A Appendix

A.1 Preliminaries

A.1.1 Preliminaries of Cognitive Science

Theories in cognitive science about how concepts are represented includes classical theory, prototype theory, exemplar theory and knowledge theory [13].

(1) Classical theory The classical theory can traced back to Aristotle. In it, any concept can be precisely defined.

However, modern cognitive science has shown that concepts are almost impossible to define precisely in the real world.

(2) Prototype theory In prototype theory, a concept can be represented by a prototype, but how to give a prototype corresponding to a concept is still a very difficult problem.

(3) Exemplar theory Unlike classical theory and prototype theory, exemplar theory argues that no definition or prototype can fully encompass the entire concept. Conversely, the representation of a concept in the human mind is the set of objects in human memory that belong to the concept.

(4) Knowledge theory Knowledge theory holds that concepts are part of our common sense knowledge about the world. Human are unable to learn a concept in isolation from anything else. On the contrary, the process by which human learn concepts is part of all our understanding of the world around us.

From the perspective of cognitive science, knowledge theory seems to be the most complete, and a large number of cognition experimental results can support it. However, knowledge theory is difficult to implement for designing a purely data-driven learning machine. Therefore, exemplar theory is chosen to solve the problem of concept representation in this paper. The reason is that it not only has good self-consistency in cognitive science, but also it can be naturally integrated with the data-driven machine learning paradigm.

(5) Concept is fuzzy. The number of objects in the real world is far greater than the number of concepts, so a small number of concepts can never map perfectly onto all objects. At the same time, objects belonging to the same concept may have certain differences, and objects belonging to different concepts may also have certain similarities. Therefore, concept must be of fuzziness. See page 21 of literature [13] for a more detailed discussion.

A.1.2 Preliminaries of Binary Relation

Definition 5 ([32]) *Given two sets A, B, a map $\mathrm{R} : A \times B \to \{0, 1\}$ is called a binary relation from A to B.*

Definition 6 ([32]) *Given a set A, a map $\mathrm{R} : A \times A \to \{0, 1\}$ is called a binary relation on A.*

Definition 7 ([32]) *Given a set A and a binary relation R on A, if R satisfies the following three properties:*
(1) Reflexivity: $\forall a \in A, \mathrm{R}((a, a)) = 1$,
(2) Symmetry: $\forall a, b \in A$, if $\mathrm{R}((a, b)) = 1$, then $\mathrm{R}((b, a)) = 1$,
(3) Transitivity: $\forall a, b, c \in A$, if $\mathrm{R}((a, b)) = 1$ and $\mathrm{R}((b, c)) = 1$, then $\mathrm{R}((a, c)) = 1$,
then R is called a equivalence relation (ER) on A.

Remark: To unify the description of the full paper, the above definitions are given in a different form from the literature [32], but they are equivalent mathematically.

Definition 8 ([32]) *Given two sets A and $P = \{P_1, P_2, \cdots, P_m\}, m \geq 1$, if P satisfies the following three properties:*
(1) $\forall P_i \in P$, $P_i \neq \phi$,
(2) $\cup_{P_i \in P} = A$,
(3) $\forall P_i, P_j \in P$, if $i \neq j$, then $P_i \cap P_j = \phi$,
then P is called a partition on A.

Definition 9 ([32]) *Given a set A and a equivalence relation R on A. $\forall a \in A$, $[a]_{\mathrm{R}} = \{b | b \in A, \mathrm{R}((a,b)) = 1\}$ is called the equivalence class of a derived by R.*

Definition 10 *Given a set A and a equivalence relation R on A. Obviously, the set $A/\mathrm{R} = \{[a]_{\mathrm{R}} | a \in A\}$ is a partition on A, and it is called the partition on A derived by R.*

A.1.3 Preliminaries of Fuzzy Set

Definition 11 ([33]) *Given a set A, a map $\mu : A \to [0,1]$ is called a fuzzy set on A. And $\forall a \in A$, the $\mu(a)$ is called the membership degree of a to the fuzzy set μ.*

Definition 12 ([33]) *Given two sets A, B, a map $\mathrm{F} : A \times B \to [0,1]$ is called a binary fuzzy relation from A to B.*

Definition 13 ([33]) *Given a set A, a map $\mathrm{F} : A \times A \to [0,1]$ is called a binary fuzzy relation on A.*

Definition 14 *Given a finite set A and a binary fuzzy relation F on A. Without loss of generality, let $A = \{a_1, a_2, \cdots, a_{|A|}\}$, then the F can be described equivalently as a binary fuzzy relation matrix: $\boldsymbol{F} \in [0,1]^{|A| \times |A|}$, $f_{ij} = \mathrm{F}((a_i, a_j)), \forall i, j = 1, 2, \cdots, |A|$.*

Obviously, the binary fuzzy relation F on A is a fuzzy set on A's Cartesian product $A \times A$. As can be seen from the definition, the binary fuzzy relation is a extension of the binary relation described in **Definition** 6.

Definition 15 ([33]) *Given a set A and a binary fuzzy relation F on A. If the F satisfies*
(1) Reflexivity: $\forall a \in A, \mathrm{F}((a,a)) = 1$,
(2) Symmetry: $\forall a, b \in A, \mathrm{F}((a,b)) = \mathrm{F}((b,a))$,
then it's called a fuzzy similarity relation (FSR) on A.

Definition 16 ([33]) *Given a set A and a fuzzy similarity relation F on A. If the F satisfies*
Transitivity: $\forall a, b \in A, \mathrm{F}((a,b)) \geq \max_{c \in A} \min[\mathrm{F}((a,c)), \mathrm{F}((c,b))]$,
then it's called an fuzzy equivalence relation (FER) on A.

As can be seen from the **Definition** 16, the fuzzy equivalence relation (FER) and the equivalence relation (ER) described in the **Definition** 7 are highly correlated. The following theorem clearly shows the relationship between them.

Definition 17 ([34]) *Given a set A and a binary fuzzy relation F on A. $\forall \lambda \in [0,1]$, the binary relation*

$$\forall a, b \in A,\ \mathrm{F}^{[\lambda]}((a,b)) = \mathbb{I}[\mathrm{F}((a,b)) \geq \lambda]$$

is called the λ-cut relation of F.

Theorem 3 *Given a set A and a fuzzy equivalence relation F on A. $\forall \lambda \in [0,1]$, the λ-cut relation $\mathrm{F}^{[\lambda]}$ is an equivalence relation on A. (The proof process is straightforward and omitted.)*

Definition 18 ([20]) *Given a set A and two binary fuzzy relations F and G on A. The product (composition) binary fuzzy relation of them $\mathrm{H} = \mathrm{F} \otimes \mathrm{G}$ is defined as*

$$\forall a, b \in A,\ \mathrm{H}((a,b)) = \max_{c \in A} \min[\mathrm{F}((a,c)), \mathrm{G}((c,b))].$$

Definition 19 *Based on the **Definition** 18, given a finite set A and a binary fuzzy relation F on A, the power of F can be written as $\mathrm{F}^1 = \mathrm{F}$, $\mathrm{F}^2 = \mathrm{F} \otimes \mathrm{F}$, $\mathrm{F}^3 = \mathrm{F}^2 \otimes \mathrm{F}$, $\cdots$.*
*Let $\boldsymbol{F} \in [0,1]^{|A| \times |A|}$ be the binary fuzzy relation matrix of F (see **Definition** 14), then the power of fuzzy relation matrix $\boldsymbol{F}$ can be written as $\boldsymbol{F}^1 = \boldsymbol{F}$, $\boldsymbol{F}^2 = \boldsymbol{F} \otimes \boldsymbol{F}$, $\boldsymbol{F}^3 = \boldsymbol{F}^2 \otimes \boldsymbol{F}$, $\cdots$.*

Definition 20 *Given a finite set A, and a binary fuzzy relation F on A. G is the transitive-closure of F if and only if G satisfies*
(1) G is transitive,
(2) $\mathrm{F} \subseteq \mathrm{G}$, i.e. $\forall a, b \in A$, $\mathrm{F}((a,b)) \leq \mathrm{G}((a,b))$,
(3) if $\mathrm{F} \subseteq \mathrm{H}$ and H is transitive, then $\mathrm{G} \subseteq \mathrm{H}$.

In **Definition** 5.1 of reference [35], a more general definition, P-closure, was given, where P can be one of many properties.

