

Overlapping classification for autocoding system

Sep. 2018 / uRos2018

Yukako Toko^{*1}, Shinya Iijima^{*1}, Mika Sato-Ilic^{*1,2}

^{*1}National Statistics Center, Japan

^{*2}University of Tsukuba

Contents

1. Overview
2. Method
3. Experiments and results
4. Summary



1. Overview

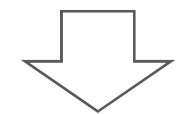
1. Overview – Coding?

Example the Family Income and Expenditure Survey

Ex.) Survey form

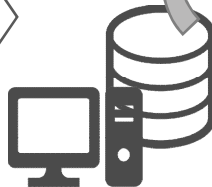
1. Cash Receipts and Cash Disbursements

(1) Kind of Receipts / Purchased Items and Their Uses	(2) Cash Receipts Yen	(3) Quantities	Unit	(4) Cash Disbursements Yen
Eating out (Pizza)		2	Persons	3240
Green tea leaves		100	g	1080
Bonito		500	g	500
Pork		360	g	480



coding

Bonito
->174
Pork
->221
...




ID	Item code	Quantity	JPY
1	396	2	3240
2	380	100	1080
3	174	500	500
4	221	360	480

1. Overview – Background

Originally developed multiclass classifier

- *Non-overlapping (exclusive classification)
- *Probability-based
- *High accuracy

But... yield a certain volume of unmatched output

- *semantic problem
 - *interpretation problem
 - *insufficiency detailed input information
- 

1. Overview – Background

To address those issues... Introduced the idea of **partition coefficient** & **partition entropy** considering the classification status of each object (or feature)

→ representing the uncertainty situation of classification of each object (or feature)

But... it still has problems when classifying objects (or feature) to exclusive classes

Main reason is ... unrealistic restriction



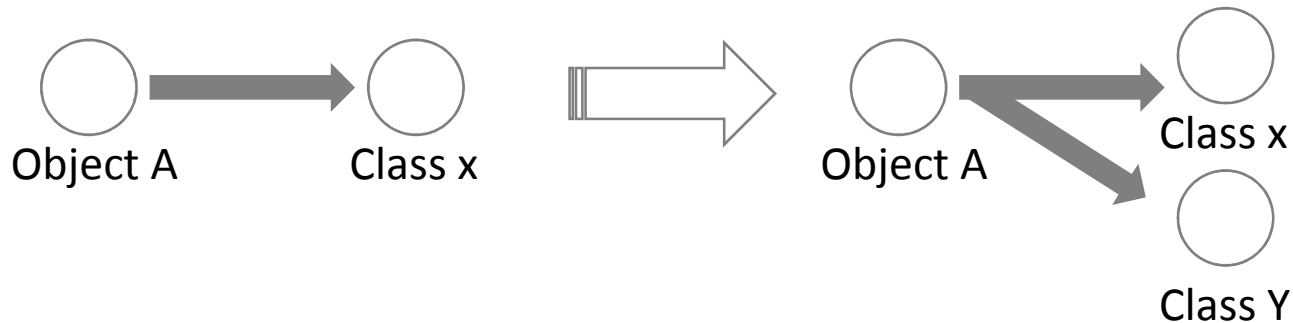
one object is classified to a single class

1. Overview – Purpose

* Develop a new algorithm for **overlapping classification**

→ allows that one object is assigned to multiple classes

→ utilize the idea of our previously proposed classifier considering the classification status of each object



