size N. FISH2 is an extension, where the training set size is learnable online. To implement a variable training size we incorporated the ideas inspired by two windowing methods [21] (KLI) and [15] (TSY). FISH2 is presented in Figure 7.

We start by calculating the distances in time and space between the target instance $\mathbf{X}_{t+1}$ and every historical instance in $\mathbf{X^H}$ as in Equation (4). The distances to each historical instance are ranked based on the distance.

Next we decide *how many* of the most similar training instances to pick using cross validation. For that we build a set of classifiers $(\mathcal{L}^1, \mathcal{L}^2, \ldots, \mathcal{L}^N)$ using different training set sizes. The validation set is formed using k historical instances, which were found to be the most similar to the target instance $\mathbf{X}_{t+1}$. We select the training size N^*, which has given the best accuracy on

The Training Set Selection method (FISH2)

INPUT

Data: $\mathbf{X^H}$, $\mathbf{y^H}$, $\mathbf{X}_{t+1}$. Param.: neighborhood size k, A . Base learner type: $\mathcal{L}$.

ALGORITHM

1. Calculate distances in time and space $\mathcal{D}_i^*$ (Equation (4)) *for* $i = 1 : t$.
2. Sort the distances from minimum to maximum $\mathcal{D}_{z1}^* < \mathcal{D}_{z2}^* < \ldots < \mathcal{D}_{zt}^*$.
3. *For* $N = k : step : t$ select the training set size
   (a) pick N instances having the smallest distances $\mathcal{D}$,
   (b) using cross-validation[a] build a classifier $\mathcal{L}^N$ using the instances $(\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zN})$ as the training set,
   (c) test $\mathcal{L}^N$ on the k nearest neighbors $(\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zk})$, record testing error e_N.
4. Find the minimum error classifier $\mathcal{L}^{N*}$, where $N^* = \arg\min_{N=k}^{t}(e_N)$.
5. Output the indexes $\{z1, \ldots, zN^*\}$.

OUTPUT

The indexes $\mathcal{I}_t = \{z1, \ldots, zN^*\}$ to form a training set $\mathbf{X_t^T}$.

[a] when test on the instance $\mathbf{X}_{zk}$, this instance is excluded from the validation set

Figure 7: FISH2 method for variable training set selection.

the validation set. The method works similarly to windowing in [21] (KLI). They use sequential instances in time to form the windows. We employ a combined distance metric in time and space.

Leave-one-out cross validation needs to be employed. It means that we repeat the validation process k times for every training set size N being checked. Each time we leave out one validation instance from the training set and then test on it. Without cross validation the training set of size k is likely to give the best accuracy, because in that case training set would be equal to the validation set.

The outcome of the method is a set of N^* indexes $\mathcal{I}_t = \{z1, \ldots, zN^*\}$. They indicate the historical instances to be picked as a training set $\mathbf{X_t^T} = (\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zN})$. Using the original instances $\mathbf{X_t^T}$ a classifier $\mathcal{L}^{N*_t}$ is trained for the final prediction of the label $\mathbf{y}_{t+1}$ for the target instance $\mathbf{X}_{t+1}$.

5.3 FISH3

FISH3 is a extension of FISH2. FISH2 uses a prefixed proportion A of distances in time and space. FISH3 can learn the proportion *online* using an additional loop of cross validation. FISH3 is presented in Figure 8. Instead of fixing the proportion between time and space distances in $\mathcal{D}$ in Equation (3) we try a number of options and pick the learner which is the most accurate on the validation set, the same principle as in FISH2.

6 Experimental Evaluation

In order to verify the properties of FISH methods we carry out extensive numerical experiments. The main goal is to illustrate the advantage of combining distances in time and space as compared to using only time or only space criterion. We implement two peer methods and run them in parallel to FISH on six datasets. In order to minimize the bias of base classifier selection we run the experiments using four different base classifiers and two alternative distance in space measures.

6.1 Datasets

We use six data sets with potential gradual drift. Three datasets are real (Luxembourg, Ozone, Electricity), three other are real with an artificially introduced drift (German, Vote2, Iono2). The expectations of the drift are related to the domain of the real data and the way artificial drift is intro-

The Training Set Selection method (FISH2)

INPUT
Data: $\mathbf{X^H}$, $\mathbf{y^H}$, $\mathbf{X}_{t+1}$. Parameters: k. Base learner type: $\mathcal{L}$.
ALGORITHM

- *For* $j = 0 : step : 1$, $\alpha_1 = j$ $\alpha_2 = 1 - \alpha_1$ every time and space proportion
  1. calculate distances $\mathcal{D}_i^j = \alpha_1 d_{i,t+1}^{(S)} + \alpha_2 d_{i,t+1}^{(\mathcal{T})}$ (Equation (3)) *for* $i = 1 : t$.
  2. Sort the distances from minimum to maximum $\mathcal{D}_{jz1}^j < \mathcal{D}_{jz2}^j < \ldots < \mathcal{D}_{jzt}^j$.
  3. *For* $N = k : step2 : t$ select the training set size
     (a) pick N instances having the smallest distances $\mathcal{D}^j$,
     (b) using cross-validation[a] build a classifier $\mathcal{L}^{jN}$ using the instances $(\mathbf{X}_{jz1}, \ldots, \mathbf{X}_{jzN})$ as the training set,
     (c) test $\mathcal{L}^{jN}$ on the k nearest neighbors $(\mathbf{X}_{jz1}, \ldots, \mathbf{X}_{jzk})$, record testing error e_N^j.
- Find the minimum error classifier $\mathcal{L}^{jN*}$, where $jN^* = \arg\min_{j=0}^{1} \min_{N=k}^{t}(e_N^j)$.
- Output the indexes $\{jz1, \ldots, jzN^*\}$.

