Autocoding based on Multi-Class Support Vector Machine by Fuzzy c-means Method

25 Nov. 2021

uRos2021

Yukako Toko[1], Mika Sato-Ilic[2]

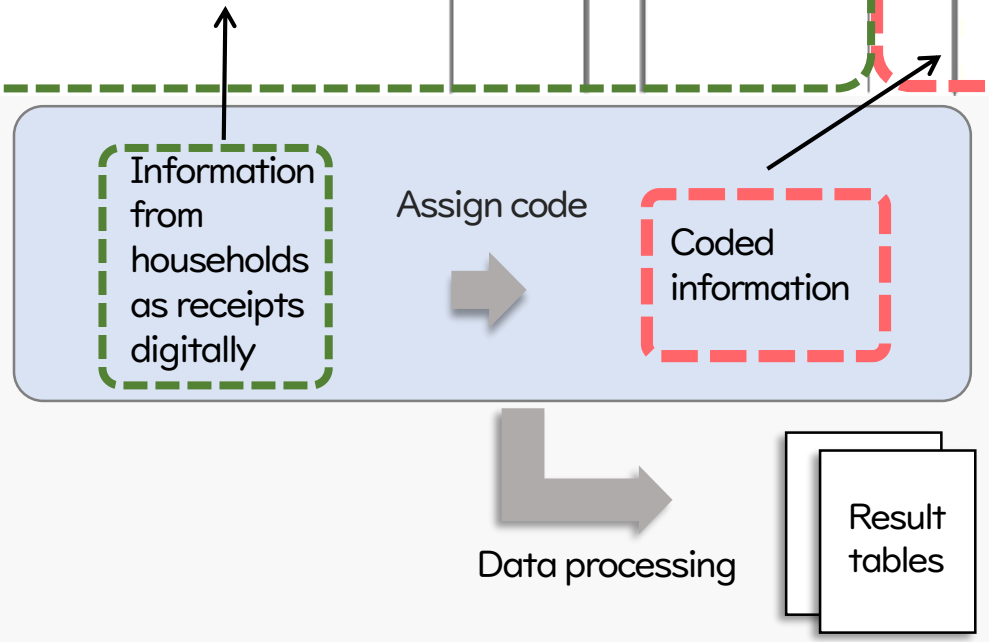
[1]National Statistics Center

[2]University of Tsukuba

Coding Task for Family Income and Expenditure Survey

Example of the family Income and Expenditure Survey data

Purchased items and their uses	Quantities	Unit	Cash disbursements (Yen)	Code	Category
Sandals (for ladies)	1	pair	3,564	672	Shoes (for ladies)
Toilet rolls			646	532	Toilet rolls
Chicken	279	g	538	222	Chicken meat
Chocolate assortment box			149	352	Chocolate
Hot chilli sauce	300	ml	214	328	Sauce



Assigning corresponding class (or code) is an essential activity for efficient data processing in official statistics!!

Background

Coding task of the Family Income and Expenditure Survey

Autocoding method:
Reliability score based Bernoulli type simple Bayes model

Improve the Bernoulli type simple Bayes model which is one of naïve Bayes by introducing fuzzy measure as the reliability score

Improved method performs well and it will be practically implemented for autocoding of the Family Income and Expenditure Survey in 2022

However,

It is well known that the Bernoulli type simple Bayes model does not perform well for large amounts of complex data

Whereas, **the data of the Family Income and Expenditure Survey included text descriptions that extracted from receipts digitally is getting large and complex**

Objective

To obtain stable results of discrimination as coding with high generalization performance dealing with cognitive uncertainty for text description data,

we propose a new autocoding method which is a combined method of Support Vector Machine(SVM) and Fuzzy c-means(FCM).

- Utilizing SVM that is a machine learning method known as high generalization performance.
- Utilizing Fuzzy c-means method that is a computational intelligence method known as high performance dealing with cognitive uncertainty with linguistically motivated computation.

Related works:
Hybrid Method of SVM Utilizing Word2Vec and Classification Method based on the Reliability Score

Hybrid method of SVM utilizing Word2Vec and the previously developed classification method based on the reliability score was developed.

Improve both the ability of high classification accuracy and generalization performance.

