Metadata of The Datasets

For the experiments in the MSDT algorithm, we have selected a total of 4 datasets. A summary of the datasets and their properties is shown in Table 1. A training set and a test set were generated from the dataset of each experiment. The "*sequences*" attribute of all the datasets are multi-valued. And, the number of the attribute value is in the range of 1 to 5. Its value is a set of item-sequences, represented as itineraries of a tourist. And, each item of an itinerary is a scenic spot, numbered from 1 to 5. And, the sequential-label attribute is derived from the "*sequences*" attribute, the content of which is the representative sequential pattern, mined by the VMSP4MSDT algorithm.

Table 1: A summary of the datasets and their properties

Dataset Name	**Content**	**N**	**Input attr.**	**The sequences attr.**	**The sequential-label attr.**	**Size of sliding windows (α)**	**Experiment**
Tourist	the baseline small dataset	100	11	1	1	5	Experiment I.1
CDNow-RFM	the other two small datasets	2,357	7	1	1	2..3	Experiment I.2
msdt2-multi-valued	the three large multi-valued datasets	20,000	9	1	1	3..5	Experiment II Experiment III.1
msdt2-single-valued	the three large single-valued datasets	20,000	6	1	1	3..5	Experiment III.2

Each metadata of the datasets is described as follows.

I. The Tourist dataset

The Tourist dataset with eleven predictor attributes, one attribute with sequences, and one class-label attribute containing a sequential-label, are shown in Table I-1. The number of various sequential-labels is 40. Except the attributes, *occupation* and *preference*, are multi-valued, the other attributes are single-valued.

Table I-1: Description of attributes of data in Experiment I.1

Attribute	***Description***	***Number of values***	***Value***
salary	Salary level	1	1 to 5
gender	Gender	1	1 or 2
age	Age	1	1 to 100
transportation	Transportation type of tour	1	1 to 3
preferences	Preferences of personal tour	1 to 5	1 to 20
hvalue	Own a house?	1	1 or 2
elevel	Education level	1	1 to 5
mstatus	Marital status	1	1 to 3
occupation	Occupation	1 to 2	1 to 10
blood-type	Blood type	1	1 to 4
tour-frequency	Type of tour frequency per year	1	1 to 3
sequences	A set of sequences	1 to 5	a sequence of 1 to 5
class-label	A sequential-label	1	a sequence of 1 to 5

II. The CDNow-RFM dataset

With respect to the CDNow-RFM dataset, as we have two sliding windows with size from 2 to 3 and two corresponding datasets, each of which is named *CDNow-RFM-alpha-X*, where *X* is the window size. Each CDNow-RFM dataset has ten attributes, including customer id, seven predictor attributes with single-values, one attribute containing sequences and one class-label attribute containing a sequential-label. The number of various sequential-labels is 12 when windows size = 2; and it is 33 when windows size = 3. The predictor attributes are extracted from the RFM properties, including recency, avg_frequency, sum_frequency, avg_quantity, sum_quantity, avg_amount and sum_amount. We extract features and sequences for each customer from the CDNow_sample dataset in the following three steps. First, as CDNow_sample

contains the entire purchase history during 6 quarters, from the first quarter of 1997 to the end of the second quarter of 1998, we extract features by summarizing one's RFM properties (Recency, Frequency and Monetary value) per quarter. About how to create an RFM summary can be referred to http://brucehardie.com/notes/022/. Second, we calculate the quarterly purchasing percentage of each customer by normalizing one's purchasing amount (i.e. the extracted Monetary value) based on the customers in the same quarter, and then assign each customer to an ABC class according to the percentage. Third, we get the features of each customer by further summarizing all the RFM properties per two quarters (i.e. within the sliding windows, $\alpha = 2$) and per three quarters (i.e. $\alpha = 3$) respectively during the six quarters; and we also get the sequences of each customer by further collecting a sequence of the ABC classes per two quarters and per three quarters during the six quarters. As the size of the training sets of the two real-life sets are small, this may result in bias and variance in the estimate.

