A new editing scheme based on a fast two-string median computation applied to OCR

José Ignacio Abreu Salas[1] **Juan Ramón Rico-Juan**[2]

[1]Universidad de Matanzas,Cuba
jose.abreu@umcc.cu

[2]Pattern Recognition and Artificial Intelligence Group
Department of Software and Computing Systems
University of Alicante, E-03071 Alicante, Spain,
juanra@dlsi.ua.es

August 18-20, 2010 (S+SSPR)

Outline

1. Motivation
2. Prototype Construction
3. Editing Algorithm
4. Experiments
5. Conclusions & Future Work

Motivation

1 Motivation
2 Prototype Construction
3 Editing Algorithm
4 Experiments
5 Conclusions & Future Work

Motivation

Introduction

- **Dataset editing** has received considerable attention from the seminal works of Wilson [Wilson, 1972].
- **Edited near neighbor rule** technique can be useful to improve **nearest neighbor classifiers response**.
- Algorithms as [Devijver and Kittler, 1980], [Ferri and Vidal, 1992], [Tomek, 1976a] and [Vázquez et al., 2005] focus on **deleting wrong tagged instances** from a training set and improve basic Wilson algorithm.
- Another group of algorithms also **changes some instances tag** as [Koplowitz and Brown, 1981] and [Tomek, 1976b].
- Syntactic coding such **strings** and **trees** are commonly used instead of vectorial representations.

Motivation

Baseline

- **Wilson algorithm** to edit training set;
- **Strings** (Freeman Chain-code) as prototypes;
- K **Nearest Neighbor** as classification technique.

Our goal

- Revise the **Wilson editing scheme** for improve classification task by adding new **artificial prototypes**.
- Define a **fast two-string** median algorithm to create a new artificial prototype.

Motivation

Baseline

- **Wilson algorithm** to edit training set;
- **Strings** (Freeman Chain-code) as prototypes;
- K **Nearest Neighbor** as classification technique.

Our goal

- Revise the **Wilson editing scheme** for improve classification task by adding new **artificial prototypes**.
- Define a **fast two-string** median algorithm to create a new artificial prototype.

Prototype Construction

1 Motivation

2 Prototype Construction

3 Editing Algorithm

4 Experiments

5 Conclusions & Future Work

Prototype Construction

Remarks

- **New prototype** is built from **two strings** (Freeman chain-code).
- This approach is **easier** and **faster** than compute median string between N prototypes.
- **The Levenshtein edit distance** [Levenshtein, 1966] is used as starting point.

Edit Distance

- Let Σ be an alphabet and $S_1 = \{S_{11}, S_{12}..S_{1m}\}$, $S_2 = \{S_{21}, S_{22}..S_{2n}\}$ two strings over Σ where $m, n \geq 0$
- The edit distance between S_1 and S_2, $D(S_1, S_2)$, is defined in terms of elementary edit operations:
  - *substitution* of a symbol $a \in S_1$ by a symbol $b \in S_2$, denoted as $w(a, b)$
  - *insertion* of a symbol $b \in S_2$ in S_1, denoted as $w(\varepsilon, b)$
  - *deletion* of a symbol $a \in S_1$, denoted as $w(a, \varepsilon)$.
- Let $Q_{Si}^{Sj} = \{q_1, q_2, ..., q_k\}$ be a sequence of edit operations transforming S_i into S_j
- Cost of the sequence, $E_{Q_{S2}^{S1}} = \sum_{i=1}^{k} e(q_i)$
- $D(S_1, S_2) = argmin\{ E_{Q_{S2}^{S1}}\}$
- The dynamic programming [Wagner and Fischer, 1974] allows to compute $D(S_1, S_2)$ in $\mathcal{O}(L_{S1} \times L_{S2})$ time.