Theorem 4 *Given a set A and a binary fuzzy relation F on A. The transitive-closure of F exists and is unique. (It is a direct corollary to the **Theorem** 5.3 of literature [35].)*

Theorem 5 *Given a finite set A, and a fuzzy similarity relation S on A. The transitive-closure of S can be obtained by*
*(1) Compute S^{2^0}, S^{2^1}, S^{2^2}, $\cdots$, by **Definition** 19, until $\mathrm{S}^{2^{k-1}} = \mathrm{S}^{2^k}$;*
(2) Then $\mathrm{S}^{2^{k-1}}$ is the transitive-closure of S and $2^{k-1} \leq |A|$.
The above process converts a Fuzzy Similarity Relation into a Fuzzy Equivalence Relation and is abbreviated as $\mathrm{T} = \mathrm{t}_{\mathrm{FSR2FER}}(\mathrm{S})$.
*(It is a direct corollary to the **Theorem** 8.2 of literature [35].)*

Remark: Given a finite set A, and a binary fuzzy relation F on it. According to **Definition** 18, the time complexity on calculating $\mathrm{F} \otimes \mathrm{F}$ is $O\left(|A|^3\right)$. So the time complexity of the $\mathrm{t}_{\mathrm{FSR2FER}}(\mathrm{F})$ is $O\left(|A|^3 \log_2 |A|\right)$ according to **Theorem** 5.

A.2 Related Work

A.2.1 Relationship to Existing Classifiers

So far, dozens of different classifiers have been designed. The literature [1] systematically summarizes existing classifiers and divides them into 17 families. The 17 families are: (1) Discriminant analysis, (2) Bayesian, (3) Neural networks, (4) Support vector machines, (5) Decision trees, (6) Rule-based classifiers, (7) Boosting, (8) Bagging, (9) Stacking, (10) Random forests, (11) Other ensembles, (12) Generalized linear models, (13) Nearest neighbors, (14) Partial least squares and principal component regression, (15) Logistic and multinomial regression, (16) Multiple adaptive regression splines, and (17) Other methods, which contains almost all existing classifiers.

The goal of the existing classifiers is to learn a function $\mathrm{f} : \mathcal{X} \to \mathcal{Y}$, which is an estimation of the posterior probability $\mathrm{p}(y|x), \forall x \in \mathcal{X}, y \in \mathcal{Y}$ from the point of view of probability theory. According to how to estimate the posterior probability, the existing classifiers can be divided as: discriminative methods, generative methods, and ensemble learning methods.

The discriminative methods model the $\mathrm{p}(y|x)$ directly and the typical examples include Support Vector Machine [36], Decision Tree [37], Deep Neural Network [38] and so on.

The generative methods estimate the $\mathrm{p}(y|x)$ indirectly by modeling the $\mathrm{p}(y, x)$. And the typical examples include Naive Bayes [16, 39], Bayes-Network [17] and so on.

The ensemble learning methods obtain the final $\mathrm{p}(y|x)$ by integrating multiple base classifiers, where each base classifier can be discriminative method or generative method. And the typical examples include AdaBoosting [40], Random forest [41] and so on.

The relationship between existing classifiers and FLM can be discussed from the following aspects.

1. **Existing classifiers and FLM solve classification problems from different perspectives.** Given an $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem (see **Definition** 1), the goal of existing classifiers is to learn the mapping $\mathrm{f} : \mathcal{X} \to \mathcal{Y}$ to approximate the unknown target function φ. Unlike it, the goal of FLM is to learn a mapping $\mathrm{f}_{\mathrm{FER}} : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ to approximate the equivalence relation derived by the target function R_φ (see **Definition** 21). It is motivated by that concept is represented on the bias of similarity, a conclusion of cognitive science. And, it is proved that any classification problem can be solved by the this approach (see **Proposition** 1).
2. **Most of the existing classifiers and FLM deal with the fuzziness in opposite ways.** The fuzziness is an intrinsic property of concept. However, most of existing classifiers ignore or even unintentionally eliminate it. Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, the 0-1 loss, $\mathbb{I}\left(\mathrm{f}(x) = \varphi(x)\right)$, has always been regarded as the ideal infallible loss function. However, it does not tolerate the fuzziness. In order to simplify the solving process, some surrogate loss functions with elegant mathematical properties are designed, such as the exponential loss
$$\exp(-\varphi(x)\mathrm{f}(x)),$$

where $\mathcal{Y} = \{-1, +1\}, \forall x \in \mathcal{X}, \mathrm{f}(x) \in \mathbb{R}$ and the cross entropy loss

$$-\sum_{i \in \mathcal{Y}} \mathbb{I}\left(\varphi(x) = i\right) \log(\mathrm{f}(x)_i),$$

where $\mathcal{Y} = \{1, 2, \cdots\}, \forall x \in \mathcal{X}, \mathrm{f}(x) \in [0,1]^{|\mathcal{Y}|}, \sum_{i \in \mathcal{Y}} \mathrm{f}(x)_i = 1$, etc. Based on these loss functions, some classical classifiers are built, such as AdaBoosting [40], Deep Neural Network [38], etc. However, minimizing these losses indirectly removes the fuzziness. In contrast, in NN-FLM, a fuzziness permissible loss function (see formula (6)) is designed, and the fuzziness is deliberately preserved during the learning process.

3. **Existing classifiers can be embedded into FLM.** FLM is a general framework and almost all existing classifiers can be embedded into it. It enables existing classifiers designed for binary-classification problem (e.g. Support Vector Machine [36]) to be directly used for multi-class classification problems without any modification (see **Proposition** 1). Different from the conventional strategies for converting multi-class classification problems into a series of binary-classification problems such as 'one-versus-one' and 'one-versus-rest' [42], the new strategy obtains only one binary-classification problem. That is to say, only one binary classifier is needed to solve the original multi-class classification problem.

Relationship to Nearest Neighbor Classifier

Nearest Neighbor Classifier (NNC) [43] is a classical family of classifiers that predicts the class label for a test sample through finding it's nearest neighbors. To date, there are still studies on improving it [44]. The differences between the NNC and FLM mainly include:

1. **They are different in the choice of similarity (distance).** In NNC, a predefined similarity is usually used. Therefore, performance of NNC depends heavily on the selection of the similarity, and it is difficult to adaptively handle different tasks. Unlike NNC, in FLM, a special similarity (FER) is learned from the training data (see formula (1)). This allows FLM to automatically adapt to different tasks.
2. **They represent concept in different ways.** In NNC, all samples are stored to represent every concept in $\mathcal{Y}$. Unlike NNC, in FLM, every concept in $\mathcal{Y}$ is represented by the selected exemplar set (see **Definition** 4). Although in exemplar theory, how many exemplars are stored for a concept in human memory is a leftover open question. Obviously, in order to form representation of concept 'dog', human do not remember all the dogs they have ever seen. What's more, storing all the training data for prediction is not a sensible approach from the perspective of time and space complexity, especially when the number of training samples is large.

Relationship to Existing Fuzzy Classifiers

In literature [21], the fuzzy classifier (FC) is defined as a classifier that uses fuzzy sets or fuzzy logic in the course of its training or operation. FCs can deal with ambiguity effectively, has strong interpretability and can easily be fused with the knowledge of experts. According to this definition, FLM is a kind of FC. The differences between FLM and the existing FCs are discussed as follows. (In order to be clearer, we take NN-FLM as an example.)

In literature [21], the existing FCs are divided into fuzzy if-then and non if-then fuzzy classifiers.

For fuzzy if-then classifiers, fuzzy sets need to be defined on each feature to construct classification rules. The number of fuzzy sets and the membership function of each fuzzy set need to be set manually. When the semantic information of features are unknown, this step is difficult to complete effectively. In some case, although the semantic information of features are known, it is almost meaningless to define fuzzy sets on them. For example, in image classification tasks, the feature of the sample is pixel. Because a single pixel often can not express high-level semantic information, the fuzzy set defined on a single pixel is often useless for classification. At the same time, defining fuzzy sets on each feature also faces efficiency problem. Assuming that the sample is image with 256×256 pixels and only 3 fuzzy sets are defined on each pixels. Then there are $3 \times 256 \times 256$ fuzzy sets to be processed and $(256 \times 256)^3$ rules will be generated, which brings a huge computational burden to training and test of classifiers.