OUTPUT
The indexes $\mathcal{I}_t = \{jz1, \ldots, jzN^*\}$ to form a training set $\mathbf{X_t^T}$.

[a] when test on the instance $\mathbf{X}_{jzk}$, this instance is excluded from the validation set

Figure 8: FISH3 method with learnable training set size and distance proportion.

duced. All the datasets imply binary classification task. The characteristics of the datasets are summarized in Table 1.

Table 1: Summary of the used datasets.

Name	Dimensions	Size	Class balance	Type of data	Source of drift	Type of drift
Luxembourg	31	1901	0.51:0.49	real	real	gradual
Ozone	72	2534	0.94:0.06	real	real	gradual
Electricity	6	2956	0.57:0.43	real	real	gradual
German	23	1000	0.70:0.30	real	simulated	gradual
Vote2	16	435	0.61:0.39	real	simulated	gradual
Iono2	43	435	0.61:0.39	real	simulated	gradual

We constructed[1] Luxembourg dataset using social survey data from [22–24][2]. Each instance is a person. The task is to predict if she is a heavy internet user. It is relevant for marketing purposes. Ozone dataset [25] consists of air measurements, the task is to predict ozone level eight hours ahead. Electricity data [8] characterizes electricity demand in Australia, the task is to predict electricity market price.

German credit data [25] consists of individual credit application records, the task is to predict bankruptcy. We introduced artificial drift in German credit data by hiding the age feature. We do not introduce any synthetic drift in Iono and Vote datasets [25]. However, we refer to the drift as artificial, since we assume the data is presented in a time order, although it is not explicitly stated. Can we claim that there is a drift? If a selective sampling gives better classification accuracy than a growing window in incremental learning process, this can be treated as the drift evidence.

In Figure 9 we visualize all the datasets on time against distance in space axes. Note that the distances to one data point are visualized, which is rather a snapshot than a representation of the whole dataset. For illustration we use cosine distances in space.

6.2 Experimental Scenario

We perform three series of experiments related to FISH1, FISH2 and FISH3 correspondingly.

[1]The dataset is available at
`http://sites.google.com/site/zliobaite/resources-1`

[2]Norwegian Social Science Data Services (NSD) acts as the archive and distributor of the ESS data.

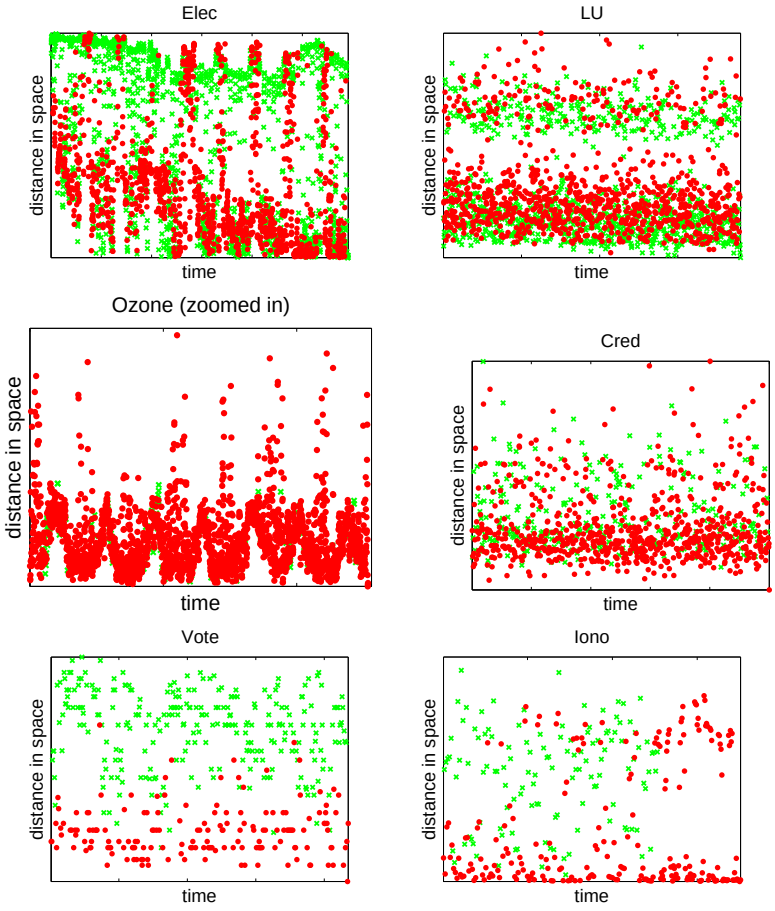

Figure 9: Visualization of the used datasets, • and × mark classes.

6.2.1 FISH1 and fixed training set size

First, we run controlled FISH1 experiments. We vary the proportion of time and space $A = \frac{\alpha_2}{\alpha_1}$ in the distance function (Equation 3) to analyze the effect to the final classification accuracy. We use fixed training set size N. The extreme $\alpha_1 = 0$ corresponds to a fixed training window. Contrary, $\alpha_2 = 0$ corresponds only the distance in space.

We include a baseline ALL, which is using all the historical data as the training set. Thus it does not select training data, every time step the training set is growing. The classifier is retrained using all the past data. If the data happens to be stationary, ALL should be the most accurate.