Toko, Y., Sato-Ilic, M.: Efficient Autocoding Method in High Dimensional Space, Romanian Statistical Review, 1, 3-16 (2021)

Related works:
Hybrid Method of Multi-Class SVM Utilizing Word2Vec and Classification Method based on the Reliability Score

Hybrid method of SVM utilizing Word2Vec and the previously developed classification method based on the reliability score was developed.

Improve both the ability of high classification accuracy and generalization performance.

Toko, Y., Sato-Ilic, M.: Efficient Autocoding Method in High Dimensional Space, Romanian Statistical Review, 1, 3-16 (2021)

Including Multi-Class SVM

To improve the accuracy of the result of SVM in the developed hybrid method , a clustering method is applied.

We propose a new hybrid method of **multi-class SVM** and **classification method based on reliability score** in which SVM applies to each obtained cluster individually.

Toko, Y., Sato-Ilic, M : A Hybrid Method of Multi-Class SVM and Classification Method Based on Reliability Score for Autocoding of the Family Income and Expenditure Survey, Czarnowski, I., Howlett, R.J., Jain, L. C. (Eds.), Smart Innovation, Systems and Technologies, 238, pp. 403-414. Springer (2021)

Objective

Text descriptions that extracted from **receipts digitally**

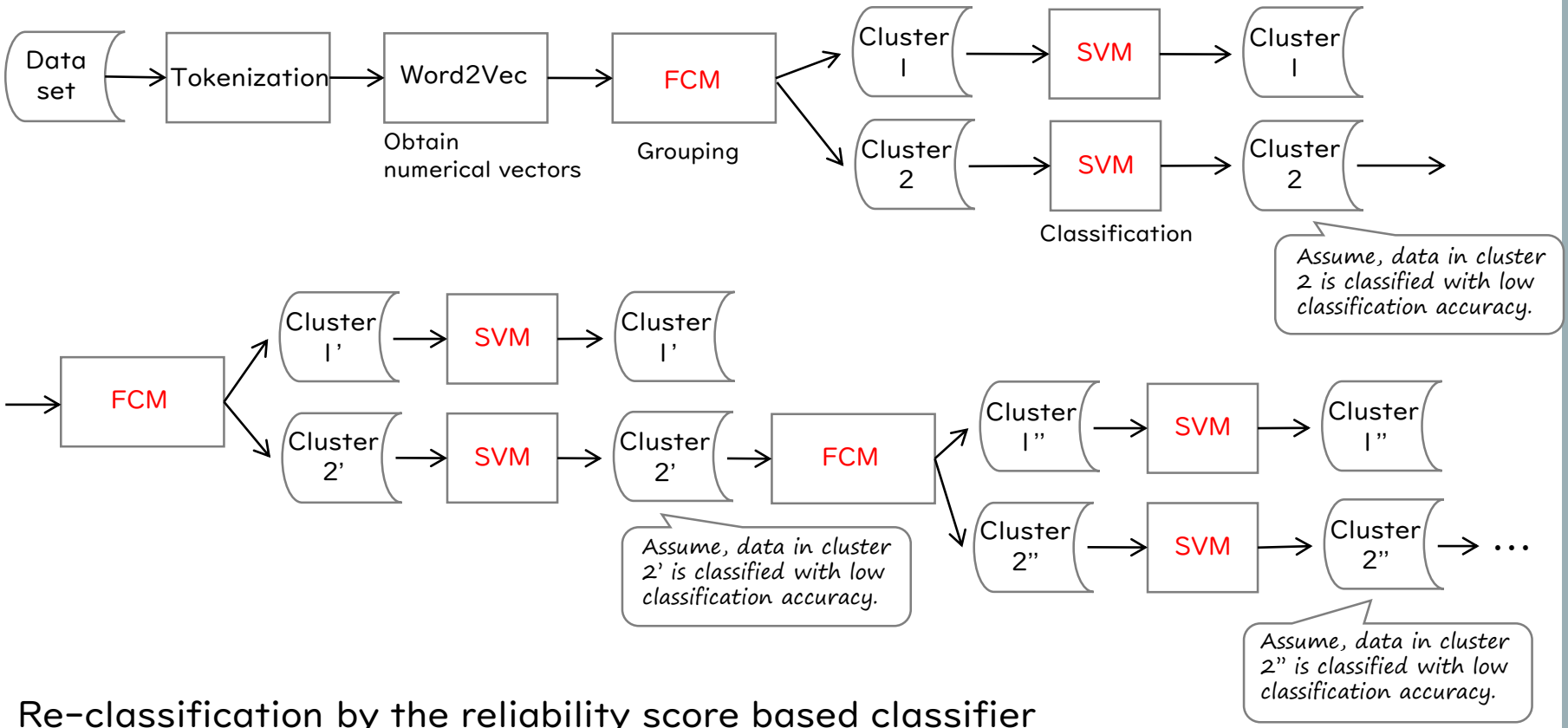
To obtain stable results of discrimination as coding with high generalization performance dealing with cognitive uncertainty for text description data,