Different from fuzzy if-then classifiers, firstly, the design of NN-FLM does not rely semantic information of features. Secondly, thanks to the excellent feature extraction ability of deep neural

network, NN-FLM can effectively extract useful features for classification from low-level semantic features. Thirdly, by skillfully designing the optimization model, NN-FLM can effectively complete training and prediction. Therefore, NN-FLM is suitable for learning from large-scale data.

Non if-then fuzzy classifiers mainly include fuzzy k-nearest neighbor (FKNN) and fuzzy-prototype (FPC) classifiers. FKNN is an extension of KNN and uses the distances to the neighbors as well as their soft labels for predicting. Unlike FKNN, FPC uses the nearest class prototype instead of the nearest training sample to predict. Among these methods, the distance between samples, the construction of class prototype, and the soft labels all directly affect the performance. All these depend on manual construction, so it is difficult to give a reasonable design when there is a lack of domain experts.

Unlike FKNN and FPC, NN-FLM takes similarity learning as the core goal. With the excellent feature extraction ability and well-designed loss, the adaptive similarity learning is realized. Meanwhile, the exemplar selection method can automatically select samples from training set without the participation of human experts.

In conclusion, the NN-FLM is more suitable for data-driven learning tasks.

A.2.2 Relationship to Distance Metric Learning

The goal of the distance metric learning [22, 23, 24] is to learn a task specific distance function,

$$\mathrm{d} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+, \tag{10}$$

to improve the performance of the model on the corresponding task. Unlike it, the goal of FLM is to learn an FER (see formula (1)), $\mathrm{f}_{\mathrm{FER}} : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ to solve original $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem. In fact, let

$$\mathrm{d}_{\mathrm{FER}} : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]\,,\ \mathrm{d}_{\mathrm{FER}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = 1 - \mathrm{f}_{\mathrm{FER}}\left(\mathbf{x}_i, \mathbf{x}_j\right),\ \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \subseteq \mathbb{R}^d. \tag{11}$$

Then $\mathrm{d}_{\mathrm{FER}}$ can also be viewed as a distance function on $\mathbb{R}^d$.

The differences between formula (10) and (11) include:

1. The $\mathrm{d}_{\mathrm{FER}}$ is a distance metric on $\mathbb{R}^d$ (i.e. $\mathrm{d}_{\mathrm{FER}}$ satisfies the basic properties of the distance metric: nonnegativity, identity of indiscernibles, symmetry, and triangle inequality, see **Theorem** 2.3 of literature [34]) while d doesn't have to be a distance metric on $\mathbb{R}^d$. For example, the Mahalanobis-like distance, $\mathrm{d}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sqrt{\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{M}\left(\mathbf{x}_i - \mathbf{x}_j\right)}$, is commonly used, where $\mathbf{M} \in \mathbb{R}^{d \times d}$ be the learnable parameter. Usually, $\mathbf{M}$ is a positive semi-definite matrix, which will not guarantee that the d satisfies the identity of indiscernibles.
2. Further, $\mathrm{d}_{\mathrm{FER}}$ is a normalized isosceles distance (see **Definition** 2.18 and **Proposition** 2.13 of literature [34]). In addition to the basic properties of the distance metric, it also satisfies two other properties:
   (a) $\mathrm{d}_{\mathrm{FER}}$ is normalized, i.e. $\forall x_i, x_j \in \mathcal{X}, 0 \leq \mathrm{d}_{\mathrm{FER}}\left(x_i, x_j\right) \leq 1$,
   (b) $\mathrm{d}_{\mathrm{FER}}$ is isosceles, i.e. $\forall x_i, x_j, x_k \in \mathcal{X}$, the triangle composed by $\mathrm{d}_{\mathrm{FER}}\left(x_i, x_j\right)$, $\mathrm{d}_{\mathrm{FER}}\left(x_i, x_k\right)$ and $\mathrm{d}_{\mathrm{FER}}\left(x_j, x_k\right)$ is an isosceles triangle, and its congruent legs are the longest side.

   The first property is derived from the definition of the binary fuzzy relation (see **Definition** 13 in **Appendix** A.1.3), and the second one is derived from the transitivity of FER (see **Definition** 16 in **Appendix** A.1.3). These two properties are crucial for classification problem, because the target function of the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is transitive (see **Definition** 21). Unfortunately, distance metric learning ignores these two crucial properties, so it can not capture the nature of classification.

A.2.3 Relationship to Siamese Network

Siamese network [25, 26, 27] is a widely used neural network that computes the similarity between two objects. Siamese network consists of two identical neural networks to learning the hidden representation of an input vector. These two neural networks work parallelly in tandem and compare their outputs at the end.

Essentially, siamese network can be viewed as a special kind of distance metric learning method. Therefore, the relationship between siamese network and FLM is the same as the relationship between distance metric learning and FLM, which will not be repeated here.

A.2.4 Relationship to Relation Network Based Methods

Recently, methods based on 'relation' network [28, 29, 30] have been proposed to solve the few-shot learning problem. The differences between these methods and FLM include:

1. The methods in the above literatures are designed for few-shot learning problems, while FLM is designed for general classification problems (see **Definition** 1). FLM can directly deal with few-shot learning problems almost without adjustment.
2. In the above literatures, the term 'relation' is only an intuitive expression, and there is no clear mathematical definition. In contrast, this paper adopts binary relation in classical set theory (see **Definition** 6 in **Appendix** A.1.2) and binary fuzzy relation (see **Definition** 13 in **Appendix** A.1.3) in fuzzy set theory. The structure of classification problems is studied with these tools and the equivalence between ER and classification problems is demonstrated (see **Proposition** 1).
3. In the above literatures, the relation network can be formally written as: $\mathrm{r} : \mathcal{X} \times \mathcal{X} \to [0, 1]$. According to the theoretical system adopted in this paper, its mathematical meaning should be that r is a binary fuzzy relation on $\mathcal{X}$ (see **Definition** 13 in **Appendix** A.1.3). And there is no special requirement for r, such as reflexivity, symmetry, transitivity, etc. In contrast, the core component of FLM is $\mathrm{f}_{\mathrm{FER}} : \mathcal{X} \times \mathcal{X} \to [0, 1]$, where $\mathrm{f}_{\mathrm{FER}}$ is an FER on $\mathcal{X}$ (see **Definition** 16 in **Appendix** A.1.3).
4. In the above literatures, the relation network is only used to calculate the relation score between the support samples (or the mean of support samples) and the query samples during the training process. It ignores the relation score between the support samples and the relation score between the query samples. The supervised information contained in these sample pairs is important for training classifiers. Especially in learning tasks such as few-shot learning, there are only a very small number of labeled samples. In contrast, in FLM, the sample pair composed of any two training samples must participate in the training process (see formula (8)), so as to ensure that a high-quality FER is learned.

A.3 Proofs

A.3.1 The Proof of Proposition 1

First, two useful definitions and a lemma are given as follows.

Definition 21 *Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, the binary relation (see **Definition** 6 in Appendix A.1.2) derived by the unknown target function φ is defined as*

$$\forall x_i, x_j \in \mathcal{X}, \mathrm{R}_{\varphi}\left(\left(x_i, x_j\right)\right) = \mathbb{I}\left(\varphi\left(x_i\right) = \varphi\left(x_j\right)\right). \tag{12}$$

*Obviously, R_{φ} is a equivalence relation (ER) on $\mathcal{X}$ (see **Definition** 7 in **Appendix** A.1.2).*

Definition 22 *Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, finding the R_{φ} is defined as the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem.*

Lemma 1 *The $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is equivalent to the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem, i.e. if one problem is solved, then the other problem will also be solved.*

Proof

The proof consists two parts.

(1) If the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is solved, then the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem will also be solved.

If the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is solved, then $\forall x \in \mathcal{X}$, the true class label $\varphi(x)$ is known. Then according to the formula (12), the R_{φ}, the binary relation derived by the target function φ, can be obtained directly. That is to say, the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem is solved.

(2) If the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem is solved, then the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem will also be solved.

If the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem is solved, i.e. R_φ is known, then the $\mathcal{X}/\mathrm{R}_\varphi = \left\{[x]_{\mathrm{R}_\varphi} | x \in \mathcal{X}\right\} = \{\mathcal{X}_i | i = 1, 2, \cdots, |\mathcal{Y}|\}$, the partition on $\mathcal{X}$ derived by R_φ, can be obtained by **Definition** 9, 10. Then, one can obtained the target function φ: $\forall x \in \mathcal{X}$, $\varphi(x) = c_i$, if $x \in \mathcal{X}_i \in \mathcal{X}/\mathrm{R}_\varphi$. That is to say, $\forall x \in \mathcal{X}$, the $\varphi(x)$ is found, i.e. the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is solved.