The pseudo code for ALL is provided in Appendix A.

6.2.2 FISH2 and variable training set size

We present FISH2 as a flagman in the FISH family and perform an extensive experimental evaluation for it. We test FISH2 by plugging in four alternative base classifiers: a parametric Nearest Mean classifier (NMC), non-parametric k Nearest Neighbors classifier (kNN) (for which we take $k = 7$), Parzen Window classifier (PWC) and not pruned decision tree (tree), see e.g. [26] for details. In addition to different base classifiers, we run the tests using two alternative distance in space measures: the Euclidean distance and cosine (the details will follow in the next section).

To support the viability of FISH2, we implement and run two peer methods for training set selection under concept drift: Klinkenberg and Joachims [21] (KLI) and Tsymbal et al. [15] (TSY). KLI method tries out a set of different training windows and selects the one showing the best accuracy on the validation data. The most recent training data is chosen as a validation set. TSY method builds a number of classifiers on different sequential training subsets. The final classifier is also selected based on the performance on a validation set. Contrary to a time based selection, used in KLI, the latter method employs distance in space (to the target instance) criterion to select a validation set. Both methods use windows to form the individual classifiers. In contrast, FISH builds individual classifiers using systematic instance selection based combining distance in time and space. The summary of KLI and TSY with the options and interpretations chosen are presented in Appendix A.

The motivation for choosing this peer group is to observe the effect of integrated instance selection (time and space) which is done in FISH. The chosen methods are able to determine training set size using cross validation,

they use no explicit change detection and are base classifier independent and do not require complex parametrization.

We also include a baseline ALL, which uses all the historical data. If the data happens to be stationary, ALL should be the most accurate.

6.2.3 FISH3 and variable time and space ratio

Finally, we compare the performances of FISH1,FISH2 and FISH3 to see what benefits in accuracy are brought by online parameter selection at a cost of increased computational complexity (as compared to FISH1 and FISH2).

We also analyze the progress of the training set size and the proportions of time and space in the distance function over time.

6.3 Implementation Details

For FISH, FISH2, FISH3 and TSY we use the Euclidean distance in space

$$d^E(\mathbf{X}_j, \mathbf{X}_l) = \sqrt{\sum_{i=1}^{p} |\mathbf{x}_j^{(i)} - \mathbf{x}_l^{(i)}|^2}, \tag{5}$$

where $\mathbf{x}_j^{(i)}$ is the i^{th} feature of the instance $\mathbf{X}_j$ and p is the dimensionality.

We also test FISH2 using an alternative cosine distance (inverse similarity) in space

$$d^C(\mathbf{X}_j, \mathbf{X}_l) = \frac{1}{|\cos(\mathbf{X}_j, \mathbf{X}_l)|} = \frac{\sqrt{\sum_{i=1}^{p}(\mathbf{x}_j^{(i)})^2}\sqrt{\sum_{i=1}^{p}(\mathbf{x}_l^{(i)})^2}}{|\sum_{i=1}^{p}\mathbf{x}_j^{(i)}\mathbf{x}_l^{(i)}|}. \tag{6}$$

The features are scaled to the interval $[0, 1]$ before calculating the distance in space. We use linear distance in time, as defined in Equation (2). Distances in time and space are scaled to $d^{(S)}, d^{(\mathcal{T})} \in [0, 1]$ before calculating the proportion $\alpha_1 : \alpha_2$.

We use the following setting for the methods.

- For FISH1 and TSY we use training set size $N = 40$, FISH2, FISH3 and KLI have adaptable set size, ALL has a growing set size.
- KLI and TSY operate in batch mode, we use batch size 15 for both.
- For TSY we set maximum ensemble size to 7, and use 7 nearest neighbors for error estimation.

- The fixed weights proportions of time and space in the distance function for FISH1 and FISH2 are $\alpha_1 : \alpha_2 = 1 : 1$ for all the data. In the first series of experiments we used variable ration of $\alpha_1 : \alpha_2$.

- For FISH2 and FISH3 we took training set sizes for cross validation with a step 5 to speed up the experiments.

- If there are too few instances from one class in a formed sample, the label is assigned according to the major class.

For Ozone, Elec and LU data backward search for FISH2, FISH3, KLI and TSY was limited to 1000 instances to reduce the complexity of the experiment. For testing with the decision tree using Elec and Ozone data we subsampled taking every 5^{th} instance to speed up the experiments.

7 Evaluation

We evaluate FISH2 performance based on the *testing error* and *complexity*. We analyze the progress of the experiments to draw qualitative conclusions.

To evaluate the accuracy, we calculate the ranks of the peer methods. The best method for a given data set is ranked 1, the worst method is ranked 4. The ranks for each data set sum up to 10. An average rank over all the datasets is calculated for each classifier and used as performance measure.

In order to estimate a statistical significance of the differences between the error rates of the methods, for real datasets we used the McNemar [27] paired test, which does not require assumption about i.i.d. origin of the data.

To evaluate the applicability, we calculate the worst case and the average complexity of the six peer methods. We count the number of data passes required to make a classification decision for one observation at time t. The results (approximations) are presented in Table 2. Granularity g is a step of the time and space proportion[3]. We also present the parameters that need to be prefixed in advance for each method.

The run time of FISH2 is reasonable for sequential data, for all five methods it takes up to 1 min for NMC, kNN and PWC and for the decision tree it is $\sim$ 5 times longer to cast a classification decision for *one time observation* on a 1.46 GHz PC, 1GB RAM. For implementation MATLAB 7.5 is used.