we propose a new autocoding method which is a combined method of Support Vector Machine(SVM) and Fuzzy c-means(FCM).

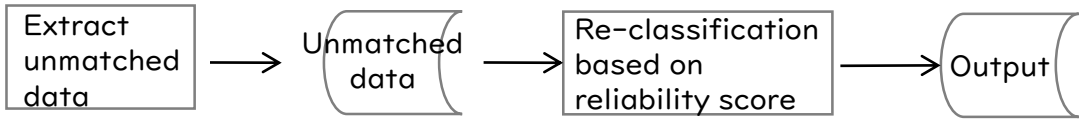
- Utilizing Fuzzy c-means method that is a computational intelligence method known as high performance dealing with cognitive uncertainty.
- Computational intelligence is linguistically motivated computational paradigms, theory and design of fuzzy logic, neural networks, and evolutionary computation.

Overview of the proposed algorithm

Classification method of SVM and FCM



Re-classification by the reliability score based classifier



Fuzzy c-means (FCM)

$$J(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{n}(u_{ik})^m \|\boldsymbol{x}_i - \boldsymbol{v}_k\|^2$$

$\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ip})$: $i\ th$ object, $i = 1, \cdots, n$

$V = (v_{ka})$, $\boldsymbol{v}_k = (v_{k1}, \cdots, v_{kp})$: center of cluster k, $k = 1, \cdots, K, a = 1, \cdots, p$

$U = (u_{ik})$, u_{ik}: degree of belongingness of $i\ th$ object to a cluster k

$u_{ik} \in [0,1]$, $\sum_{k=1}^{K} u_{ik} = 1$, $i = 1,2 \ldots, n$

m: weighting exponent; $m \in (1, \infty)$ K: number of clusters n: number of objects p: number of variables

algorithm

Step 1. Initialize the degree of belongingness of objects to clusters

$$u_{ik} = \left[\sum_{j=1}^{K}\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{v}_k\|}{\|\boldsymbol{x}_i - \boldsymbol{v}_j\|}\right)^{\frac{2}{m-1}}\right]^{-1}$$

Step 2. Calculate the cluster centers

$$\boldsymbol{v}_k = \frac{\sum_{i=1}^{n}(u_{ik})^m \boldsymbol{x}_i}{\sum_{i=1}^{n}(u_{ik})^m}$$

Step 3. Update the degree of belongingness of objects to clusters

Step 4. Stop if the difference of the degree of belongingness of objects to clusters and the degree calculated in the previous iteration is smaller than ε; Otherwise, iterate steps 2 and 3.