Combining (1) and (2), the lemma is proved. □

Based on the above conclusion, the proof of the **Proposition** 1 is given as follows.

Proof

The target function $\varphi^\dagger$ of the adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem is equivalent to R_φ, the binary relation derived by the target function of $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, because of

$$\forall x_1, x_2 \in \mathcal{X},\ \mathrm{R}_\varphi\left((x_1, x_2)\right) = \mathbb{I}\left[\varphi(x_1) = \varphi(x_2)\right] = \varphi^\dagger\left((x_1, x_2)\right).$$

That is to say, the adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem is equivalent to the $(\mathcal{X}, \mathcal{Y}, \varphi)$-ER problem.

According to **Lemma** 1, one can obtained that the adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem is equivalent to the original $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem. □

A.3.2 The Proof of Theorem 1

Proof

Reduction to absurdity.

For $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \in D_{\text{train}}$ and $y_1 \neq y_2$, let $([\mathbf{x}_1; \mathbf{x}_1], 1)$, $([\mathbf{x}_2; \mathbf{x}_2], 1)$, $([\mathbf{x}_1; \mathbf{x}_2], 0)$, and $([\mathbf{x}_2; \mathbf{x}_1], 0)$ be 4 composite labeled samples obtained by formula (3).

Assume that these four new sample are linearly separable, i.e. $\exists \mathbf{w} \in \mathbb{R}^{2d}, b \in \mathbb{R}$, such that

$$[\mathbf{w}_1; \mathbf{w}_2]^T [\mathbf{x}_1; \mathbf{x}_1] + b > 0, \tag{13a}$$
$$[\mathbf{w}_1; \mathbf{w}_2]^T [\mathbf{x}_2; \mathbf{x}_2] + b > 0, \tag{13b}$$
$$[\mathbf{w}_1; \mathbf{w}_2]^T [\mathbf{x}_1; \mathbf{x}_2] + b < 0, \tag{13c}$$
$$[\mathbf{w}_1; \mathbf{w}_2]^T [\mathbf{x}_2; \mathbf{x}_1] + b < 0, \tag{13d}$$

where $[\mathbf{w}_1; \mathbf{w}_2] = \mathbf{w}$, and $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$.

From formula (13a) and (13b), we have

$$(\mathbf{w}_1 + \mathbf{w}_2)^T \mathbf{x}_1 + b > 0, \tag{14a}$$
$$(\mathbf{w}_1 + \mathbf{w}_2)^T \mathbf{x}_2 + b > 0. \tag{14b}$$

Further, we have

$$(\mathbf{w}_1 + \mathbf{w}_2)^T (\mathbf{x}_1 + \mathbf{x}_2) + 2b > 0. \tag{15}$$

From formula (13c) and (13d), we have

$$\mathbf{w}_1^T \mathbf{x}_1 + \mathbf{w}_2^T \mathbf{x}_2 + b < 0, \tag{16a}$$
$$\mathbf{w}_1^T \mathbf{x}_2 + \mathbf{w}_2^T \mathbf{x}_1 + b < 0. \tag{16b}$$

Further, we have

$$(\mathbf{w}_1 + \mathbf{w}_2)^T (\mathbf{x}_1 + \mathbf{x}_2) + 2b < 0. \tag{17}$$

There is a contradiction between formula (15) and (17), so the assumption is false, i.e. these 4 composite labeled samples obtained by formula (3) are not linearly separable.

So the $\left(\mathbb{R}^{2d}, \{0, 1\}, \varphi^\dagger\right)$-classification problem with the samples defined as formula (3) is not linearly separable. □

A.3.3 The Proof of Theorem 2

First, a useful definition and a lemma are given as follows.

Definition 23 *Given a finite set A and a binary fuzzy relation matrix $\boldsymbol{R}$ on it, $\forall\lambda\in[0,1]$ the tuple $\left(A,\boldsymbol{R}^{(\lambda)}\right)$ is called as the λ-strictly-cut graph, where, A is the set of the nodes of the graph and λ-strictly-cut relation matrix, $\boldsymbol{R}^{(\lambda)}\in\{0,1\}^{|A|\times|A|}$, $\forall i,j=1,2,\cdots,|A|,\ r_{ij}^{(\lambda)}=\mathbb{I}\left(r_{ij}>\lambda\right)$, is the adjacency matrix of the graph.*
Obviously, if $\boldsymbol{R}$ is symmetric, then $\forall\lambda\in[0,1]$, the strictly-cut graph $\left(A,\boldsymbol{R}^{(\lambda)}\right)$ is undirected graph.

Lemma 2 *Given a finite set A and a fuzzy similarity relation matrix $\boldsymbol{S}$ on A, then $\forall l=1,2,3,\cdots$, $\forall i,j=1,2,\cdots,|A|$, and $\forall\lambda\in[0,1]$, $s_{ij}^{l}>\lambda$ if and only if $\exists$ path $\langle a_i,\cdots,a_j\rangle$ in graph $\left(A,\boldsymbol{S}^{(\lambda)}\right)$ and the length of it is l. Where s_{ij}^{l} is entry in the ith row and the jth column of matrix $\boldsymbol{S}^{l}$, and $\boldsymbol{S}^{l}$ is the the l-th power of $\boldsymbol{S}$ (see **Definition** 19).*

Proof

By mathematical induction.

(1) When $l=1$, $\forall i,j=1,2,\cdots,|A|$, and $\forall\lambda\in[0,1]$,

$$s_{ij}^{1}>\lambda\Leftrightarrow s_{ij}>\lambda\Leftrightarrow s_{ij}^{(\lambda)}=1\Leftrightarrow\exists\text{ path }\langle a_i,a_j\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\text{ and the length of it is 1.}$$

(2) When $l=2$, $\forall i,j=1,2,\cdots,|A|$, and $\forall\lambda\in[0,1]$,

$$\begin{aligned}s_{ij}^{2}>\lambda\ &\Leftrightarrow\max_{k=1,2,\cdots,|A|}\min\left[s_{ik},s_{kj}\right]>\lambda\\&\Leftrightarrow\exists\,k\in\{1,2,\cdots,|A|\},\min\left[s_{ik},s_{kj}\right]>\lambda\\&\Leftrightarrow\exists\,k\in\{1,2,\cdots,|A|\},s_{ik}>\lambda\wedge s_{kj}>\lambda\\&\Leftrightarrow\exists\text{ path }\langle a_i,a_k,a_j\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\text{ and the length of it is 2.}\end{aligned}$$

(3) When $l=L-1, L>3$, assuming that $\forall i,j=1,2,\cdots,|A|$, and $\forall\lambda\in[0,1]$,

$$s_{ij}^{L-1}>\lambda\Leftrightarrow\exists\text{ path }\left\langle a_i,a_{p_2},a_{p_3},\cdots,a_{p_{L-1}},a_j\right\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\text{ and the length of it is }L-1.$$

(4) When $l=L, L>3$,$\forall i,j=1,2,\cdots,|A|$, and $\forall\lambda\in[0,1]$,

$$\begin{aligned}s_{ij}^{L}>\lambda\ &\Leftrightarrow\max_{k=1,2,\cdots,|A|}\min\left[s_{ik}^{L-1},s_{kj}\right]>\lambda\\&\Leftrightarrow\exists\,k\in\{k=1,2,\cdots,|A|\},\min\left[s_{ik}^{L-1},s_{kj}\right]>\lambda\\&\Leftrightarrow\exists\,k\in\{k=1,2,\cdots,|A|\},s_{ik}^{L-1}>\lambda\wedge s_{kj}>\lambda\\&\Leftrightarrow\exists\,k\in\{k=1,2,\cdots,|A|\},\\&\quad\exists\text{ path }\left\langle a_i,a_{p_2},a_{p_3},\cdots,a_{p_{L-1}},a_k\right\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\wedge\\&\quad\exists\text{ edge }\langle a_k,a_j\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\\&\Leftrightarrow\exists\text{ path }\left\langle a_i,a_{p_2},a_{p_3},\cdots,a_{p_{L-1}},a_k,a_j\right\rangle\text{ in graph }\left(A,\mathbf{S}^{(\lambda)}\right)\text{ and the length of it is }L.\end{aligned}$$

Combining (1)-(4), the lemma is proved. □

Based on the above conclusion, the proof of **Theorem** 2 is given as follows.