III. The msdt2-multi-valued dataset

The dataset is a large dataset of tourists with nine predictor attributes, one attribute with sequences, and one class-label attribute containing a sequential-label, are shown in Table III-1. The large dataset has the same features and values as the dataset used by both the multi-valued and multi-labeled algorithms of MMC and MMDT, except it has two additional attributes, one for item-sequences and the other for sequential-label. And, some features of the dataset are multi-valued. Except the predictor attributes, *car*, *hobby* and *occupation* are multi-valued, the other predictor attributes are single-valued. To learn a classifier from incrementally-added sequences of training instances within some successive sliding windows per classification lifecycle, we set the user-specified parameters that the sliding window sizes, $\alpha = 3..5$. Therefore, the value of the "*sequences*" attribute is a set of item-sequences with lengths from 3 to 5, which correspond three most recent sliding windows respectively with incrementally-added sizes. The item-sequences of each record are generated by the combination of the following five functions as:

(a) FunctionA(each data record, sequence length) = <1 2 3> for 3-sequence; <1 2 3 2> for 4-sequence; <1 2 3 2 3> for 5-sequence, if the data record satisfies the following condition:

$[(\text{gender} = 1) \wedge (20{,}000 \leq \text{salary} \leq 100{,}000) \wedge (\text{car} \in [1..10])] \vee$
$[(\text{gender} = 2) \wedge (100{,}000 \leq \text{salary} \leq 150{,}000) \wedge (\text{car} \in [11..20])] \vee$
$[\text{occupation} \in [1, 6]]$

(b) FunctionB(each data record, sequence length) = <2 4 3> for 3-sequence; <2 4 3 5> for 4-sequence; <2 4 3 5 3> for 5-sequence, if the data record satisfies the following condition:

[(age < 40) ∧ (car ∈ [1..10])] ∨
[(40 ≤ age < 60) ∧ (car ∈ [11..15])] ∨
[occupation ∈ [2]]

(c) FunctionC(each data record, sequence length) = <3 2 1> for 3-sequence; <3 2 1 4> for 4-sequence; <3 2 1 4 2> for 5-sequence, if the data record satisfies the following condition:

[(mstatus = 1) ∧ (20,000 ≤ salary ≤ 100,000) ∧ (hobby ∈ [1..9] ∨ car =10)] ∨
[(mstatus ∈ [2, 3]) ∧ (100,000 ≤ salary ≤ 150,000) ∧ (hobby ∈ [11..19] ∨ car = 20)] ∨ [occupation ∈ [3..5]]

(d) FunctionD(each data record, sequence length) = <5 4 2> for 3-sequence; <5 4 2 1> for 4-sequence; <5 4 2 1 2> for 5-sequence, if the data record satisfies the following condition:

[(mstatus = 1) ∧ (20,000 ≤ salary ≤ 40,000)] ∨
[(mstatus = 2) ∧ (40,000 < salary ≤ 80,000)] ∨
[occupation ∈ [7..8]]

(e) FunctionE(each data record, sequence length) = <3 1 5> for 3-sequence; <3 1 5 4> for 4-sequence; <3 1 5 4 1> for 5-sequence, if the data record satisfies the following condition:

[(elevel ∈ [1]) ∧ (hobby ∈ [1..9]) ∧ (mstatus = 1)] ∨
[(elevel ∈ [5]) ∧ (hobby ∈ [10..20]) ∧ (mstatus ∈ [2..3])] ∨
[occupation ∈ [9..10]]

In the above five functions, as a record satisfies any condition of a function, the corresponding sequence with respective window size will be assigned to it. Certainly, a record may satisfy several conditions of the five functions that produce multiple sequences. And then, we apply the VMSP4MSDT algorithm to select the representative sequential pattern from the multiple sequences for each record. Finally, we assign the maximal sequential pattern (i.e., the *sequential-label*) to be the class label of each record. The number of various sequential-labels is 10 while $\alpha = 3$; 13 while $\alpha = 4$; and 10 while $\alpha = 5$.

The principle of designing these five functions is that the length of their maximal sequential patterns should not be longer than 2. In other words, the least overlapping patterns will be the better sequences. Take the 3-sequence as an example, we choose from its first character to its third character randomly. Thus, we assign the sequences <1 2 3>, <2 4 3>, <3 2 1>, <5 4 2> and <3 1 5> for the above five functions respectively. Suppose the minimum support rate is 50%, the maximal sequential patterns of the five sequences are <1>, <2> and <3>. They are all only 1-sequences. Likewise, as for 4-sequences and 5-sequences, they are done in the same way.