* Define a **new reliability score**

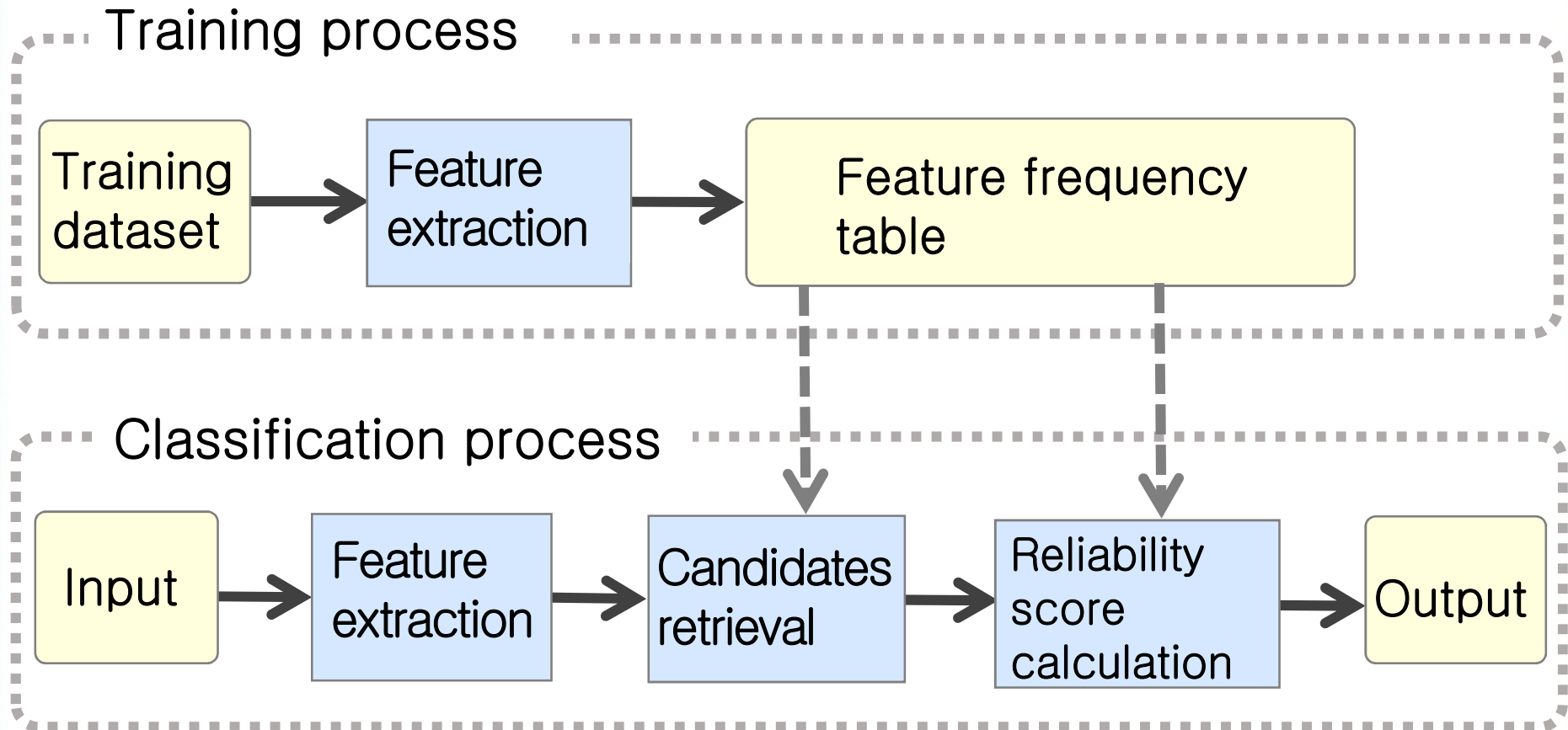
→ assist a user in the assignment of an object to codes

→ utilize the idea of partition entropy as weights of the score

2. Method



2. Method – Structure



2. Method – Algorithm Training process

Example of training data

Chocolate cream pie : 345 (other confectionaries)

text description

classification code

Step 1: **Tokenize** → *chocolate, cream, pie*

Step 2: **word-level N-gram** ($N=1,2$) & entire sentence

→ *uni-gram : chocolate, cream, pie*

bi-gram : chocolate + cream, cream + pie

entire sentence : chocolate + cream + pie

Step 3: **Feature frequency table**

ex.)

<i>feature</i>	<i>code</i>	<i>count</i>
<i>chocolate</i>	<i>345</i>	<i>2</i>
<i>chocolate</i>	<i>352</i>	<i>10</i>
<i>cream</i>	<i>345</i>	<i>6</i>
<i>pie</i>	<i>345</i>	<i>32</i>
<i>pie</i>	<i>376</i>	<i>57</i>
<i>chocolate+cream</i>	<i>345</i>	<i>2</i>
<i>...</i>	<i>...</i>	<i>...</i>

2. Method – Algorithm

Classification process

Example of evaluate data

Chocolate ice-cream

text description

Step 1: **Extract features** → *chocolate, ice-cream*
chocolate + ice-cream

Step 2: **Retrieval** of the corresponding **classification codes**
and **frequencies**

candidate code (item name), frequency

feature	code	count
chocolate	352	598
chocolate	345	193
chocolate	356	83
ice-cream	356	384
ice-cream	397	197
chocolate+ice-cream	356	78
strawberry+ice-cream	356	53
...