Proof

The proof consists two parts.

(1) $\forall p,q=1,2,\cdots,n,\ y_p=y_q$,
if $\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{\alpha,\beta}\left(s_{ij},y_i,y_j\right)=0$, then $\mathcal{L}_{\alpha,\beta}\left(t_{pq},y_p,y_q\right)=0$.

$\because\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{\alpha,\beta}\left(s_{ij},y_i,y_j\right)=0$
$\therefore\mathcal{L}_{\alpha,\beta}\left(s_{pq},y_i,y_j\right)=\max\{\beta-s_{ij},0\}=0$
$\therefore s_{pq}\geq\beta$.

And
$\because$ $\mathbf{T}$ is the transitive-closure of $\mathbf{S}$
$\therefore t_{pq} \geq s_{pq}$ (see **Definition** 20)

$\therefore t_{pq} \geq s_{pq} \geq \beta$
$\therefore \mathcal{L}_{\alpha,\beta}\left(t_{pq}, y_p, y_q\right) = \max\{\beta - t_{ij}, 0\} = 0.$

(2) $\forall p, q = 1, 2, \cdots, n,\ y_p \neq y_q$,
if $\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{\alpha,\beta}\left(s_{ij}, y_i, y_j\right) = 0,$ then $\mathcal{L}_{\alpha,\beta}\left(t_{pq}, y_p, y_q\right) = 0.$

Reduction to absurdity.

Let $X_{\text{train}} = \{\mathbf{x}_i | \left(\mathbf{x}_i, y_i\right) \in D_{\text{train}}\}$ be the set of training samples and let $\left(X_{\text{train}}, \mathbf{S}^{(\alpha)}\right)$ be the corresponding α-strictly-cut graph.

Assume that $\exists p, q = 1, 2, \cdots, n,\ y_p \neq y_q,$ such that $\mathcal{L}_{\alpha,\beta}\left(t_{pq}, y_p, y_q\right) > 0.$

$\because \mathcal{L}_{\alpha,\beta}\left(t_{pq}, y_p, y_q\right) = \max\{t_{pq} - \alpha, 0\} > 0$
$\therefore t_{pq} > \alpha.$

According to **Theorem** 5, $\mathbf{T}$ can be written as the power of $\mathbf{S}$. Without loss of generality, let $\mathbf{T} = \mathbf{S}^K$. Then we have $t_{pq} = s_{pq}^K > \alpha$.

According to **Lemma** 2, we have

$$\begin{aligned} s_{pq}^K > \alpha \quad &\Leftrightarrow \quad \exists \text{ path } \langle \mathbf{x}_p, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \cdots, \mathbf{x}_{r_K}, \mathbf{x}_q \rangle \text{ with length } K \text{ in graph } \left(X_{\text{train}}, \mathbf{S}^{(\alpha)}\right) \\ &\Leftrightarrow \quad \exists\, \mathbf{x}_p, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \cdots, \mathbf{x}_{r_K}, \mathbf{x}_q \in X_{\text{train}}, \\ &\qquad \text{such that } s_{pr_2} > \alpha, s_{r_2 r_3} > \alpha, \cdots, s_{r_{K-1} r_K} > \alpha, s_{r_K q} > \alpha. \end{aligned}$$

$\because y_p \neq y_q$
$\therefore \exists \left(p^*, q^*\right) \in \{\left(p, r_2\right), \left(r_2, r_3\right), \cdots, \left(r_{K-1}, r_K\right), \left(r_K, q\right)\}$, such that $y_{p^*} \neq y_{q^*}$
$\therefore \mathcal{L}_{\alpha,\beta}\left(s_{p^*q^*}, y_{p^*}, y_{q^*}\right) = \max\{s_{p^*q^*} - \alpha, 0\} > 0$
$\therefore \sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{\alpha,\beta}\left(s_{ij}, y_i, y_j\right) \geq \mathcal{L}_{\alpha,\beta}\left(s_{p^*q^*}, y_{p^*}, y_{q^*}\right) = \max\{s_{p^*q^*} - \alpha, 0\} > 0$
It contradicts to the known condition.

Combining (1) and (2), we have

$$\begin{aligned} \sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{\alpha,\beta}\left(t_{ij}, y_i, y_j\right) &= \sum_{i=1}^{n}\sum_{j=1, y_i = y_j}^{n}\mathcal{L}_{\alpha,\beta}\left(t_{ij}, y_i, y_j\right) \\ &+ \sum_{i=1}^{n}\sum_{j=1, y_i \neq y_j}^{n}\mathcal{L}_{\alpha,\beta}\left(t_{ij}, y_i, y_j\right) \\ &= 0 + 0 \\ &= 0 \end{aligned},$$

i.e. the theorem is proved. □

A.4 Experiment Details

A.4.1 Experiment Details of Section 4.1

(1) Data set The MNIST handwritten digit data set [31], one of the most basic data sets used to test performance learning algorithm, is chosen to demonstrate how FLM works. It has a training set of 60,000 samples, and a test set of 10,000 samples, and each sample is a gray image of a handwritten digits. There are a total of 10 concepts, i.e. $0, 1, \cdots, 9$.

(2) Settings of NN-FLM

1. The feature extraction network (see formula (4)) is a 5-layer convolutional network [2]. It can be formally described as follows

$$\begin{aligned}
\mathbf{h}^{(1)} &= \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathbf{x};\mathbf{K}_1\right);\mathbf{w}_1,\mathbf{b}_1\right)\right)\\
\mathbf{h}^{(2)} &= \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathbf{h}^{(1)};\mathbf{K}_2\right);\mathbf{w}_2,\mathbf{b}_2\right)\right)\\
\mathbf{h}^{(3)} &= \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathbf{h}^{(2)};\mathbf{K}_3\right);\mathbf{w}_3,\mathbf{b}_3\right)\right)\\
\mathbf{h}^{(4)} &= \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathbf{h}^{(3)};\mathbf{K}_4\right);\mathbf{w}_4,\mathbf{b}_4\right)\right)\\
\mathbf{h}^{(5)} &= \mathrm{Softmax}\left(\mathrm{BN}\left(\mathrm{Linear}\left(\mathbf{h}^{(4)};\mathbf{K}_5\right);\mathbf{w}_5,\mathbf{b}_5\right)\right)
\end{aligned},$$

   where $\Theta = \{\ \mathbf{K}_1 \in \mathbb{R}^{48\times1\times7\times7}$, $\mathbf{w}_1 \in \mathbb{R}^{48}$, $\mathbf{b}_1 \in \mathbb{R}^{48}$, $\mathbf{K}_2 \in \mathbb{R}^{96\times48\times7\times7}$, $\mathbf{w}_2 \in \mathbb{R}^{96}$, $\mathbf{b}_2 \in \mathbb{R}^{96}$, $\mathbf{K}_3 \in \mathbb{R}^{144\times96\times7\times7}$, $\mathbf{w}_3 \in \mathbb{R}^{144}$, $\mathbf{b}_3 \in \mathbb{R}^{144}$, $\mathbf{K}_4 \in \mathbb{R}^{192\times144\times7\times7}$, $\mathbf{w}_4 \in \mathbb{R}^{192}$, $\mathbf{b}_4 \in \mathbb{R}^{192}$, $\mathbf{K}_5 \in \mathbb{R}^{10\times3072}$, $\mathbf{w}_5 \in \mathbb{R}^{10}$, $\mathbf{b}_5 \in \mathbb{R}^{10}$ $\}$ is the set of the learnable parameters.

2. For the loss (8), $\alpha = 0.2$, $\beta = 0.8$ is set to control the degree of the fuzziness.

3. The number of exemplars of every class (see **Definition** 4), n_{exe}, is set to 5.

4. A regularization term $\mathcal{R}(\Theta)$ is added into the loss (8) to control the complexity of the model,

$$\mathcal{R}(\Theta) = \frac{\gamma}{n_{\mathrm{para}}}\sum_{i=1}^{5}\|\mathbf{K}_i\|_F^2 + \|\mathbf{w}_i\|_2^2 + \|\mathbf{b}_i\|_2^2,$$

   where $n_{\mathrm{para}} = (48\times1\times7\times7+48+48)+(96\times48\times7\times7+96+96)+(144\times96\times7\times7+144+144)+(192\times144\times7\times7+192+192)+(10\times3072+10+10)$ is the number of learnable parameters and $\gamma = 0.1$ is the trade-off parameter.