[3]The number of options tried out, we use 10. Option 1: $\alpha_1 = 0, \alpha_2 = 1$, option 2: $\alpha_1 = 0.1, \alpha_2 = 0.9, \ldots$, option 10: $\alpha_1 = 1, \alpha_2 = 0$

Table 2: Method complexities. b - batch size; M - ensemble size; k - testing neighborhood size; N - training set size; A - time/space weight; g - granularity of the time and space proportion; t - time since the start of the sequence.

Method	Worst case	Average	Parameters
ALL	t	the same	–
KLI	$\frac{t^2}{2} + \frac{tb}{2}$	$\frac{t^2}{2b} + \frac{t}{2}$	b
TSY	$t(k+1) + N$	$t(k+1) + \frac{N}{b}$	b,M,k,N
FISH1	$t(N+2) + \frac{N(2-N)}{4}$	the same	A,N
FISH2	$\frac{t^2 k}{2} + \frac{t(k+2)}{2}$	the same	A, k
FISH3	$g(\frac{t^2 k}{2} + \frac{t(k+2)}{2})$	the same	k

Finance, biomedical applications are the domains where the data is scarce, imbalanced, while concept drift is very relevant. For example, in bankruptcy prediction an observation might be received once per day or even per week, while the model needs to be constantly updated and economic cycles imply concept drift. In supermarket stock management, stock quantity needs to be predicted once per week, thus only 52 observations are received per year. In such application cases even one hour of the method run for the decision would not be an issue.

8 Results and Discussion

In this section we present and discuss experimental results.

8.1 FISH1 results

We run controlled experiments with FISH1 varying the proportion of time and space contribution in the distance measure. By controlled we mean that we fix the setting except one parameter, which is the proportion of time and space in the distance function. We investigate the effect of the proportion to the testing accuracy on the six real dataset. We allow the proportion $\alpha_1 : \alpha_2$ change from $0:1$ to $1:0$ with a step 0.01.

We use NMC as the base classifier to simplify the setup as much as possible to analyze the effect of the proportions of time and space in the distance function to the testing accuracy.

The testing results for each of the six datasets are provided in Figure 10. We plot the testing accuracy against the proportion of time and space in

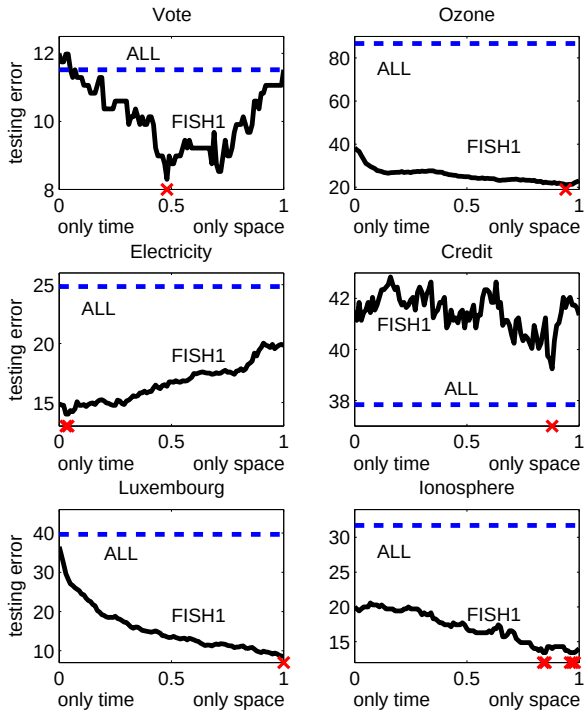

Figure 10: FISH1: testing errors. $*$ on x axis denote minimum error.

the distance function. $\alpha_1 = 0$ means that only the distance in time is used, which corresponds to a training window of a fixed size. $\alpha_1 = 1$ (implies $\alpha_2 = 0$) means that only the distance in space is used. All the values in between indicate different proportions of time and space in the distance function in training set selection.

In Table 3 we provide the numerical results using a step 0.1 for α_1 values. Although the primary purpose of the experiment is to analyze the relation between the proportions of time and space in the distance function and accuracy, we also indicate statistical significance of the difference between FISH1 and the baseline ALL using McNemar test.

The results both in Figure 10 and in the table show that already using a primitive fixed size N technique the best accuracy is achieved in combination of distances in time and space. The minimum error is heavily shifted towards distance in space in most of the datasets ('Ozon','Cred','Iono' and 'LU'),

Table 3: Testing errors. The best accuracy for each column is underlined. Symbol '•' indicates that the method performed significantly better than ALL, '∘' indicates that the method performed significantly worse than ALL, and '−' indicates no difference (at $\alpha = 0.05$).