Fast Median String Computation

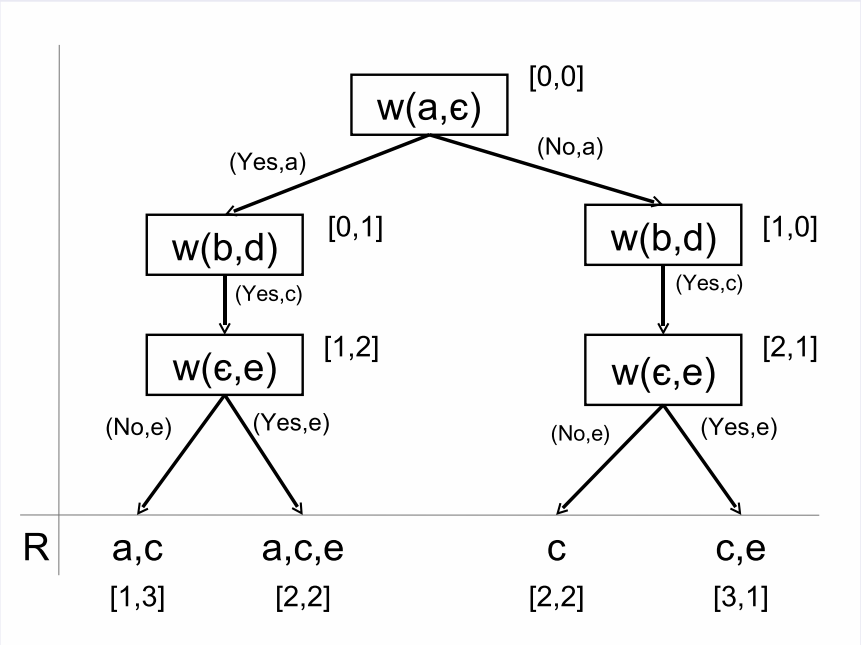
Median String

- The median of a set T of strings is defined as $argmin_R\{\sum D(R, S_i)|S_i \in T\}$
- In this case, two strings are used.
- Goodness criterion $argmin_R\{|D(R, S_1) - D(R, S_2)|\}$
- It satisfies $D(S_1, S_2) = D(S_1, R) + D(R, S_2)$

Fast Median String Computation

Example

- $S_1 = \{a, b\}$ and $S_2 = \{d, e\}$
- Costs: $e(w(\cdot, \varepsilon)) = e(w(\varepsilon, \cdot)) = 1$ and $e(w(x, y)) = lexical\_order(|x - y|)$
- Minimum cost sequence $Q_{S1}^{S2} = \{w(a, \varepsilon), w(b, d), w(\varepsilon, e)\} = 1 + 2 + 1 = 4$.