5. The stochastic gradient descent method Adam [45] optimizer are used to train the model. The size of the batch is set to 2048. We stopped iterating until the loss values of 10 consecutive epochs do not change significantly.

(3) Supplementary analysis of experimental results

The top-1, top-2, and top-3 accuracy of NN-FLM are 99.69%, 99.95%, and 99.98%, respectively.

It should be pointed out that the accuracy obtained in the experiment is slightly lower than the current state-of-the-ar on MNIST data[3]. On the one hand, the goal of this experiment is to illustrate the working mechanism of NN-FLM, rather than pursuing higher accuracy. In most of the state-of-the-art methods, strategies (such as designing specific neural network structure, data augmentation, and ensemble learning, etc) are used to achieve higher accuracy. These strategies can be embedded in NN-FLM for higher accuracy. On the other hand, when the accuracy reaches a certain level, pursuing higher accuracy is not only meaningless, but will lead to other problems. Suppose there is a classifier that predicts the 'correct' labels for all the images in Figure 3c. When it is applied to a field where the cost of misclassification is extremely high (e.g. the medical field), a serious consequences may be caused. In this case, it would be wiser to leave the fuzzy images to human as NN-FLM does.

A.4.2 Experiment Details of Section 4.2

(1) Data sets A total of 121 data sets are used in the experiments. In these data sets, the number of samples varies from 10 (trains) to 130,064 (miniboone), the number of features varies from 3 (haberman-survival) to 262 (arrhythmia), and the number of classes varies from 2 (a total of 54 data sets) to 100 (plant-texture, plant-margin, and plant-shape). More information about these data sets can be found in Table 1, 2 of the literature [1]. For the convenience of discussion, each data set is given a unique ID, see Table 2.

(2) Training-test split In order to ensure the reproducibility of the experiments and the fairness of the comparisons, in literature [1], all methods are subjected to the same 4-fold cross validation on every data set. This experiment also used the same training-test split on all the 121 data sets. The

[2]The network structure design comes from `https://github.com/ansh941/MnistSimpleCNN`

[3]See the leaderboard at `https://paperswithcode.com/sota/image-classification-on-mnist`

whole 121 data sets and partitions are available from: `http://persoal.citius.usc.es/manuel.fernandez.delgado/papers/jmlr/data.tar.gz`.

(3) Comparison methods and settings A total of 179 classifiers from these 17 families are used as comparison methods (see Table 3). The names and implementation details of these 179 classifiers can be found in **Section** 2.2 of literature [1].

(4) Settings of NN-FLM In this experiment, the implementation details of NN-FLM are as follows.

1. The feature extraction network (see formula (4)) is a 3-layer full connected network. It can be formally described as follows

$$\mathrm{h}(\mathbf{x};\Theta) = \mathrm{softmax}\left\{\mathrm{LeakyReLU}\left[\mathrm{LeakyReLU}\left(\mathbf{x}\mathbf{W}_1+\mathbf{b}_1\right)\mathbf{W}_2+\mathbf{b}_2\right]\mathbf{W}_3+\mathbf{b}_3\right\},$$

   where $\Theta = \left\{\mathbf{W}_1 \in \mathbb{R}^{d\times 128},\ \mathbf{b}_1 \in \mathbb{R}^{128},\ \mathbf{W}_2 \in \mathbb{R}^{128\times 64},\ \mathbf{b}_2 \in \mathbb{R}^{64},\ \mathbf{W}_3 \in \mathbb{R}^{64\times 32},\ \mathbf{b}_3 \in \mathbb{R}^{32}\right\}$ is the set of the learnable parameters and d is the number of the features of the data set.
2. For the loss (8), $\alpha = 0.2$, $\beta = 0.8$ is set to control the degree of the fuzziness for all the experiments.
3. The number of exemplars of every concept (see **Definition** 4), n_{exe}, is set to 5 for all the experiments. If the number of training samples in the concept is less than 5, then all the training samples are selected as exemplars.
4. For all the experiments, a regularization term $\mathcal{R}(\Theta)$ is added into the loss (8) to control the complexity of the model,

$$\mathcal{R}(\Theta) = \frac{\gamma}{n_{\mathrm{para}}}\sum_{i=1}^{3}\|\mathbf{W}_i\|_F^2 + \|\mathbf{b}_i\|_2^2,$$

   where $n_{\mathrm{para}} = d\times 128 + 128 + 128\times 64 + 64 + 64\times 32 + 32$ is the number of learnable parameters and $\gamma = 0.1$ is the trade-off parameter.
5. For all the experiments, the stochastic gradient descent method Adam [45] optimizer are used to train the model. The size of the batch is set to $\min(4096, n)$, where n is the number of the training samples. We stopped iterating until the loss values of 10 consecutive epochs do not change significantly.

(5) Experimental results The results of the 179 comparison methods on the 121 data set are available from: `http://persoal.citius.usc.es/manuel.fernandez.delgado/papers/jmlr/results.txt`.

The average accuracy of NN-FLM on 4-fold cross validations are recorded in Table 4. Limited by space, it is impractical to list the accuracy of the 179 comparison methods on 121 data sets. Therefore, the top-10 methods with the smallest average rank on the 121 data sets are picked for comparison, and their accuracies are recorded in Table 4.

Table 4: The prediction accuracy (%) of the top-10 methods among 180 methods on 121 data sets, with the rank of the method among the 10 methods (up right).

ID	**NN-FLM**	parRF-t	rf-t	svm-C	rforest-R	svm Poly-t	C5.0-t	svm Radial Cost-t	elm-kernel-m	avNNet-t
1	80.5[1]	74.9	75.3	77.3	74.2	77.4	77.9	80.1	77.5	79.5
2	78.2[1]	73.1	75.5	75.0	71.2	73.1	74.8	74.8	74.6	73.4
3	98.0[1]	96.6	97.0	96.1	97.0	96.9	96.7	95.8	96.1	96.9
4	83.2[1]	80.9	80.8	81.1	80.5	78.3	79.4	79.3	82.7	77.8
5	99.6[1]	98.3	98.6	99.0	97.5	96.1	98.7	99.2	98.1	95.3
6	57.5[1]	55.1	54.0	55.6	52.4	55.9	54.7	55.1	55.9	56.4
7	88.5[1]	87.5	86.8	86.2	88.4	86.7	86.5	85.5	86.6	87.0
8	100.0[1]	80.3	78.9	78.3	81.2	77.5	79.9	78.7	73.6	77.5
9	88.3[1]	84.7	84.0	86.4	83.3	85.5	87.1	82.5	75.0	81.0