[*345(other confectionaries), 193*
352(chocolate), 598
356(ice-cream), 83
[*356(ice-cream), 384*
397(eat-out at cafe), 197
[*356(ice-cream), 78*

2. Method – Algorithm Classification process

Step 3: Calculate probability $\tilde{p}_{j k}$ for every retrieved candidate

$$\tilde{p}_{j k} = \frac{n_{j k} + \beta}{n_j + \alpha}, \quad n_j = \sum_{k=1}^K n_{j k}$$

$n_{j k}$: number of objects in a class k with j -th feature in the training dataset
 α, β : given constant, K : number of classes

$$\alpha = \beta = 0, \quad \tilde{p}_{j k} = \frac{n_{j k}}{n_j}, \quad n_j = \sum_{k=1}^K n_{j k}$$



Step 4: Determine top \tilde{K} ($\tilde{K} = 2, \dots, K$) promising candidates for each feature based on $\tilde{p}_{j k}$

feature	code	$\tilde{p}_{j j}$
chocolate	345(other confectionaries)	0.22...
	352(chocolate)	0.68...
	356(ice-cream)	0.09...
ice-cream	356(ice-cream)	0.66...
	397(eat-out at café)	0.34...
chocolate+ice-cream	356(ice-cream)	1

2. Method – Algorithm Classification process

★ Step 5: Calculate the new reliability score \bar{p}_{jk}

$$\bar{p}_{jk} = \tilde{p}_{jk} \left(1 + \sum_{m=1}^{\tilde{K}} \tilde{p}_{jm} \log_K \tilde{p}_{jm} \right)$$

$\{\tilde{p}_{j1}, \dots, \tilde{p}_{j\tilde{K}}\}$: the selected \tilde{K} largest values of \tilde{p}_{jk} , $\tilde{p}_{j1} \geq \dots \geq \tilde{p}_{j\tilde{K}} \geq \dots \geq \tilde{p}_{jK}$

\tilde{K} : selected classes for the j -th feature, $\tilde{K} \in \{2, \dots, K\}$

feature	code	\bar{p}_{jk}
chocolate	345(other confectionaries)	0.15
	352(chocolate)	0.48
ice-cream	356(ice-cream)	0.5
	397(eat-out at café)	0.26
chocolate+ice-cream	356(ice-cream)	1

Step 6: Determine top L ($L = 1, 2, 3, \dots$) candidate codes

What if $L=3$?

→ candidate codes : 356, 352, and 397

2. Method – Reliability score

\bar{p}_{jk} : reliability score of j -th feature to k code (or class)

$$\bar{p}_{jk} = \tilde{p}_{jk} (1 + \sum_{m=1}^{\tilde{K}} \tilde{p}_{jm} \log_K \tilde{p}_{jm})$$



Probability of feature j to code k



Classification status of feature j
over the \tilde{K} largest codes

Transformation from \tilde{p}_{jk} to
classification status of feature j

If both values are large, \bar{p}_{jk} will be larger
Otherwise, \bar{p}_{jk} will be smaller

3. Experiments & results



3. Experiments & results – Experiment 1, Dataset

Data : Family Income and Expenditure Survey

Volume : approx. 5.2 million instances



approx. 4.5 million instances for training

approx. 0.65 million instances for evaluation



1. Cash Receipts and Cash Disbursements

(1) Kind of Receipts / Purchased Items and Their Uses	(2) Cash Receipts Yen	(3) Quantities		(4) Cash Disbursements Yen
		Unit		
1 <i>Eating out (Pizza)</i>		2	<i>Persons</i>	3240
2 <i>Green tea leaves</i>		100	<i>g</i>	1080
3 <i>Bonito</i>		500	<i>g</i>	500
4 <i>Pork</i>		360	<i>g</i>	480
5				