Editing Algorithm

1 Motivation

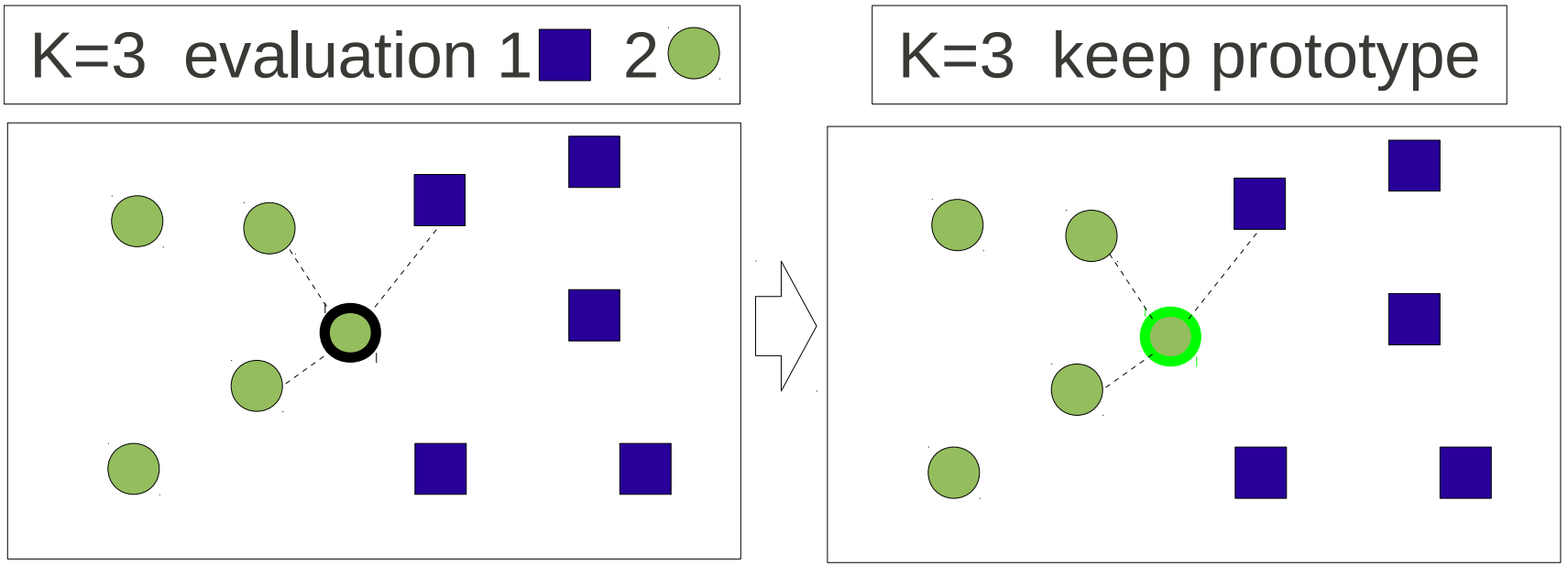
2 Prototype Construction

3 Editing Algorithm

4 Experiments

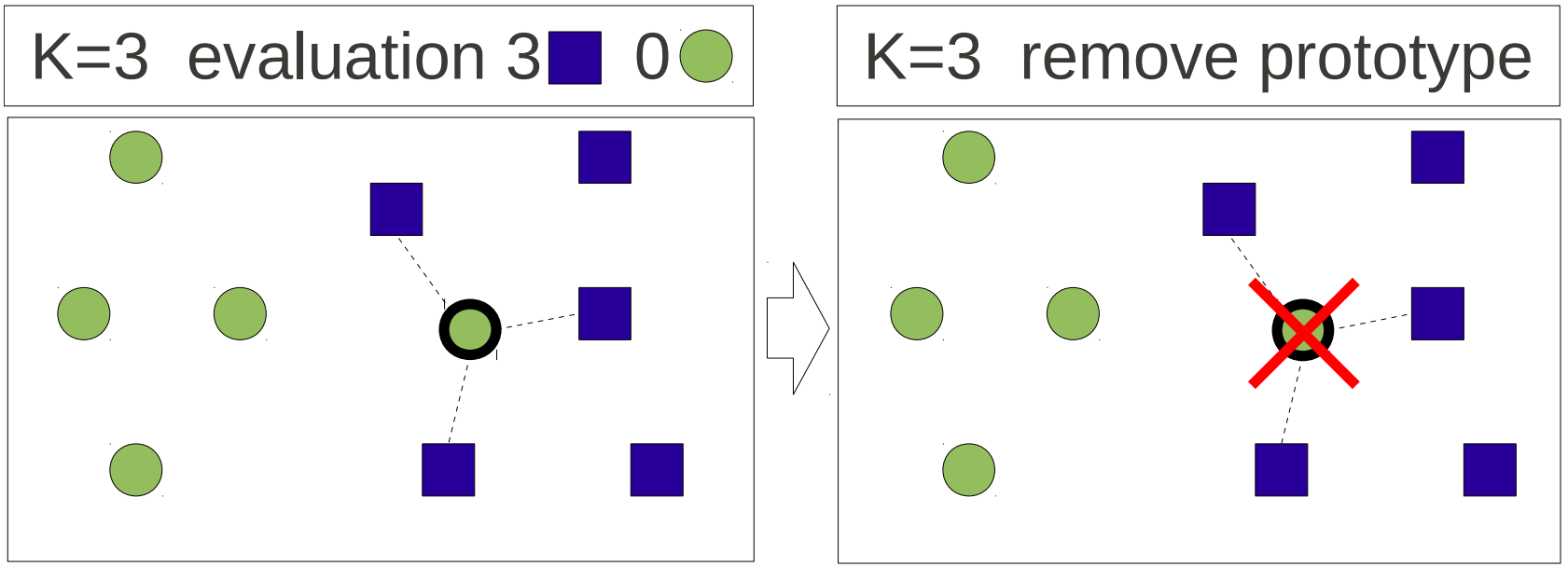
5 Conclusions & Future Work

Wilson vs. our scheme (JJWilson): well classified