Continuation of Table 4

ID	**NN-FLM**	parRF-t	rf-t	svm-C	rforest-R	svm Poly-t	C5.0-t	svm Radial Cost-t	elm-kernel-m	avNNet-t
10	91.0[1]	88.0	88.0	88.0	85.0	88.0	88.0	90.0	84.0	88.0
11	78.3[1]	67.7	69.6	74.0	71.4	74.2	72.9	75.2	72.7	76.1
12	65.7[1]	59.4	59.7	58.6	57.6	59.1	56.1	58.4	58.6	59.7
13	54.2[1]	41.4	35.8	35.5	41.1	44.8	40.0	45.5	42.7	43.1
14	45.0[1]	37.4	32.5	31.5	35.0	36.5	38.0	32.0	31.5	38.1
15	90.2[1]	82.6	83.2	85.9	81.4	82.5	85.7	85.2	84.0	83.9
16	82.8[1]	55.3	54.1	53.6	55.1	55.3	49.3	52.5	70.3	66.7
17	88.5[1]	28.6	28.7	28.9	28.8	28.8	30.8	28.9	21.4	82.2
18	96.0[1]	93.2	92.9	94.9	93.5	92.3	94.6	95.5	94.0	92.6
19	75.3[1]	71.7	72.8	72.4	72.2	73.6	71.3	71.8	74.5	71.7
20	94.5[1]	90.4	89.8	93.4	91.9	90.8	87.0	88.3	90.2	90.6
21	91.9[1]	61.1	61.1	51.9	61.1	55.6	72.2	64.1	66.0	78.2
22	68.5[1]	65.7	65.7	65.3	66.2	66.9	62.5	65.5	62.3	66.4
23	96.7[1]	53.7	53.7	53.2	53.7	67.6	53.9	48.6	62.5	60.4
24	94.2[1]	92.8	92.1	92.3	92.4	92.4	92.0	92.2	94.0	93.0
25	95.8[1]	92.5	92.0	95.1	92.1	93.5	93.0	94.1	94.4	94.2
26	99.0[1]	97.0	97.4	97.2	97.4	98.0	81.6	97.4	98.7	93.7
27	95.3[1]	89.2	90.8	93.4	88.8	91.3	90.8	93.3	94.4	90.3
28	99.7[1]	95.5	95.6	97.6	95.5	97.6	96.1	97.7	97.8	89.9
29	93.3[1]	84.0	84.9	79.8	84.6	86.8	86.8	85.9	88.5	86.8
30	86.0[1]	74.8	71.0	72.1	71.2	69.9	69.8	70.0	65.4	67.0
31	80.5[1]	65.1	70.9	80.4	65.2	75.0	71.8	69.6	71.7	71.7
32	94.0[1]	88.2	86.3	80.0	86.5	87.3	86.3	89.2	88.0	90.3
33	73.9[1]	71.4	72.5	72.8	66.3	71.4	71.4	71.4	72.8	71.4
34	75.0[1]	61.2	68.9	70.5	70.5	71.1	71.1	71.1	72.7	68.9
35	54.3[1]	49.4	48.5	47.3	50.3	46.1	45.8	47.3	52.7	46.1
36	93.1[1]	89.6	90.2	88.6	90.7	89.4	58.8	88.0	90.4	85.6
37	81.6[1]	55.1	55.1	55.1	55.1	55.1	57.8	55.1	58.3	59.9
38	69.5[1]	67.7	67.8	68.2	67.4	67.8	67.5	67.8	67.3	67.8
39	89.9[1]	83.0	83.0	88.1	85.7	85.6	81.1	84.1	86.9	85.6
40	85.9[1]	75.8	76.4	84.7	75.4	85.1	76.2	84.2	84.8	82.6
41	66.9[1]	62.1	62.2	48.0	59.2	56.9	57.7	55.0	47.4	60.3
42	88.3[1]	85.2	84.5	85.4	82.4	86.1	85.2	86.5	86.7	86.1
43	88.0[1]	86.4	85.1	83.8	84.6	85.5	84.2	83.9	82.5	86.8
44	79.8[1]	71.7	68.8	68.3	77.9	69.1	70.0	74.5	70.2	69.8
45	63.2[1]	62.5	62.1	60.8	61.5	63.0	61.4	62.1	61.2	62.3
46	90.4[1]	82.7	84.2	85.6	84.1	87.5	80.8	87.0	90.4[1]	81.7
47	86.6[1]	82.3	83.0	86.0	83.1	83.7	81.3	84.7	82.5	83.3
48	95.8[1]	83.3	87.3	83.3	83.3	88.1	87.5	83.3	91.7	87.5
49	75.0[1]	60.0	62.5	62.5	53.1	56.2	59.0	44.6	37.5	52.4
50	89.2[1]	87.9	85.2	87.2	87.2	85.1	85.1	85.9	87.2	81.8
51	93.7[1]	89.9	88.7	87.8	87.6	92.0	92.4	93.7[1]	87.8	92.2
52	86.8[1]	81.4	81.9	86.0	80.9	85.1	81.0	85.5	86.8[1]	85.3
53	99.7[1]	98.3	98.7	99.5	98.8	99.0	97.8	99.5	99.2	98.7
54	100.0[1]	66.7	87.5	81.2	87.5	81.2	48.3	100.0[1]	93.8	75.0
55	93.4[1]	92.1	92.3	90.2	89.5	90.5	91.7	92.2	90.8	91.4
56	83.9[1]	80.9	82.6	82.5	81.9	83.4	83.6	82.1	81.0	83.7
57	93.3[1]	88.7	90.6	86.5	92.3	85.1	86.9	81.1	86.5	82.9
58	85.7[2]	81.6	81.1	84.1	77.9	82.9	80.2	83.3	86.0[1]	84.4
59	75.0[2]	64.9	66.8	76.0[1]	72.1	58.1	62.8	60.8	64.4	62.1
60	78.5[1]	77.1	77.1	76.8	77.9	76.9	75.9	77.6	78.1	74.6

Continuation of Table 4

ID	**NN-FLM**	parRF-t	rf-t	svm-C	rforest-R	svm Poly-t	C5.0-t	svm Radial Cost-t	elm-kernel-m	avNNet-t
61	100.0[1]	98.9	98.9	98.3	98.9	98.3	98.3	98.9	100.0[1]	98.9
62	99.0[1]	98.1	96.1	98.0	95.0	94.2	95.1	92.3	96.0	93.9
63	98.1[1]	96.8	97.0	97.0	95.8	97.9	96.0	98.1[1]	96.1	97.5
64	100.0[1]	91.7	87.5	50.0	87.5	75.0	79.2	62.5	87.5	87.5
65	98.9[2]	85.3	84.8	99.0[1]	86.5	91.8	84.0	96.6	91.8	93.8
66	89.0[2]	87.2	86.6	86.3	88.1	87.5	83.6	86.3	89.6[1]	86.6
67	88.1[2]	82.5	84.7	88.6[1]	79.7	82.5	80.6	86.1	84.7	73.6
68	92.5[1]	92.0	92.0	92.0	92.0	92.0	92.0	92.0	91.4	71.7
69	97.1[1]	94.3	93.4	95.7	94.7	93.8	92.9	93.4	95.2	96.2
70	88.1[1]	12.0	52.0	64.0	52.0	52.0	8.0	48.0	76.0	60.0
71	76.4[1]	68.4	70.2	71.1	69.7	71.4	70.7	71.4	72.8	72.2
72	93.3[2]	93.8[1]	−−	−−	−−	−−	93.8[1]	−−	−−	92.2
73	99.6[2]	97.7	97.8	99.4	97.8	99.6[2]	98.7	99.5	99.8[1]	98.6
74	90.2[2]	89.9	90.0	90.0	90.5[1]	89.5	90.0	89.8	89.6	89.6
75	97.3[3]	97.8[1]	97.5[2]	95.6	93.9	87.8	96.4	96.6	91.7	90.2
76	97.9[3]	97.8	98.2[1]	96.7	97.9	96.3	98.1[2]	96.5	97.3	95.0
77	98.7[1]	96.7	96.0	96.6	96.7	98.0	95.3	94.0	98.6	97.3
78	66.3[2]	65.1	64.7	66.0	64.9	65.9	64.5	66.5[1]	66.1	67.4
79	99.8[3]	99.1	99.2	100.0[1]	99.0	99.0	98.6	99.6	99.9[2]	98.3
80	90.3[5]	91.0[3]	90.8[4]	91.9[1]	91.0[3]	89.8	89.8	90.8	91.3[2]	86.9
81	66.0[4]	69.0[1]	69.0[1]	64.0	68.9[2]	61.1	68.1[3]	63.4	61.6	61.4
82	86.2[4]	87.4[1]	87.5[2]	−−	85.2	−−	90.4[1]	−−	−−	77.8
83	94.8[5]	95.1[3]	95.0[4]	93.8	95.2[2]	93.5	95.4[1]	93.4	93.3	94.3
84	94.4[5]	94.1	93.7	95.8[4]	94.5	95.9[2]	90.5	95.5[3]	96.4[1]	83.7
85	97.2[1]	97.1	97.1	97.2[1]	97.1	97.1	97.2[1]	97.1	96.8	97.1
86	99.4[3]	99.4[3]	99.5[2]	98.9	98.6	99.5[2]	99.6[1]	99.2	99.5[2]	99.4[3]
87	77.2[5]	79.0[2]	78.6[3]	76.1	78.5[4]	76.2	80.4[1]	75.8	75.9	73.2
88	82.3[6]	86.2[2]	87.2[1]	83.1	85.3[3]	−−	74.9	−−	85.0	32.9
89	73.0[4]	77.4[1]	76.1[2]	72.3	75.2[3]	70.6	71.9	65.5	71.5	63.5
90	74.0[5]	78.0[1]	75.7[3]	68.9	77.8[2]	70.2	74.2[4]	70.6	69.3	66.3
91	81.0[5]	83.4[3]	84.5[2]	84.5[2]	83.1[4]	−−	77.0	−−	85.7[1]	31.9
92	97.8[2]	97.8[2]	97.8[2]	96.7	98.1[1]	97.5	96.7	98.1[1]	96.7	98.1[1]
93	95.3[8]	96.4	96.5	97.4[1]	96.7[3]	96.4	95.7	96.8[2]	−−	64.6
94	85.9[4]	88.1[2]	88.1[2]	84.4	86.6[3]	83.1	88.5[1]	84.3	83.6	79.7
95	78.3[1]	76.6	76.3	78.3[1]	76.4	78.0	76.4	77.2	78.1	77.1
96	97.9[2]	97.2	97.3	97.8	97.5	98.0[1]	96.8	97.8	97.9[2]	97.8
97	87.2[6]	88.0[3]	88.1[2]	87.7	88.3[1]	86.7	87.7	87.3	−−	86.9
98	93.4[4]	94.3[2]	94.1[3]	91.9	94.7[1]	92.6	94.7[1]	93.1	92.7	91.9
99	95.1[4]	94.0	94.0	97.0[2]	96.0[3]	97.0[2]	89.0	96.0[3]	97.0[2]	98.0[1]
100	86.7[3]	85.7	85.6	86.5	84.9	87.1[1]	84.2	86.8[2]	86.6	87.1[1]
101	99.5[4]	99.7[3]	99.8[2]	99.7[3]	98.1	−−	100.0[1]	−−	99.4	93.8
102	98.1[3]	96.6	96.7	98.6[1]	95.8	98.1	97.0	98.6[1]	98.5[2]	90.2
103	62.2[6]	63.9	64.1	72.2[1]	64.9[3]	−−	57.8	−−	71.7[2]	31.1
104	85.1[4]	86.1[2]	86.2[1]	85.0	86.1[2]	84.9	85.6[3]	85.0	−−	85.1
105	86.3[4]	92.9[1]	85.7[3]	85.7[3]	85.7[3]	64.3	89.3[2]	85.7[3]	85.7[3]	42.9
106	86.3[5]	85.7	85.8	86.7[3]	86.9[2]	86.6	84.9	86.6	87.0[1]	86.2
107	98.3[5]	99.4[3]	99.4[3]	100.0[1]	98.7	99.8[2]	93.1	99.8[2]	100.0[1]	82.0
108	97.0[3]	97.5[1]	97.4[2]	96.7	97.4[2]	96.6	97.4[2]	96.4	96.3	96.7
109	98.4[2]	98.9[1]	98.9[1]	96.5	98.9[1]	96.6	93.4	97.1	94.9	98.4[2]
110	87.3[2]	88.2[1]	88.2[1]	86.8	88.2[1]	88.2[1]	76.5	88.2[1]	88.2[1]	85.3
111	60.1[5]	63.0[2]	63.0[2]	60.8	63.7[1]	59.8	61.3	60.5	61.6[3]	59.3