	space wght. α_1	time wght. α_2	Luxe	Ozon	Elec	Cred	Vote	Iono
FISH1	0	1	36.53•	38.29•	14.86•	41.54∘	11.98−	20.00•
FISH1	0.1	0.9	24.42•	27.75•	<u>14.75</u>•	41.74∘	11.29−	20.29•
FISH1	0.2	0.8	19.00•	26.92•	15.13•	41.94∘	10.37−	19.43•
FISH1	0.3	0.7	17.21•	27.48•	15.13•	42.24∘	10.60−	18.86•
FISH1	0.4	0.6	14.84•	26.29•	15.91•	42.04∘	10.14−	17.43•
FISH1	0.5	0.5	13.42•	24.91•	16.75•	41.44∘	<u>8.76</u>•	16.86•
FISH1	0.6	0.4	12.79•	24.04•	17.46•	42.24∘	9.22−	16.57•
FISH1	0.7	0.3	11.74•	23.69•	17.53•	40.54∘	<u>8.76</u>−	16.57•
FISH1	0.8	0.2	10.95•	23.25•	17.70•	40.94−	9.91−	<u>14.00</u>•
FISH1	0.9	0.1	9.53•	<u>21.71</u>•	19.66•	40.94−	10.83−	14.29•
FISH1	1	0	<u>8.58</u>•	23.02•	19.83•	41.34∘	11.52−	<u>14.00</u>•
ALL			39.68	86.70	24.84	<u>37.84</u>	11.52	31.71

which is explainable by the data origin. The datasets were picked expecting heterogeneous structure in space and also to have a temporal order. 'LU' data has it's minimum testing error at the very extreme distance in space proportion (time proportion is 0), which we could observe in the data plot in Figure 9. Visually, the time order in 'LU' data is weak. On the contrary, 'Elec' data has visually strong time order and that correlates with the testing results. One can observe in Figure 10 that the minimum testing error for 'Elec' is close to the time proportion extreme. It suggests the training based on windowing would be preferable. Change detection technique, which is in principle variable sized windowing, was applied by Gama et al [28], who introduced this data set for concept drift problems.

The dashed lines in Figure 10 indicate the baseline testing error, which is achieved by using full history as a training set (ALL). The primitive FISH1 using a fixed training size already outperforms ALL in five out of six datasets. Absolute error in 'Ozon' is so high since the classes are heavily unbalanced (major class makes 94%). In 'Cred' FISH1 is worse than ALL in all the time and space proportions. It suggests that either the data is stationary, or the fixed size of the training set is very much non optimal.

Table 4: FISH2: testing errors, Euclidean distance in space. The best accuracy for each column is underlined. Symbol '•' indicates that the method performed significantly worse than FISH2, '∘' indicates that the method performed significantly better than FISH2, and '−' indicates no difference (at $\alpha = 0.05$).

	base	Luxe	Ozon	Elec	Cred	Vote	Iono	RANK
FISH2	NMC	11.89	34.31	15.16	36.94	8.53	17.43	*1.33*
KLI		30.89•	22.90∘	19.97•	36.24−	11.29•	21.71•	*2.08*
TSY		35.89•	37.23•	15.47−	40.64•	11.29•	20.57−	*2.75*
ALL		39.68•	86.70•	24.84•	37.84−	11.52•	31.71•	*3.83*
FISH2	kNN	14.63	7.03	15.06	30.13	8.76	22.00	*2.08*
KLI		15.74−	7.11−	18.98•	30.03−	9.68−	22.86−	*3.00*
TSY		28.79•	7.03−	13.16∘	31.43−	10.60−	23.14−	*3.25*
ALL		11.84∘	6.99−	19.86•	28.83−	8.29−	22.29−	*1.67*
FISH2	PWC	12.37	70.79	41.08	34.33	8.99	12.86	*1.75*
KLI		14.42•	38.81∘	46.06•	34.63−	10.37−	15.14•	*3.00*
TSY		26.42•	54.72∘	43.62•	36.54−	9.68−	19.71•	*3.25*
ALL		11.68−	84.88•	43.62•	34.43−	8.53−	12.86−	*2.00*
FISH2	tree	0.37	9.99	13.54	31.03	7.37	18.00	*1.42*
KLI		0.37−	11.69•	17.36•	36.34•	9.68−	20.86−	*3.25*
TSY		0.37−	12.63•	8.97∘	37.04•	10.14−	20.29−	*3.08*
ALL		0.37−	10.50−	16.99•	32.83−	7.83−	18.57−	*2.25*

The training set size was chosen to be equal for all the data sets to keep the settings uniform and the results comparable. Next we look at the results when the training size is learnable online.

8.2 FISH2 results

We test FISH2 along with two peer methods KLI and TSY as well as the baseline ALL. We test using four alternative base classifiers and two alternative distance in space measures for the six datasets. Thus all in all we run $4 \times 2 \times 6 = 48$ experiments for each of the methods. The results are provided in Tables 4 and 5. We use McNemar paired test to estimate the statistical significance of the difference between FISH2 and the peers.

The five methods were ranked as presented in Section 7 with respect to each data set, and then the ranks were averaged (last column in Table 4).

FISH2 has the best rank by a large margin with NMC and tree classifiers, for kNN and PWC either FISH2 or ALL prevails, depending on the distance measure. The final scores averaged over all four base classifiers and two alternative distance measures are: 1.68 for FISH2, 2.83 for KLI, 3.07 for

Table 5: FISH2: testing errors, cosine distance in space. The best accuracy for each column is underlined. Symbol '•' indicates that the method performed significantly worse than FISH2, '∘' indicates that the method performed significantly better than FISH2, and '−' indicates no difference (at $\alpha = 0.05$).