Therefore, we have three large enough datasets of 3-sequence, 4-sequence and 5-

sequence, each with 20,000 records. Each dataset is named *touristXX* in the corresponding sliding window with size from 3 to 5, such that *XX* is from "l3" to "l5". To execute holdout-validation, we have generated 8 training-sets for each *touristXX* of the three large datasets. All the experiments use 5,000 records as the test dataset. The size of 8 training-set are 90, 1,000, 4,000, 6,000, 8,000, 10,000, 12,000 and 14,000 records respectively.

Table III-1: Description of data attributes in Experiment II

Attribute	***Description***	***Number of values***	***Value***
salary	Salary	1	$20,000 to $150,000
gender	Gender	1	1 to 2
age	Age	1	20 to 80
car	Car maker	1 to 3	1 to 20
hobby	Hobby	1 to 5	1 to 20
hvalue	House Value	1	$50,000 to $150,000
elevel	Education level	1	1 to 5
mstatus	Marital status	1	1 to 3
occupation	Occupation	1 to 2	1 to 10
sequences	A set of item-sequences	1 to 5	a sequence of 1 to 5 from functionA to functionE
class-label	A sequential-label	1	a sequence of 1 to 5

Except the two attributes, sequences and class-label, the values of other attributes are uniformly chosen.

IV. The msdt2-single-valued dataset

We remove all the multi-valued features of the msdt2-multi-valued dataset in Table III-1. And, we name it as the large single-valued dataset. Besides, we should further revise the five functions mentioned above, because the multi-valued features constitute some of the conditions of those functions. The number of various sequential-labels is 47 while $\alpha = 3$; 243 while $\alpha = 4$; and 878 while $\alpha = 5$. Next, the revised five functions are described as:

(a) FunctionA(each data record, sequence length) = <1 2 e_{a1}> for 3-sequence, where e_{a1} is randomly chosen from 1 to 5; <1 2 e_{a1} e_{a2}> for 4-sequence, after fixing e_{a1} in which

e_{a2} is randomly chosen from 1 to 5; <1 2 e_{a1} e_{a2} e_{a3} > for 5-sequence, after fixing e_{a1} and e_{a2} in which e_{a3} is randomly chosen from 1 to 5; if the data record satisfies the following condition:

[(gender = 1) ∧ (20,000 ≤ salary ≤ 100,000)] ∨
[(gender = 2) ∧ (100,000 ≤ salary ≤ 150,000)] ∨
[elevel ∈ [3]]

(b) FunctionB(each data record, sequence length) = <2 4 3> for 3-sequence; <2 4 3 5> for 4-sequence; <2 4 3 5 3> for 5-sequence, if the data record satisfies the following condition:

[(age < 40) ∧ (20,000 ≤ salary ≤ 100,000)] ∨
[(40 ≤ age < 60) ∧ (100,000 ≤ salary ≤ 150,000)] ∨
[elevel ∈ [1]]

(c) FunctionC(each data record, sequence length) = <3 2 1> for 3-sequence; <3 2 1 4> for 4-sequence; <3 2 1 4 2> for 5-sequence, if the data record satisfies the following condition:

[(mstatus = 1) ∧ (20,000 ≤ salary ≤ 100,000) ∧ (50,000 ≤ hvalue ≤ 100,000)] ∨
[(mstatus ∈ [2, 3]) ∧ (100,000 ≤ salary ≤ 150,000) ∧(100,000 ≤ hvalue ≤ 150,000)] ∨ [elevel ∈ [4..5]]

(d) FunctionD(each data record, sequence length) = <5 4 2> for 3-sequence; <5 4 2 1> for 4-sequence; <5 4 2 1 2> for 5-sequence, if the data record satisfies the following condition:

[(mstatus = 1) ∧ (20,000 ≤ salary ≤ 40,000)] ∨
[(mstatus = 2) ∧ (40,000 < salary ≤ 80,000)] ∨
[elevel ∈ [2..3]]

(e) FunctionE(each data record, sequence length) = <3 1 5> for 3-sequence; <3 1 5 4> for 4-sequence; <3 1 5 4 1> for 5-sequence, if the data record satisfies the following condition:

[(elevel ∈ [1]) ∧ (gender = 1) ∧ (mstatus = 1)] ∨
[(elevel ∈ [5]) ∧ (gender = 2) ∧ (mstatus ∈ [2..3])] ∨
[(45 ≤ age < 60)]