Continuation of Table 4

ID	**NN-FLM**	parRF-t	rf-t	svm-C	rforest-R	svm Poly-t	C5.0-t	svm Radial Cost-t	elm-kernel-m	avNNet-t
112	78.7[3]	79.1[1]	78.9[2]	78.5	77.5	78.3	77.4	78.9[2]	78.0	78.6
113	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]
114	99.8[3]	100.0[1]	100.0[1]	99.9[2]	100.0[1]	99.8[3]	95.5	99.8[3]	−−	99.6
115	59.1[6]	68.6[2]	68.0[3]	64.4	69.1[1]	56.3	66.2	59.0	65.6	55.9
116	60.4[5]	70.1[1]	69.1[2]	46.4	67.7[3]	53.1	65.0	51.0	51.0	52.6
117	63.3[7]	86.4[3]	86.5[2]	82.6	70.5	51.7	88.8[1]	67.9	−−	33.7
118	92.7[5]	99.6[3]	99.7[2]	92.2	99.4	91.6	99.9[1]	92.5	92.3	85.7
119	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]
120	88.6[5]	95.2[3]	95.7[1]	86.8	95.4[2]	85.9	94.9	87.0	85.3	86.4
121	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]	100.0[1]
Mean	**86.5**[1]	82.0[2]	81.6[3]	80.4	81.2	77.1	80.6	77.3	77.2	79.4

−−: The classifier made an error on the corresponding data set (see **Section** 3.1 of literature [1]).

(6) The 12 small sample data sets To verify the performance of NN-FLM on small sample classification problem, we selected 12 data sets with small number of samples. And the basic information of these data sets is shown in Table 5.

Table 2: The ID of 121 data sets

ID	Data set	ID	Data set	ID	Data set
1	blood	41	teaching	81	wine-quality-red
2	breast-cancer	42	vertebral-column-2clases	82	connect-4
3	breast-cancer-wisc	43	vertebral-column-3clases	83	spambase
4	breast-cancer-wisc-prog	44	breast-tissue	84	semeion
5	car	45	congressional-voting	85	ozone
6	contrac	46	conn-bench-sonar-mines-rocks	86	chess-krvkp
7	credit-approval	47	heart-hungarian	87	steel-plates
8	cylinder-bands	48	lenses	88	plant-margin
9	echocardiogram	49	lung-cancer	89	arrhythmia
10	fertility	50	lymphography	90	glass
11	haberman-survival	51	musk-1	91	plant-texture
12	heart-cleveland	52	oocytes-trisopterus-states-2f	92	dermatology
13	heart-switzerland	53	synthetic-control	93	letter
14	heart-va	54	balloons	94	cardiotocography-10clases
15	hepatitis	55	energy-y2	95	pima
16	hill-valley	56	mammographic	96	twonorm
17	image-segmentation	57	molec-biol-promoter	97	magic
18	ionosphere	58	oocytes-merluccius-nucleus-4d	98	cardiotocography-3clases
19	led-display	59	pittsburg-bridges-TYPE	99	annealing
20	low-res-spect	60	statlog-german-credit	100	waveform
21	monks-1	61	wine	101	nursery
22	monks-2	62	zoo	102	ringnorm
23	monks-3	63	breast-cancer-wisc-diag	103	plant-shape
24	oocytes-merluccius-states-2f	64	trains	104	adult
25	oocytes-trisopterus-states-5b	65	balance-scale	105	hayes-roth
26	optical	66	ecoli	106	waveform-noise
27	parkinsons	67	libras	107	conn-bench-vowel-deterding
28	pendigits	68	spectf	108	page-blocks
29	pittsburg-bridges-MATERIAL	69	seeds	109	thyroid
30	pittsburg-bridges-REL-L	70	audiology-std	110	horse-colic
31	pittsburg-bridges-SPAN	71	ilpd-indian-liver	111	yeast
32	pittsburg-bridges-T-OR-D	72	miniboone	112	titanic
33	planning	73	musk-2	113	mushroom
34	post-operative	74	bank	114	statlog-shuttle
35	primary-tumor	75	energy-y1	115	wine-quality-white
36	soybean	76	statlog-image	116	flags
37	spect	77	iris	117	chess-krvk
38	statlog-australian-credit	78	abalone	118	wall-following
39	statlog-heart	79	tic-tac-toe	119	acute-nephritis
40	statlog-vehicle	80	statlog-landsat	120	molec-biol-splice
				121	acute-inflammation

Table 3: The 179 comparison classifiers from 17 families

ID	Family	# Classifiers
F1	Discriminant analysis	20
F2	Bayesian approaches	6
F3	Neural networks	21
F4	Support vector machines	10
F5	Decision trees	14
F6	Rule-based methods	12
F7	Boosting	20
F8	Bagging	24
F9	Stacking	2
F10	Random forests	8
F11	Other ensembles	11
F12	Generalized Linear Models	5
F13	Nearest neighbor methods	5
F14	Partial least squares and principal component regression	6
F15	Logistic and multinomial regression	3
F16	Multivariate adaptive regression splines	2
F17	Other Methods	10
	Sum	**179**

Table 5: The 12 small sample data sets

ID	Data set	# samples	# classes	# training samples per class
S1	trains	10	2	3.75
S2	lenses	24	3	6.00
S3	balloons	16	2	6.00
S4	audiology-std	171	18	7.13
S5	lung-cancer	32	3	8.00
S6	zoo	101	7	10.82
S7	plant-margin	1600	100	12.00
S8	plant-texture	1600	100	12.00
S9	plant-shape	1600	100	12.00
S10	soybean	307	18	12.79
S11	pittsburg-bridges-TYPE	105	6	13.13
S12	breast-tissue	106	6	13.25

training samples per class = $\frac{3}{4} \times$ (# samples $\div$ # classes), because the 4 fold cross validation is used in this experiments.