	base	Luxe	Ozon	Elec	Cred	Vote	Iono	RANK
FISH2		12.68	35.25	15.57	38.14	8.76	16.86	*1.67*
KLI	NMC	30.89•	22.90∘	19.97•	36.24−	11.29•	21.71•	*2.08*
TSY		35.89•	37.23−	15.47−	40.64−	11.29•	20.57−	*2.58*
ALL		39.68•	86.70•	24.84•	37.84−	11.52•	31.71•	*3.67*
FISH2		14.79	6.99	15.19	29.93	8.53	21.71	*1.75*
KLI	kNN	15.74−	7.11−	18.98•	30.03−	9.68−	22.86−	*3.17*
TSY		28.79•	7.03−	13.16∘	31.43−	10.60•	23.14−	*3.33*
ALL		11.84∘	6.99−	19.86•	28.83−	8.29−	22.29−	*1.75*
FISH2		12.68	72.25	39.26	34.83	8.29	13.34	*2.00*
KLI	PWC	14.42•	38.81∘	46.06•	34.63−	10.37•	15.14−	*2.83*
TSY		26.42•	54.72∘	43.62•	36.54−	9.68−	19.71•	*3.25*
ALL		11.68∘	84.88•	43.62•	34.43−	8.53−	12.86−	*1.92*
FISH2		0.37	10.03	12.79	31.34	7.60−	17.71	*1.71*
KLI	tree	0.37−	11.69•	17.36•	36.34•	9.68−	20.86−	*2.83*
TSY		0.37−	12.63•	8.97∘	37.04•	10.14−	20.29−	*3.06*
ALL		0.37−	10.50−	16.99•	32.83−	7.83−	18.59−	*2.40*

TSY and 2.42 for ALL.

Using kNN, PWC and tree as a base classifier, ALL method outperform TSY and KLI according to the rank score. It implies, that under this setting there would be little point in employing those concept drift responsive methods and increasing complexity, as simple retraining (ALL) would do well. The results in favor of FISH2 are significant in about half of the cases. Some of the results indicate no statistical difference, however, it should be taken into account that the datasets are not large and the test is non parametric.

FISH2 method is designed to work where concept drift is not clearly expressed. These are the situations of gradual drift, reoccurring concepts. ALL method outperforms all the drift responsive methods but not FISH2 with kNN as a base classifier.

Windowing methods work well on Elec data, because the drifts in this data are more sudden. Elec data shows the biggest need for concept drift adaptive methods, because for these datasets ALL method performs relatively the worst from the peer group.

The credits for FISH2 performance in the peer group shall be given to similarity based training set selection. KLI addresses only similarity in time (training window). TSY uses only similarity in time for classifier training, but then they use similarity in space for classifier selection. We employ a combination of distance in time and space already in classifier building phase.

It is interesting to look why FISH2 outperforms KLI and TSY in terms of accuracy even on the datasets demanding training windows. This is because FISH2 uses adaptive validation set as compared to KLI and variable training set size as compared to TSY.

In FISH2 experiments we fixed equal proportion of time and space in the distance function $\alpha_1 : \alpha_2 = 1 : 1$, in order to have uniform comparable setups for all the datasets. In the next section we look if FISH2 results can be improved by allowing the time and space proportion to be learnable online.

8.3 FISH3 results

FISH3 implements variable training set size and variable proportions of time and space in the distance function, both are learnable online. Recall, that we had different time and space proportions in FISH1 experiments. However, in FISH1 the proportion is fixed for all the experiment. In FISH3 we have a variable proportion for every time step and it is learnable online.

In Table 6 we compare the accuracies of the three FISH methods using

Table 6: FISH3: variable proportion of time and space. The best accuracy for each column is underlined.

	Luxe	Ozon	Elec	Cred	Vote	Iono
FISH1	14.63	<u>26.49</u>	15.74	41.44	8.76	16.86
FISH2	11.89	34.31	15.16	36.94	<u>8.53</u>	17.43
FISH3	<u>10.53</u>	37.47	<u>13.77</u>	<u>36.34</u>	<u>8.53</u>	<u>15.43</u>
mean α_1	0.77	0.62	0.41	0.46	0.32	0.52
ALL	39.68	86.70	24.84	37.84	11.52	31.71

simple settings: NMC classifier and Euclidean distance in space. We use the same fixed proportion of time and space as before for both FISH1 and FISH2 ($\alpha_1 : \alpha_2 = 1 : 1$).

FISH3 has the best accuracy in all cases except for Ozone data, which is very highly imbalanced. We include a baseline ALL to verify if our concept drift responsive methods make sense. The differences between ALL and FISH accuracies are statistically significant everywhere except in Credit data.

It might be argued that improvement in accuracy shown by FISH3 as compared to FISH2 is marginal. In fact, the differences between FISH2 and FISH3 are statistically significant in three out of six datasets (Luxembourg, Ozone and Electricity) which are more than twice longer than the remaining ones, besides they have a natural temporal order, while the remaining three have assumed temporal order.

Let us look at the time and space proportion. In Table 6 we provide averaged space proportion (mean α_1). Luxembourg and Ozone datasets are inclined towards distance in space, while Vote and Electricity data shows preference to distance in time. That is not fully consistent with the observations in FISH1 experiments, Section 8.1. Note that in FISH1 experiments the time and space proportion was fixed for all the run on a dataset, while here we allowed the proportion to vary every time step. In Figure 11 we plot the progress of the time and space proportion for all six datasets. The line is smoothed using a moving average of 5 to emphasize the tendencies against individual peaks.

It can be concluded that if the domain allows increased complexity variable training set size and variable time and space proportions are worth applying to gradually drifting datasets.