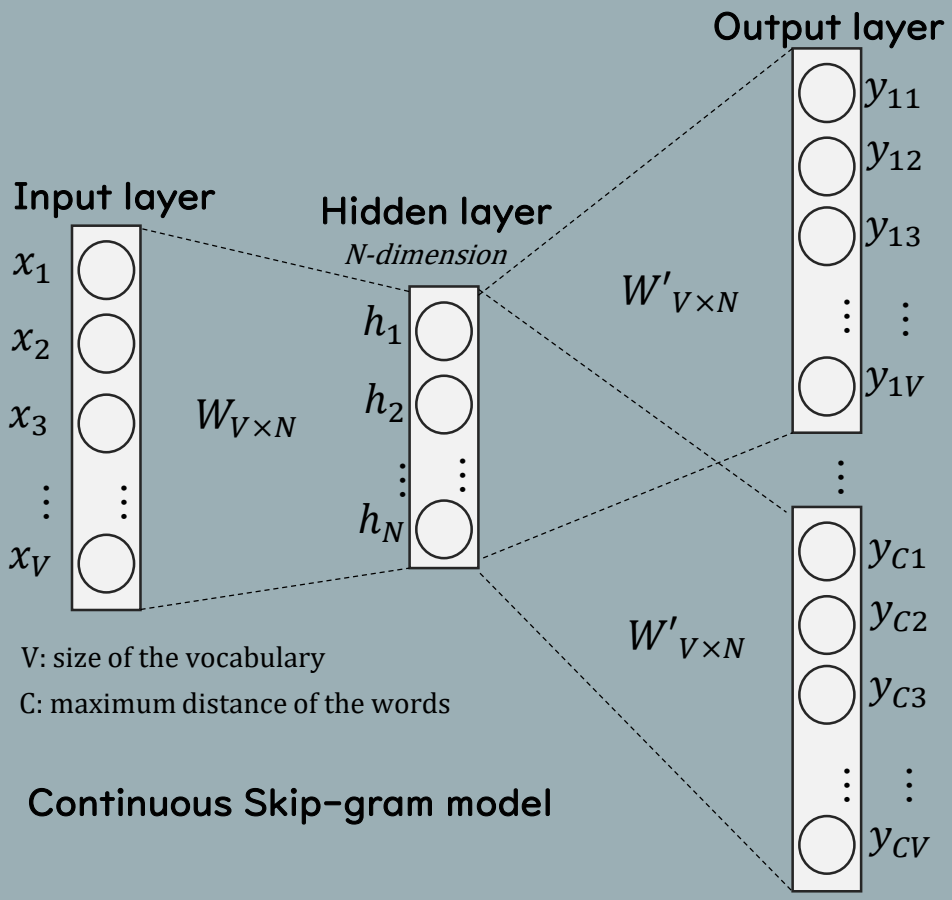
Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

Word2Vec

Word2vec algorithm learns word association from a given dataset utilizing a neural network model based on an idea of a Neural Probabilistic Language Model.

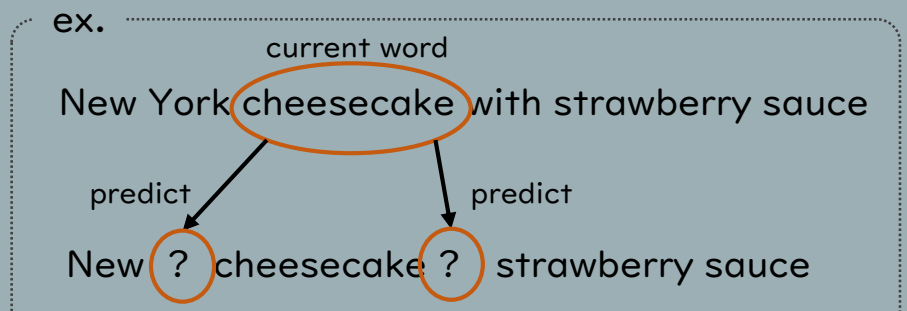
The essence of the idea is to avoid the curse of dimensionality by Distributed Representations of words.

It produces a vector space and each word in the given dataset is assigned a corresponding numerical vector of a word in the produced vector space.



Skip-gram model uses each current word in order to predict words within a certain range before and after the current word.

It gives less weight to the distant context words.