3. Experiments & results – Experiment 1, Result

Classification accuracy of the proposed classifier

	Number of total instances	Number of matched instances	Number of cumulative matched instances	Cumulative accuracy
1 st candidate	655,572	592,342	592,342	0.904
2 nd candidate		30,275	622,617	0.950
3 rd candidate		9,240	631,857	0.964
4 th candidate		4,274	636,131	0.970
5 th candidate		2,519	638,650	0.974

$$\text{Cumulative accuracy} = \frac{\sum_{i=1}^T M_i}{N}$$

N : the number of input instances

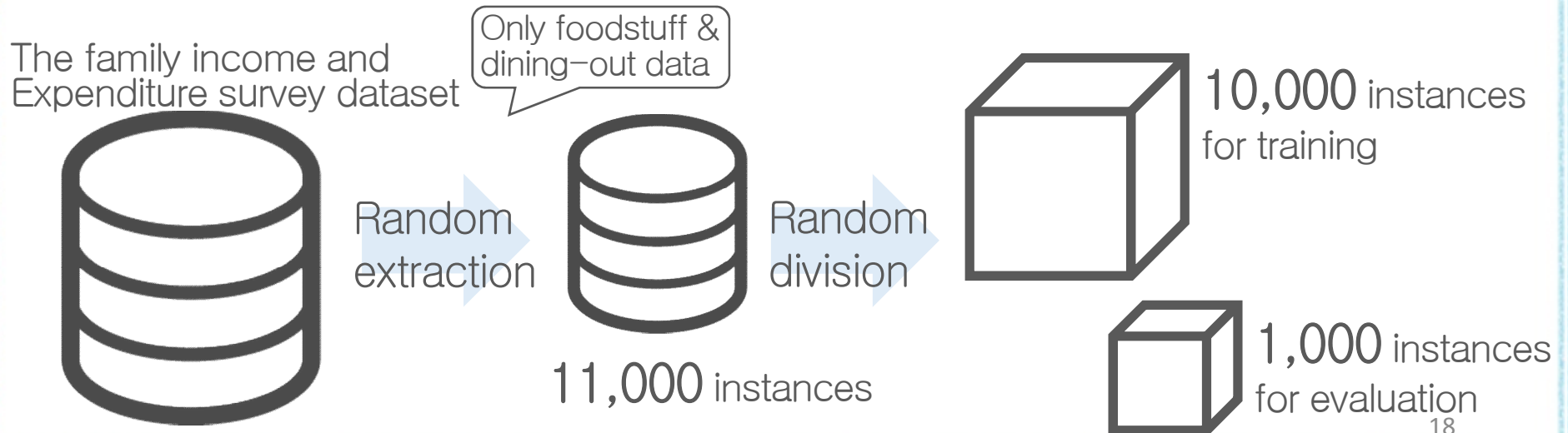
M_i : the number of matched instances at i -th candidate

3. Experiments & results – Experiment 2, Dataset

The family income and Expenditure survey mini dataset

No.	Contents	Classification code	Number of instances in dataset 1	Number of instances in dataset 2	Number of instances in dataset 3
1	Cereals	A	1,018	1,007	1,049
2	Fish and shellfish	B	927	950	926
3	Meat	C	775	746	765
4	Dairy products and eggs	D	717	727	729
5	Vegetables and seaweed	E	2,966	2,954	2,913
6	Fruits	F	485	505	498
7	Oils, fats, and seasonings	G	661	713	686
8	Cakes and candies	H	1,026	1,025	1,048
9	Cooked food	I	1,221	1,211	1,270
10	Beverages, including alcoholic beverages	J	868	845	814
11	Meals outside the home	K	336	317	302