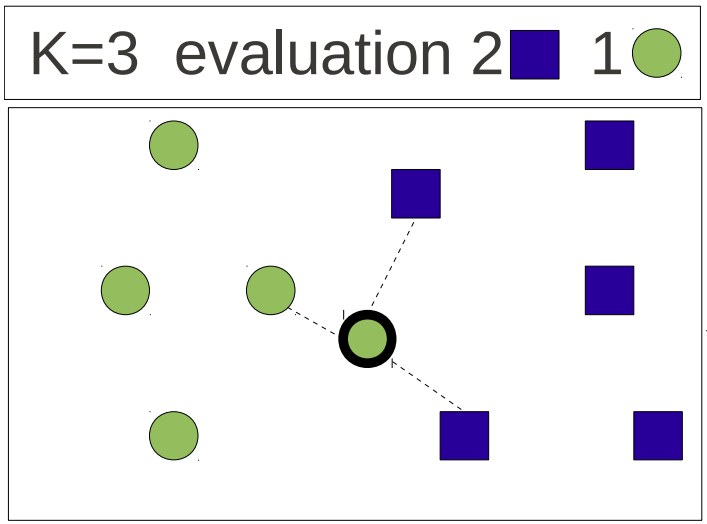
The prototype is K-NN well classified

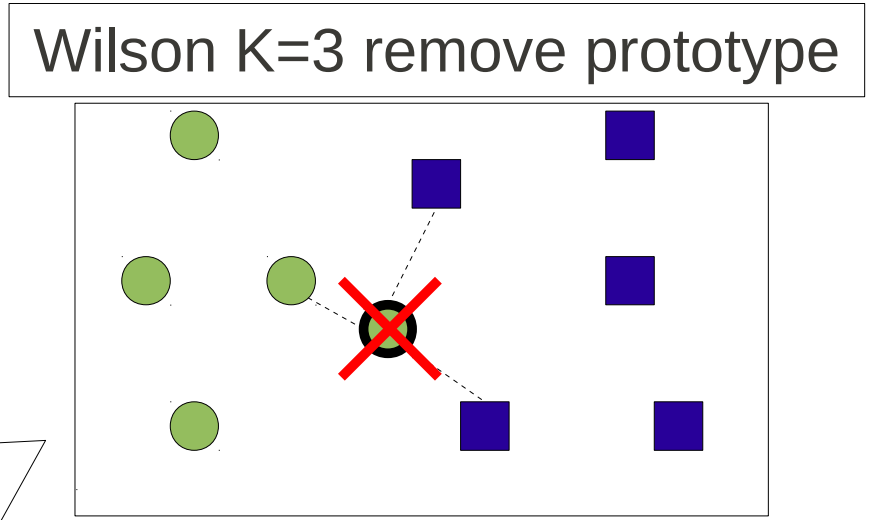
Wilson vs. our scheme (JJWilson): bad classified

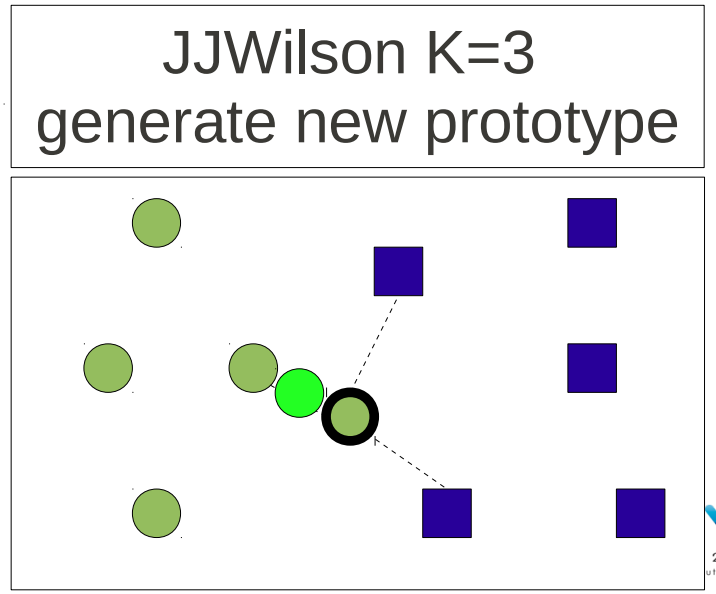


The prototype is K-NN bad classified
(0 nearest neighbors from the same class)