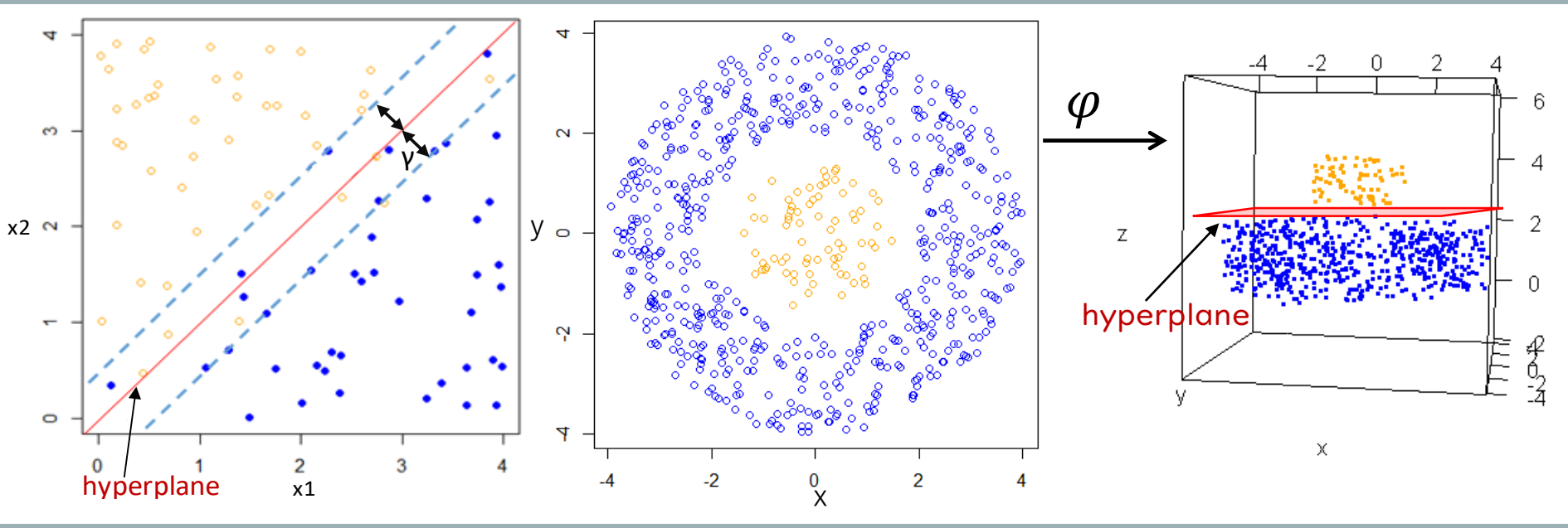


Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013)

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model, Journal of Machine Learning Research, 3, pp. 1137-1155 (2003)

Support Vector Machine (SVM)

- SVM is a supervised machine learning algorithm for classification.
- It finds the maximum-margin hyperplane in high dimensional space for classification.
- It performs a non-linier classification mapping of input data into a higher dimensional space.



Classification method based on reliability score

$\bar{p}_{jk}$: Reliability Score of j-th object to a class k

$$\bar{p}_{jk} = T\left(\tilde{\tilde{p}}_{jk}, \sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm}^2\right), \qquad j = 1, \dots, J, \qquad k = 1, \dots, \tilde{K}_j.$$

$$\bar{p}_{jk} = T\left(\tilde{\tilde{p}}_{jk}, 1 + \sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm} \log_K \tilde{\tilde{p}}_{jm}\right), \qquad j = 1, \dots, J, \qquad k = 1, \dots, \tilde{K}_j.$$

Probability measure

Relative frequency of object ***j*** to class ***k***

Fuzzy

Classification status of object ***j*** over the $\tilde{K}_j$ classes

Explanation of the uncertainty of the training data.
Utilization on the deference of measurement of uncertainty.

Transformation from $\tilde{\tilde{p}}_{jk}$ to classification status of object ***j***

$$\bar{\bar{p}}_{jk} = g(n_j)\bar{p}_{jk}$$

The classifier arranges $\{p_{j1}, \dots, p_{jK}\}$ in descending order and creates $\{\tilde{p}_{j1}, \dots, \tilde{p}_{jK}\}$, such as $\tilde{p}_{j1} \geq \cdots \geq \tilde{p}_{jK}, j = 1, \dots, J$.
After that, $\{\tilde{\tilde{p}}_{j1}, \dots, \tilde{\tilde{p}}_{j\tilde{K}_j}\}, \tilde{K}_j \leq K$ are created.