The FISH family of methods should be regarded as an extension to existing techniques. It emphasizes that time and space relations are not

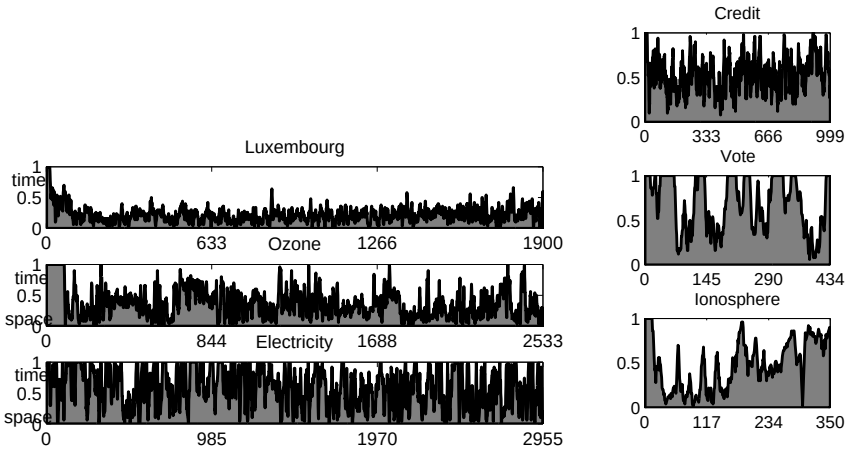

Figure 11: Progress of the time and space proportions in FISH3.

discrete, but can be viewed in a continuous space.

9 Conclusion

We formulated a concept of similarity in time and space in for adaptive training set selection. It leads to a range of training set selection strategies from based on training window based to instance selection in space.

Based on the formulated concept we developed a family of methods for training set selection under concept drift. FISH1 uses a preset proportion of time and space in the distance function and preset training set size. FISH2 learns the training set size online at every time step using cross validation on the historical data. FISH3 learns online both the training set size and the proportions of time and space in the distance function.

With FISH1 we demonstrate that for a gradually drifting data combination of distances in time and space can lead to a better classification accuracy than using a single technique.

FISH2 shows the best accuracy in the peer group on the datasets exhibiting gradual drifts and a mixture of several concepts. The method complexity is reasonable for field applications.

FISH3 demonstrates that the proportion of time and space in the distance function is learnable online.

We show the advantages of combination of the distances in time and space. Combining time and space for training instance selection contributes

to improvement of classifier generalization performance under gradual concept drift, since this way heterogeneous nature of the drifting data can be captured.

References

[1] G. Hulten, L. Spencer, P. Domingos, Mining time-changing data streams, in: KDD '01: Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 97–106.

[2] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine Learning 23 (1) (1996) 69–101.

[3] J. Beringer, E. Hullermeier, Efficient instance-based learning on data streams, Intelligent Data Analysis 11 (6) (2007) 627–650.

[4] R. Klinkenberg, Learning drifting concepts: Example selection vs. example weighting, Intelligent Data Analysis 8 (3) (2004) 281–300.

[5] B. Wenerstrom, C. Giraud-Carrier, Temporal data mining in dynamic feature spaces, in: ICDM '06: Proceedings of the 6th international conference on Data Mining, IEEE Computer Society, 2006, pp. 1141–1145.

[6] I. Žliobaitė, Combining time and space similarity for small size learning under concept drift, in: ISMIS '09: Proceedings of the 18th international symposium on Methodologies for Intelligent Systems, Vol. 5722 of LNCS, 2009, pp. 412–421.

[7] D. Aha, D. Kibler, Instance-based learning algorithms, in: Machine Learning, 1991, pp. 37–66.

[8] M. Harries, Splice-2 comparative evaluation: Electricity pricing, technical report, The University of South Wales (1999).

[9] D. Lin, An information-theoretic definition of similarity, in: ICML '98: Proceedings of the 15th international conference on Machine Learning, Morgan Kaufmann Publishers, 1998, pp. 296–304.

[10] C. Aggarwal, Towards systematic design of distance functions for data mining applications, in: KDD '03: Proceedings of the 9th ACM

SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 9–18.

[11] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[12] Y. Law, C. Zaniolo, An adaptive nearest neighbor classification algorithm for data streams, in: PKDD '05: Proceedings of the 9th European conference on Principles and Practice of Knowledge Discovery in Databases, Vol. 3721 of LNCS, Springer, 2005, pp. 108–120.

[13] V. Ganti, J. Gehrke, R. Ramakrishnan, Mining data streams under block evolution, SIGKDD Exploration Newsletter 3 (2) (2002) 1–10.

[14] M. Lazarescu, S. Venkatesh, Using selective memory to track concept effectively, in: Proceedings of international conference on Intelligent Systems and Control, 2003, pp. 14–20.

[15] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Information Fusion 9 (1) (2008) 56–68.

[16] H. Valizadegan, P. Tan, A prototype-driven framework for change detection in data stream classification, in: CIDM '07: Proceedings of IEEE symposium on Computational Intelligence and Data Mining, 2007, pp. 88 – 95.

[17] I. Katakis, G. Tsoumakas, I. Vlahavas, Tracking recurring contexts using ensemble classifiers: an application to email filtering, Knowledge and Information Systems 22 (3) (2010) 371–391.

[18] K. Ueno, X. Xi, E. Keogh, D. Lee, Anytime classification using the nearest neighbor algorithm with applications to stream mining, in: ICDM '06: Proceedings of the 6th international conference on Data Mining, IEEE Computer Society, 2006, pp. 623–632.

[19] M. Black, R. J. Hickey, Maintaining the performance of a learned classifier under concept drift, Intelligent Data Analysis 3 (6) (1999) 453–474.

[20] C. Atkeson, A. Moore, S. Schaal, Locally weighted learning, Artificial Intelligence Review 11 (1-5) (1997) 11–73.