Wilson vs. our scheme (JJWilson): the difference







The prototype is K-NN bad classified (there are **some nearest neighbors** from the same class)

Experiments

1 Motivation

2 Prototype Construction

3 Editing Algorithm

4 Experiments

5 Conclusions & Future Work

Preparing experiments

Database

- A contour subsets (digits and letters) are extracted from the NIST SPECIAL DATABASE 3 of the National Institute of Standards and Technology.
- 4-fold cross-validation technique are used (60-training and 20-test instances per class)
- K-Nearest Neighbor rule is used to edit and classify.
  - Edit: $K = 3..17$
  - Classify: $K = 1..17$

Experiments: Characters

Character set: Average % error rate (4-folds)

		K on Edition													
		K=3		K=5		K=7		K=9		K=11		K=13		K=15	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	13.7	16.5	14.6	15.8	13.8	16.3	13.8	16.8	13.8	17.1	**13.4**	17.1	**13.4**	17.5	**13**
3	14.7	17.6	15.3	17.6	14.6	17.5	**14.4**	17.8	**14.2**	18.8	**14.0**	18.9	**13.9**	19.6	**13**
5	15.4	17.6	**15.2**	17.9	**14.4**	17.9	**14.1**	18.5	**14.1**	18.9	**14.2**	19.5	**14.1**	19.8	**13**
7	16.0	19.4	16.2	19.6	**15.1**	19.9	**15.2**	19.8	**14.7**	20.2	**14.2**	20.7	**14.0**	20.8	**14**
9	17.1	19.5	**16.3**	20.1	**15.5**	20.3	**15.3**	20.5	**14.8**	21.0	**14.8**	21.6	**14.5**	21.8	**14**
11	17.7	20.0	17.7	20.8	**16.5**	20.8	**16.1**	21.3	**15.4**	21.6	**15.0**	22.3	**15.0**	22.3	**15**
13	18.3	21.0	**18.2**	21.2	**17.1**	21.7	**16.4**	22.1	**16.5**	22.5	**15.8**	22.8	**15.5**	23.3	**15**
15	18.6	22.0	18.9	21.6	**18.0**	22.3	**17.1**	22.6	**16.7**	23.3	**16.2**	23.5	**16.3**	23.7	**16**
17	19.6	22.1	**18.8**	22.7	**18.0**	23.0	**17.5**	23.8	**17.4**	24.0	**16.9**	24.0	**16.5**	24.6	**16**

Experiments: Characters

Character set: Average % error rate (4-folds)

		K on Edition									
		K=3		K=5		K=7		K=9		K=11	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	13.7	16.5	14.6	15.8	13.8	16.3	13.8	16.8	13.8	17.1	**13.4**
3	14.7	17.6	15.3	17.6	14.6	17.5	**14.4**	17.8	**14.2**	18.8	**14.0**
5	15.4	17.6	**15.2**	17.9	**14.4**	17.9	**14.1**	18.5	**14.1**	18.9	**14.2**
7	16.0	19.4	16.2	19.6	**15.1**	19.9	**15.2**	19.8	**14.7**	20.2	**14.2**
9	17.1	19.5	**16.3**	20.1	**15.5**	20.3	**15.3**	20.5	**14.8**	21.0	**14.8**
11	17.7	20.0	17.7	20.8	**16.5**	20.8	**16.1**	21.3	**15.4**	21.6	**15.0**
13	18.3	21.0	**18.2**	21.2	**17.1**	21.7	**16.4**	22.1	**16.5**	22.5	**15.8**
15	18.6	22.0	18.9	21.6	**18.0**	22.3	**17.1**	22.6	**16.7**	23.3	**16.2**
17	19.6	22.1	**18.8**	22.7	**18.0**	23.0	**17.5**	23.8	**17.4**	24.0	**16.9**