T : T-norms (Menger, K, the National Academy of Sciences, USA 1942)

p_{jk}: Relative frequency of object ***j*** to class ***k***

$$p_{jk} = \frac{n_{jk}}{n_j}, \qquad n_j = \sum_{k=1}^{K} n_{jk}, \qquad j = 1, \dots, J, \qquad k = 1, \dots, K$$

n_{jk}: Number of text descriptions in a class ***k*** with ***j***-th object in the training dataset

$g(n_j)$: Weight for control size of object ***j*** $\qquad g(n_j) = n_j / \sqrt{1 + n_j{}^2}, \qquad g(n_j) = \tanh n_j$

Toko, Y., Iijima, S., Sato-Ilic, M.: Generalization for Improvement of the Reliability Score for Autocoding, Journal of Romanian Statistical Review, 3, pp. 47-59 (2019)

Toko, Y., Sato-Ilic, M.: Improvement of the training dataset for supervised multiclass classification, Czarnowski, I., Howlett, R.J., Jain, L. C. (Eds.), Smart Innovation, Systems and Technologies, 193, pp. 291-302. Springer, Singapore (2020)

Proposed method

Step 1 Tokenize each text description into words by Sudachi

Step 2 Obtaining numerical vectors corresponding to words applying Word2Vec

1. Produce dataset for word2vec concatenating all tokenized words consecutively.
2. Train word2vec models to the produced dataset.

Each unique word in the dataset will be assigned a corresponding numerical vector

Step 3 Normalize each feature of the obtained set of numerical vectors

Step 4 Produce sentence vectors for each text description based on the normalized vectors

1. Obtain a corresponding numerical vector for each word in each text description from the set of normalized numerical vectors
2. Calculate the sum of numerical vectors for each text description

Proposed method

Step 5 Apply fuzzy c-means method

1. Fuzzy c-means method is applied to sentence vectors produced in step 4 to classify them into K clusters. The number of clusters K is given in advance.
2. A whole data with obtained sentence vectors is divided according to the degree of belongingness of data to clusters.

Step 6 Assign corresponding class applying SVM for each dataset of each cluster

1. Train a support vector machine for each cluster.
2. Predict a corresponding class for each target text description.

Determine the followings for training a support vector machine

- Cost parameter
- Kernel function to be applied
- Gamma parameter if radial basis function kernel is applied as a kernel function
- Type of methods: one-versus-one or one-versus- the-rest

Proposed method

Step 7 Extract datasets that assigned classes with low classification accuracy in step 6

Dataset that assigned classes with high classification accuracy in step 6 are accepted as classification results.

Step 8 Perform step 5 thorough step 7 iteratively to the extracted dataset in step 7 until obtaining enough better result

Step 9 Extract unmatched data

Step 10 Implement re-classification based on reliability score to the extracted unmatched data

Numerical Example

Data

Dataset : Family Income and Expenditure Survey

Data size : approx. 810 thousand instances (or sentences)

Contents : purchased items names in Japanese and
corresponding codes (approx. 520 different codes are available)

Implementation

We have confirmed our python code is run in R utilizing "reticulate" package that provides a comprehensive set of tools for interoperability between R and python.

The following python libraries are applied:

- **gensim : training a word2vec model**
  *Type of model architecture: skip-gram model *Number of vector dimensions: 100
  *Number of training iterations: 10,000 *Window size =2
- **skfuzzy : fuzzy c-means clustering**
  *Number of clusters: 2 *m parameter: 1.1
  *Error rate: 0.0001 *Maximum number of iterations:10,000
- **scikit-learn : normalization, grid search**, and **training support vector machines**
  *kernel: radial basis function kernel
  *cost parameter and gamma parameter: selected by grid search

Numerical Example

1st iteration: Classification accuracy of each cluster

Cluster label	Number of text descriptions				Accuracy	SVM	
	Total	Train	Test	Correctly assigned		Cost	Gamma
C1	566,355	509,719	56,636	48,596	0.858	10	0.0012
C2	248,136	223,322	24,814	22,762	0.917	100	0.001
Total	814,491	733,041	81,450	71,358	0.876		

accept

Implement 2nd iteration to cluster C1 that has lower classification accuracy.

2nd iteration: Classification accuracy of each cluster

Cluster label	Number of text descriptions				Accuracy	SVM	
	Total	Train	Test	Correctly assigned		Cost	Gamma
C1_1	249,261	224,334	24,927	19,635	0.788	10	0.0012
C1_2	317,094	285,384	31,710	28,488	0.898	10	0.0012
Total	566,355	509,718	56,637	48,123	0.850		

accept

Implement 3rd iteration to cluster C1_1 that has lower classification accuracy.