[21] R. Klinkenberg, T. Joachims, Detecting concept drift with support vector machines, in: ICML '00: Proceedings of the 17th international conference on Machine Learning, Morgan Kaufmann Publishers Inc., 2000, pp. 487–494.

[22] R. Jowell, the Central Co-ordinating Team, European social survey : 2002/2003, Technical report, London: Centre for Comparative Social Surveys, City University (2003).

[23] R. Jowell, the Central Co-ordinating Team, European social survey : 2004/2005, Technical report, London: Centre for Comparative Social Surveys, City University (2005).

[24] R. Jowell, the Central Co-ordinating Team, European social survey : 2006/2007, Technical report, London: Centre for Comparative Social Surveys, City University (2007).

[25] A. Asuncion, D. Newman, Uci machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007).
URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[26] R. Duda, P. Hart, D. Stork, Pattern Classification (2nd Edition), Wiley-Interscience, 2000.

[27] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.

[28] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: SBIA '04: Proceedings of the 17th Brazilian symposium on Artificial Intelligence (Advances In Artificial Intelligence), Vol. 3171 of LNAI, Springer, 2004, pp. 286–295.

[29] R. Klinkenberg, I. Renz, Adaptive information filtering: Learning drifting concepts, in: Proceedings of AAAI-98/ICML-98 workshop Learning for Text Categorization, 1998, pp. 33–40.

All history training set (ALL)

INPUT
Data: historical instances $\mathbf{X^H} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ with labels $\mathbf{y^H}$, target instance $\mathbf{X}_{t+1}$ without a label. Base learner type: $\mathcal{L}$.
ALGORITHM
Train the classifier $\mathcal{L}_t$ using a training set $\mathbf{X_t^T} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$. **OUTPUT**
Trained classifier $\mathcal{L}_t$ to be applied to the testing instance $\mathbf{X}_{t+1}$.

Figure 12: All history training set (ALL).

A Peer Methods

In this Appendix we present pseudo codes and the settings used for the peer methods, which we implemented and used in experimental evaluation.

In Figure 12 we provide a pseudo code for the incremental growing window, which is used as a baseline.

Klinkenberg and Rentz [29] method is presented in Figure 13. The original work used Support Vector Machines (SVM) as the base classifier. We use the method with different base classifiers.

Tsymbal et al [15] method is presented in Figure 14.

WINDOW SELECTION METHOD (KLI)

INPUT
Data: historical instances $\mathbf{X^H} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ with labels $\mathbf{y^H}$, target instance $\mathbf{X}_{t+1}$ without a label.
Parameters: batch size m. Base learner type: $\mathcal{L}$.
ALGORITHM

1. *For* $j = 1$ *to* t (all the batches),
   (a) *for* $k = 1$ *to* m (all the instances in the last batch),
      i. build a classifier on the data in $\{\mathbf{X}_j, \ldots, \mathbf{X}_t\}$, using cross-validation,
      ii. test the classifier on the excluded instance,
      if correctly classified $e_k = 0$, *else* $e_k = 1$,
   (b) calculate the error $E_j = \frac{1}{m} \sum_{k=1}^{m} e_k$ for past window of size $w_j = j \times m$.
2. Find minimum error $j^* = \arg\min_{j=1}^{t} E_j$.

OUTPUT
The index j^* to form a training set $\mathbf{X_t^T} = (\mathbf{X}_{j*}, \ldots, \mathbf{X}_t)$.

Figure 13: Window Selection method (KLI).

Dynamic ensemble method (TSY)

INPUT
Data: historical instances $\mathbf{X^H} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ with labels $\mathbf{y^H}$, target instance $\mathbf{X}_{t+1}$ without a label.
Parameters: training set size N, batch size m, maximal ensemble size M, neighborhood size k.
Base learner type: $\mathcal{L}$.
ALGORITHM

1. Build classifier $\mathcal{L}_t$ using $\mathbf{X}_{t-N+1}, \ldots, \mathbf{X}_t$ labeled instances.
2. *If* ensemble size $< M$ *then* add $\mathcal{L}_t$ to the ensemble, *else* replace the ensemble member which showed the largest error on the latest batch.
3. Find k nearest neighbors to $\mathbf{X}_{t+1}$: $\{\mathbf{X}_{z1}, \ldots, \mathbf{X}_{zk}\}$ using Heterogeneous Euclidean overlap measure d^H, which is the Euclidean distance with normalized features.
4. *For* $j = 1$ *to* M, calculate weights (for each ensemble member $\mathcal{L}_1, \ldots, \mathcal{L}_M$)
   $w(\mathcal{L}_j) = \frac{\sum_{s=1}^{k} (\frac{1}{d^H(\mathbf{X}_{t+1}, \mathbf{X}_{zs})} r_j(\mathbf{X}_{zs}))}{\sum_{s=1}^{k} \frac{1}{d^H(\mathbf{X}_{t+1}, \mathbf{X}_{zs})}}$,
   where *if* $\mathcal{L}_j$ is correct in predicting the label of $\mathbf{X}_{zs}$
   then $r_j(\mathbf{X}_{zs}) = 1$, *else* $r_j(\mathbf{X}_{zs}) = -1$.
5. Select $\mathcal{L}^* = \mathcal{L}_j*$, where $j^* = \arg\max_{j=1}^{M} w(\mathcal{L}_j)$.

OUTPUT
Classifier $\mathcal{L}^*$ for decision making.

Figure 14: Dynamic ensemble method (TSY).