Experiments: Characters

% error rate (4-folds)

K on Edition											
K=7		K=9		K=11		K=13		K=15		K=17	
Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
16.3	13.8	16.8	13.8	17.1	**13.4**	17.1	**13.4**	17.5	**13.2**	17.8	**13.1**
17.5	**14.4**	17.8	**14.2**	18.8	**14.0**	18.9	**13.9**	19.6	**13.8**	19.9	**13.3**
17.9	**14.1**	18.5	**14.1**	18.9	**14.2**	19.5	**14.1**	19.8	**13.8**	20.2	**13.7**
19.9	**15.2**	19.8	**14.7**	20.2	**14.2**	20.7	**14.0**	20.8	**14.0**	21.3	**14.1**
20.3	**15.3**	20.5	**14.8**	21.0	**14.8**	21.6	**14.5**	21.8	**14.6**	22.2	**14.6**
20.8	**16.1**	21.3	**15.4**	21.6	**15.0**	22.3	**15.0**	22.3	**15.2**	23.1	**15.0**
21.7	**16.4**	22.1	**16.5**	22.5	**15.8**	22.8	**15.5**	23.3	**15.5**	23.7	**15.7**
22.3	**17.1**	22.6	**16.7**	23.3	**16.2**	23.5	**16.3**	23.7	**16.0**	24.2	**16.0**
23.0	**17.5**	23.8	**17.4**	24.0	**16.9**	24.0	**16.5**	24.6	**16.5**	24.8	**16.4**

Experiments: Digits

Digit set: Average % error rate (4-folds)

		K on Edition															
		K=3		K=5		K=7		K=9		K=11		K=13		K=15		K=17	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	1.8	2.8	1.9	2.6	1.8	2.5	1.8	2.9	**1.6**	2.8	**1.6**	2.8	**1.6**	2.8	**1.5**	2.8	**1.5**
3	2.0	3.1	2.3	2.9	2.3	3.0	2.0	3.3	**1.9**	3.4	2.0	3.4	2.0	3.8	**1.9**	3.9	**1.8**
5	3.0	3.6	**2.9**	3.6	**2.8**	3.8	**2.8**	4.3	**2.6**	4.3	**2.6**	4.1	**2.5**	4.3	**2.3**	4.4	**2.5**
7	3.5	4.3	3.5	4.3	**2.9**	4.3	**2.9**	4.5	**2.6**	4.6	**2.6**	4.8	**2.6**	5.0	**2.5**	5.1	**2.4**
9	3.6	4.3	**3.5**	4.1	**3.3**	4.3	**3.0**	4.6	**2.9**	4.9	**2.8**	4.9	**2.8**	5.1	**2.8**	5.4	**2.8**
11	4.1	4.5	**2.9**	4.5	**2.9**	4.6	**2.9**	4.6	**3.0**	4.6	**3.6**	4.8	**3.5**	4.9	**3.5**	5.3	**3.4**
13	4.4	4.8	**3.1**	4.8	**3.3**	5.0	**3.1**	5.0	**3.3**	5.3	**4.0**	5.6	**3.5**	5.9	**3.5**	5.9	**3.4**
15	4.8	5.1	**3.8**	5.1	**3.8**	5.4	**3.6**	5.5	**3.8**	5.5	**4.5**	6.1	**4.3**	6.3	**4.1**	6.3	**3.9**
17	4.9	5.1	**4.1**	5.1	**4.0**	5.4	**3.8**	5.5	**3.9**	5.5	**4.6**	6.1	**4.2**	6.0	**4.3**	6.1	**4.3**

Experiments: Digits

Digit set: Average % error rate (4-folds)