3rd iteration: Classification accuracy of each cluster

Cluster label	Number of text descriptions				Accuracy	SVM	
	Total	Train	Test	Correctly assigned		Cost	Gamma
C1_1_1	88,614	79,752	8,862	7,420	0.837	10	0.0012
C1_1_2	160,647	144,582	16,065	12,275	0.764	10	0.0012
Total	249,261	224,334	24,927	19,695	0.790		

accept

Implement 4th iteration to cluster C1_1_2 that has lower classification accuracy.

Numerical Example

4th iteration: Classification accuracy of each cluster

Cluster label	Number of text descriptions				Accuracy	SVM	
	Total	Train	Test	Correctly assigned		Cost	Gamma
C1_1_2_1	61,398	55,258	6,140	5,556	0.905	10	0.0012
C1_1_2_2	99,249	89,324	9,925	6,700	0.675	10	0.0012
Total	160,647	144,582	16,065	12,256	0.763		

accept

accept

Summary

Cluster label	Number of text descriptions				Accuracy
	Total	Train	Test	Correctly assigned	
C2	248,136	223,322	24,814	22,762	0.917
C1_2	317,094	285,384	31,710	28,488	0.898
C1_1_1	88,614	79,752	8,862	7,420	0.837
C1_1_2_1	61,398	55,258	6,140	5,556	0.905
C1_1_2_2	99,249	89,324	9,925	6,700	0.675
Total	814,491	733,040	81,451	70,926	0.871

Comparison of classification accuracy of the proposed method and the previously proposed method

Classification method	Accuracy
Combined method of multi-class SVM by **fuzzy c-means method** and the reliability score (Propsed method)	0.922
Combined method of multi-class SVM by **k-means method** and the reliability score (Previously propsed method)	0.919

hybrid

Note that, the previously proposed method applied meCab for tokenization whereas the proposed method applied Sudachi.

Numerical Example

Comparison of classification accuracy of SVM by fuzzy c-means method and SVM by k-means method

Classification accuracy of SVM by fuzzy c-means method (proposed method) (1st iteration)

Cluster label	Number of text descriptions				Accuracy	SVM	
	Total	Train	Test	Correctly assigned		Cost	Gamma
C1	566,355	509,719	56,636	48,596	0.858	10	0.0012
C2	248,136	223,322	24,814	22,762	0.917	100	0.001
Total	814,491	733,041	81,450	71,358	0.876		

Classification accuracy of SVM by k-means method (previously proposed method)

Cluster	Number of text descriptions				Accuracy	SVM	
	Total	Training	Evaluation	Correctly assigned		cost	gamma
Cluster 1	4,568	4,111	457	375	0.821	30	0.0001
Cluster 2	37,454	33,708	3,746	3,719	0.993	100	0.0010
Cluster 3	137,157	123,441	13,716	12,068	0.880	30	0.0010
Cluster 4	148,585	133,726	14,859	12,909	0.869	10	0.0064
Cluster 5	38,003	34,202	3,801	3,082	0.811	10	0.0010
Cluster 6	31,288	28,159	3,129	3,116	0.996	100	0.0010
Cluster 7	275,852	248,266	27,586	22,929	0.831	90	0.0255
Cluster 8	47,421	42,678	4,743	4,243	0.895	90	0.0001
Cluster 9	48,332	43,498	4,834	4,093	0.847	100	0.0010
Cluster 10	45,831	41,247	4,584	3,672	0.801	10	0.0010
Total	814,491	733,036	81,455	70,206	0.862		

Note that, the previously proposed method applied MeCab for tokenization whereas the proposed method applied Sudachi.

Conclusion

We propose a new autocoding method that is a combined method of SVM and FCM with the reliability score.

Data is obtained as receipts digitally which include complex representation of text descriptions.

- Utilizing FCM(Fuzzy c-means) that is a computational intelligence method known as high performance dealing with cognitive uncertainty with linguistically motivated computation.

- Utilizing SVM that is a machine learning method known as high generalization performance.

The numerical example shows a better performance of the proposed method with the Family Income and Expenditure Survey.

Utilizing "reticulate" package in R, our python code run in R easily.