Foodstuff and dining-out items, 11 different codes



3. Experiments & results – Experiment 2, Result

Classification accuracy of the proposed classifier

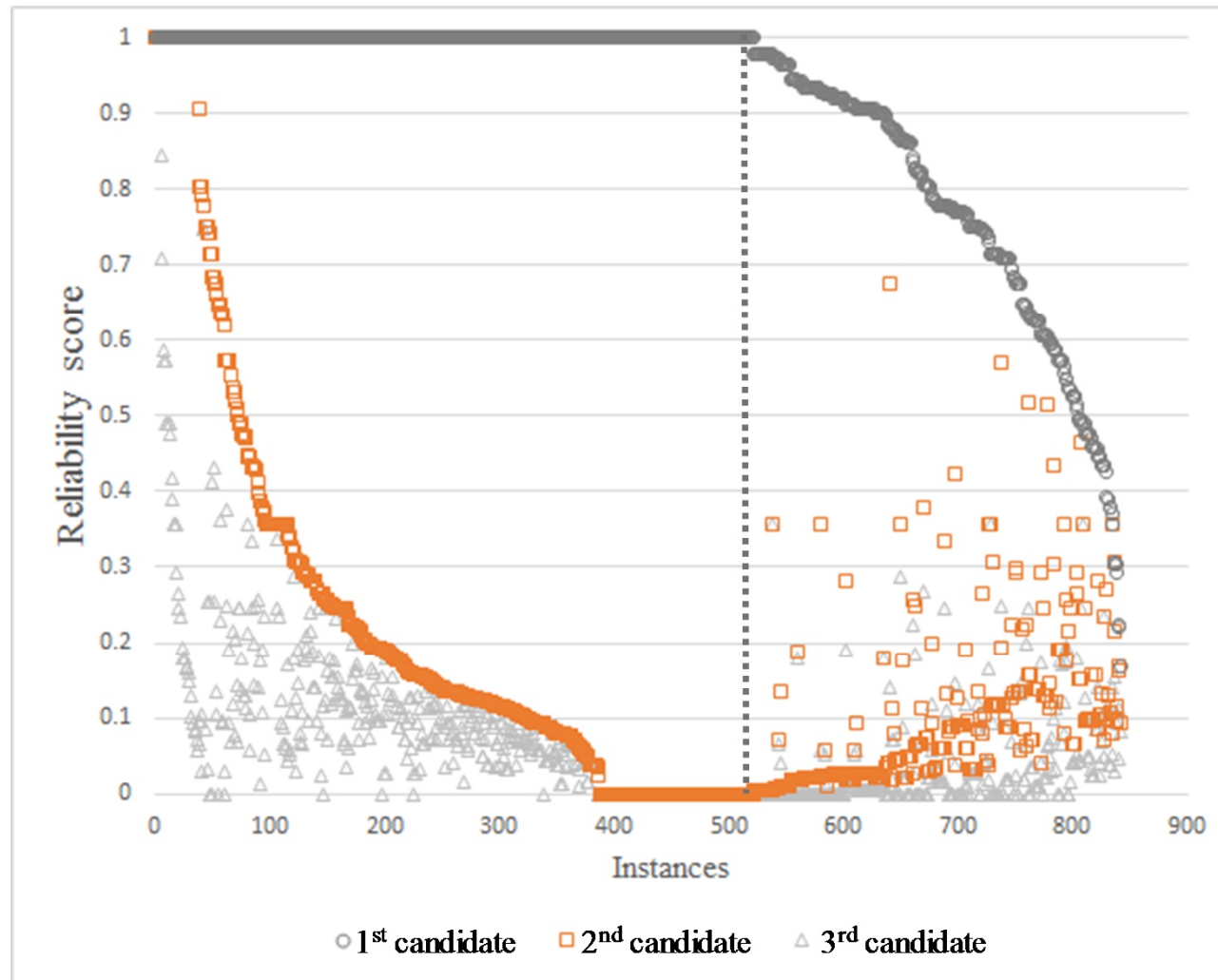
		Number of total instances	Number of matched instances	Number of cumulative matched instances	Cumulative accuracy
dataset 1	1st candidate	1,000	842	842	0.842
	2nd candidate		68	910	0.910
	3rd candidate		14	924	0.924
dataset 2	1st candidate		832	832	0.832
	2nd candidate		69	901	0.901
	3rd candidate		26	927	0.927
dataset 3	1st candidate		837	837	0.837
	2nd candidate		59	896	0.896
	3rd candidate		32	928	0.928

Classification accuracy of competing classifiers

		Number of total instances	Number of matched instances	Accuracy
dataset 1	Our previous classifier	1,000	842	0.842
	Random forest		822	0.822
dataset 2	Our previous classifier		819	0.819
	Random forest		822	0.822
dataset 3	Our previous classifier		839	0.839
	Random forest		802	0.802