		K on Edition											
		K=3		K=5		K=7		K=9		K=11		K=13	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	1.8	2.8	1.9	2.6	1.8	2.5	1.8	2.9	**1.6**	2.8	**1.6**	2.8	**1.6**
3	2.0	3.1	2.3	2.9	2.3	3.0	2.0	3.3	**1.9**	3.4	2.0	3.4	2.0
5	3.0	3.6	**2.9**	3.6	**2.8**	3.8	**2.8**	4.3	**2.6**	4.3	**2.6**	4.1	**2.5**
7	3.5	4.3	3.5	4.3	**2.9**	4.3	**2.9**	4.5	**2.6**	4.6	**2.6**	4.8	**2.6**
9	3.6	4.3	**3.5**	4.1	**3.3**	4.3	**3.0**	4.6	**2.9**	4.9	**2.8**	4.9	**2.8**
11	4.1	4.5	**2.9**	4.5	**2.9**	4.6	**2.9**	4.6	**3.0**	4.6	**3.6**	4.8	**3.5**
13	4.4	4.8	**3.1**	4.8	**3.3**	5.0	**3.1**	5.0	**3.3**	5.3	**4.0**	5.6	**3.5**
15	4.8	5.1	**3.8**	5.1	**3.8**	5.4	**3.6**	5.5	**3.8**	5.5	**4.5**	6.1	**4.3**
17	4.9	5.1	**4.1**	5.1	**4.0**	5.4	**3.8**	5.5	**3.9**	5.5	**4.6**	6.1	**4.2**

Experiments: Digits

ror rate (4-folds)

K on Edition										
=7	K=9		K=11		K=13		K=15		K=17	
JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1.8	2.9	**1.6**	2.8	**1.6**	2.8	**1.6**	2.8	**1.5**	2.8	**1.5**
2.0	3.3	**1.9**	3.4	2.0	3.4	2.0	3.8	**1.9**	3.9	**1.8**
2.8	4.3	**2.6**	4.3	**2.6**	4.1	**2.5**	4.3	**2.3**	4.4	**2.5**
2.9	4.5	**2.6**	4.6	**2.6**	4.8	**2.6**	5.0	**2.5**	5.1	**2.4**
3.0	4.6	**2.9**	4.9	**2.8**	4.9	**2.8**	5.1	**2.8**	5.4	**2.8**
2.9	4.6	**3.0**	4.6	**3.6**	4.8	**3.5**	4.9	**3.5**	5.3	**3.4**
3.1	5.0	**3.3**	5.3	**4.0**	5.6	**3.5**	5.9	**3.5**	5.9	**3.4**
3.6	5.5	**3.8**	5.5	**4.5**	6.1	**4.3**	6.3	**4.1**	6.3	**3.9**
3.8	5.5	**3.9**	5.5	**4.6**	6.1	**4.2**	6.0	**4.3**	6.1	**4.3**

Conclusions & Future Work

1 Motivation

2 Prototype Construction

3 Editing Algorithm

4 Experiments

5 Conclusions & Future Work

Conclusions & Future Work

Conclusions

- A **novelty method** was presented to **edit a dataset of contours** encoded by Freeman Chain-code.
- A **new fast procedure** to compute the **median between two strings** based on a string edit distance is explained.
- **Experiments** show that our edit **scheme behaves well** on the studied datasets.

Future work

- Revise **misclassified instance** detection, and consider **other criteria** like SMOTE (Synthetic Minority Over-sampling TEchnique).
- **Others datasets** could be studied and **compared** with **additional edit methods**.
- **Our fast median two-strings algorithm** could be extended to compute the average of N **examples**.

Conclusions & Future Work

Conclusions

- A **novelty method** was presented to **edit a dataset of contours** encoded by Freeman Chain-code.
- A **new fast procedure** to compute the **median between two strings** based on a string edit distance is explained.
- **Experiments** show that our edit **scheme behaves well** on the studied datasets.

Future work

- Revise **misclassified instance** detection, and consider **other criteria** like SMOTE (Synthetic Minority Over-sampling TEchnique).
- **Others datasets** could be studied and **compared** with **additional edit methods**.
- **Our fast median two-strings algorithm** could be extended to compute the average of N **examples**.

The end

Thanks for your attention

A new editing scheme based on a fast two-string median computation applied to OCR

José Ignacio Abreu Salas[1] **Juan Ramón Rico-Juan**[2]

[1]Universidad de Matanzas,Cuba
jose.abreu@umcc.cu

[2]Pattern Recognition and Artificial Intelligence Group
Department of Software and Computing Systems
University of Alicante, E-03071 Alicante, Spain
juanra@dlsi.ua.es