References

1. Hacking, W., Willenborg, L.: Method Series Theme: Coding; interpreting short descriptions using a classification, Statistics Methods. Statistics Netherlands (2012) Available at: https://www.cbs.nl/en-gb/our-services/methods/statistical-methods/throughput/throughput/coding, last accessed 2021/10/19
2. Gweon, H., Schonlau, M., Kaczmirek, L., Blohm, M., Steiner, S.: Three Methods for Occupation Coding Based on Statistical Learning, Journal of Official Statistics, 33(1), pp. 101-122 (2017)
3. Toko, Y., Wada, K., Iijima, S., Sato-Ilic, M.: Supervised Multiclass Classifier for Autocoding Based on Partition Coefficient, Czarnowski, I., Howlett, R.J., Jain, L. C., and Vlacic, L. (eds.), Smart Innovation, Systems and Technologies, 97, pp. 54-64. Springer, Switzerland (2018a)
4. Toko, Y., Iijima, S., Sato-Ilic, M.: Overlapping Classification for Autocoding System, Journal of Romanian Statistical Review, 4, pp. 58-73 (2018b)
5. Statistics Bureau of Japan: Outline of the Family Income and Expenditure Survey. Available at: https https://www.stat.go.jp/english/data/kakei/1560.html, last accessed 2021/10/18
6. Toko, Y., Iijima, S., Sato-Ilic, M.: Generalization for Improvement of the Reliability Score for Autocoding, Journal of Romanian Statistical Review, 3, pp. 47-59 (2019)
7. Toko, Y., Sato-Ilic, M.: Improvement of the training dataset for supervised multiclass classification, Czarnowski, I., Howlett, R.J., Jain, L. C. (Eds.), Smart Innovation, Systems and Technologies, 193, pp. 291-302. Springer, Singapore (2020)
8. Toko, Y., Sato-Ilic, M.: Efficient Autocoding Method in High Dimensional Space, Romanian Statistical Review, 1, pp. 3-16 (2021)
9. Toko, Y., Sato-Ilic, M : A Hybrid Method of Multi-Class SVM and Classification Method Based on Reliability Score for Autocoding of the Family Income and Expenditure Survey, Czarnowski, I., Howlett, R.J., Jain, L. C. (Eds.), Smart Innovation, Systems and Technologies, 238, pp. 403-414. Springer, Singapore (2021)
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press (2000)
11. Bezdeck, J.C., Ehrlich, R., Full, W. : FCM: Fuzzy C-Means Algorithm. Computers and Geoscience, 10(2–3), pp. 191–203 (1984)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013)
13. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model, Journal of Machine Learning Research, 3, pp. 1137-1155 (2003)
14. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
15. Bezdek, J.C., Keller J., Krisnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers (1999)
16. Menger, K.: Statistical metrics, Proceedings of the National Academy of Sciences of the United States of America, 28, pp. 535-537 (1942)
17. Mizumoto, M.: Pictorical representation of fuzzy connectives, Part I: Cases of *T*-norms, t-Conorms and Averaging Operators, Fuzzy Sets and Systems, 31, pp. 217-242 (1989)
18. Schweizer, S., Sklar, A.: Probabilistic metric spaces. Dover Publications, New York (2005)
19. Takaoka, K., Hisamoto, S., Kawahara, N., Sakamoto, M., Uchida, Y., Matsumoto, Y.,: Sudachi: a Japanese Tokenizer for Business, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC) , May 2018, Miyazaki, Japan, pp. 2246-2249, European Language Resources Association (2018)
20. Statistics Bureau of Japan: Income and Expenditure Classification Tables (revised in 2020). Available at: https://www.stat.go.jp/english/data/kakei/ct2020.html, last accessed 2021/10/18

References

21. Ushey, K., Allaire JJ, Tang Y.,: reticulate: Interface to ‘python’, R package version 1.22, https://CRAN.R-project.org/package=reticulate (2021)

22. Rehurek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora, Proceedings of LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45-50 (2010)

23. scikit-fuzzy development team: skfuzzy. Available at: https://pythonhosted.org/scikit-fuzzy, last accessed 2021/11/05

24. Josh Warner; Jason Sexauer; scikit-fuzzy; twmeggs; alexsavio; Aishwarya Unnikrishnan; et al., JDWarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2.

25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830 (2011)