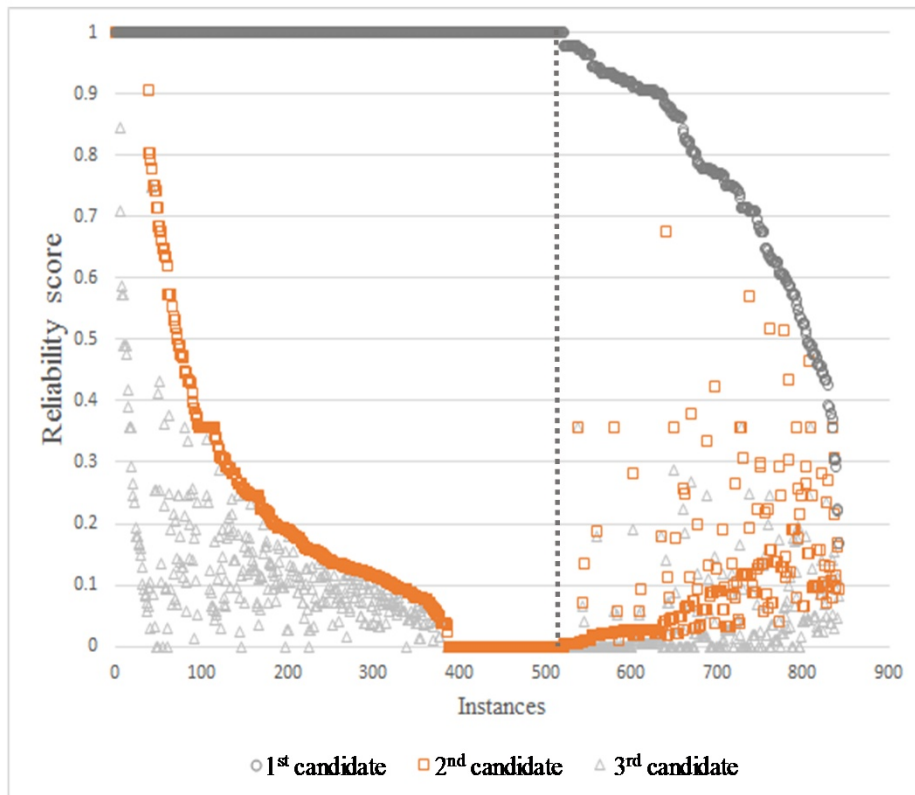
3. Experiments & results – Experiment 2, Result

Reliability score of instances that match with the 1st candidate code in dataset 1

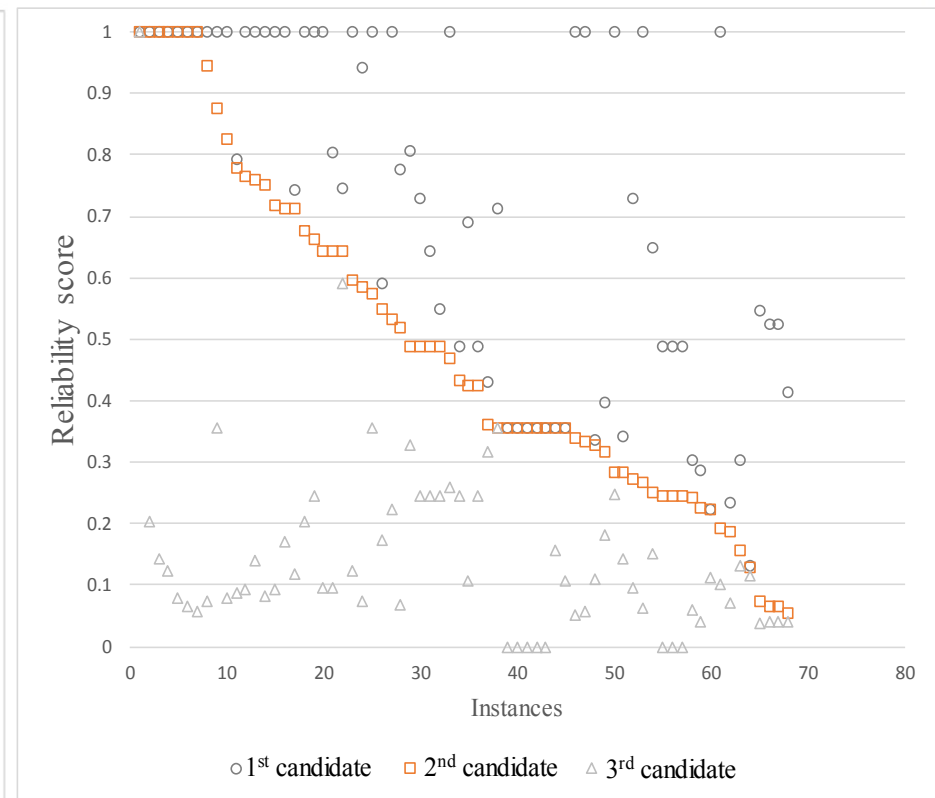


3. Experiments & results – Experiment 2, Result

Reliability score of instances that match with the 1st candidate code in dataset 1



Reliability score of instances that match with the 2nd candidate code in dataset 1



4. Summary



4. Summary

- * Proposed a new algorithm for overlapping classification
- * Listed multiple candidates according to the new defined reliability score
- * Improved the classification performance from our previous study
- * Implemented in R



Thank you
for your
attention!

ytoko@nstac.go.jp