Spectral Learning with Type-2 Fuzzy Numbers for Question/Answering System

Asli Celikyilmaz[1] I. Burhan Turksen[2]

1. Computer Sciences Division, University in California, Berkeley, CA, USA
2. TOBB Economy and Technology University, Ankara, Turkey & University of Toronto, Canada

Abstract— *Graph-based semi-supervised learning has recently emerged as a promising approach to data-sparse learning problems in natural language processing. They rely on graphs that jointly represent each data point. The problem of how to best formulate the graph representation remains an open research topic. In this paper, we introduce a type-2 fuzzy arithmetic to characterize the edge weights of a formed graph as type-2 fuzzy numbers. The fuzzy numbers are identified by the changing parameters of the fuzzy kernel nearest neighbor algorithm, namely the degree of fuzziness and the hyper-parameter of the Gaussian kernel function, both of which have an effect on the uncertainty in forming the affinity matrix of the graph. We introduce a new graph-based semi-supervised learning with the type-2 arithmetic operations. We apply this technique in the framework of label propagation and evaluate on a question answering task. We demonstrate that the type-2 SSL can improve the prediction accuracy and can be considered to be the an alternative tool for text mining applications of computational linguistics.*

Keywords— Graph-based semi-supervised learning, kernel fuzzy k-nearest neighbor, type-2 fuzzy numbers.

1 Introduction and Motivation

In building reliable models for real systems, identification of exact values of variables of model equations are required. In real life practices, precise values of parameters may not be obtained due to imprecise, noisy, vague, or incomplete nature of information. Fuzzy logic provides explanatory tools for such tasks, mainly because of its capability to manage imprecise categories to represent imperfect information, by means of fuzzy sets, graduality, measures of resemblance or aggregation methods. Type-1 fuzzy sets may not be enough to explain the whole spectrum of possible results, mainly because the values used to characterize the membership functions of type-1 fuzzy numbers are usually overly precise. Usually the level of information is improperly set to define membership functions, thus it is rather necessary to use type-2 fuzzy numbers to represent uncertainties in model parameters.

In this work, we mainly focus on uncertainties in finding similarities in text mining. We consider one of the most commonly used learning methods, namely the semi-supervised learning (SSL) method [1]. It is often the case in the areas of machine learning for classification problems such as the problem of text classification on web pages, automatic translation or online question/ answering systems, etc. that one needs to deal with a very small portion of labeled data and vast amounts of unlabeled data. For such cases, graph-based SSL methods (spectral learning methods) have proved to outperform other learning methods. In graph- based methods the data is represented by the nodes of a graph (Fig. 1), the edges of which are labeled with the pairwise distance of the incident nodes. One problem with spectral learning methods is that the procedure is highly sensitive to the choice of the kernel, for example it is very sensitive to the choice of the spread (variance) of a Gaussian kernel, which naturally effects the similarity matrix defined for the given dataset.

As in the phrase of ”words can mean different things to different people”, an entailment relation between a candidate sentence and a question posed by the user may be evaluated differently by different people. For instance, a different degree of entailment may be assigned by different people for pairs of question ”Who bought Overture?” and candidate sentences such as ”Yahoo bought Overture”, ”Yahoo owns Overture”, ”Overture acquisition by Yahoo”, using linguistic terms such $strict$, $loose$, or $direct$ entailment. Current methods can only use crisp values to define such relations, which cannot be explained to a full extent. Type-2 fuzzy logic is the best fit to define the entailment relations between each sentence. To our knowledge, characterization of edge weights of a graph as type-2 fuzzy numbers, as presented in this paper, is a new approach. The novel type-2 SSL defines such uncertain entailment relation between two sentences by characterizing soft linked graphs.

In this paper we concentrate on characterization of the uncertainties in similarity measure when discovering knowledge from unstructured text using graph-based SSL algorithm. A common way to construct the affinity matrix of a graph is by application of a nearest neighbor method. We use a fuzzy k-nearest neighbor (FKNN) to allow fuzzy decisions based on fuzzy labels. In addition we use its kernel extension [2] to enable solving possible non-linearly separable problems and get non-linear fuzzy boundaries instead of linear boundaries when necessary. In addition kernel methods have proven to prevent over-fitting in high dimensional feature spaces. For these reasons, we consider applying type-2 fuzzy arithmetic to situations where the similarity between two objects, i.e, two sentences, is imprecise.

Thus, the novel type-2 SSL method learns the edge link weights via kernel fuzzy k-nearest neighbor algorithm [2] (KFKNN). We use the arithmetic operations on type-2 fuzzy numbers defined in [3]. For ease of calculations, we graduate the interval valued degree of fuzziness and the kernel parameter and obtain bounded discrete valued weights (interval valued) with associated type-2 membership grades, enabling to represent each weight link with a type-2 fuzzy number. In a way each membership value is further stretched [1] based on fuzziness of the model to capture uncertainty interval of membership values (Fig. 2). Using the interval type-2 fuzzy

[1] Zadeh[4] defines the membership values as elastic constraints that has to be stretched to get their full meaning.

ISBN: 978-989-95079-6-8

weights, we construct fuzzy graph Laplacian. The novel type-2 SSL uses label propagation algorithm [1] to obtain interval valued output values. Defuzzification follows to measure the performance of the new classifier.

In this paper we mainly focus on the application of the novel spectral classification method on a specific area of natural language processing. Text-based question answering (QA) is the process of automatically finding the answers to arbitrary questions in plain English by searching collection of text files. Recently intensive research has been continuing to grow in this area fostered by evaluation-based conferences, one of which is the Text REtrieval Conference (TREC) [5]. Among different question types, most current research focuses on the $factoid$ type questions, e.g., "*How tall is Eiffel Tower*", where the answer is short string indicating a fact with a named entity. We construct a textual entailment model to find the degree of match between the question posed by the user and the candidate answers retrieved by the search engine and then rank the sentences likely to contain the correct answer at the top.

We will briefly review graph-based SSL methods, KFKNN, and type-2 fuzzy numbers. Then we will present the novel type-2 SSL method followed by the benchmark analysis on the TREC dataset. Finally conclusions are drawn.

2 Graph Based Semi-Supervised Learning

We begin with notation and a brief summary of SSL algorithms [1], [6]. Let $X = \{x_1, ..., x_n\}$ represent n data points in $\Re^d$ and $Y = \{y_1, ..., y_n\}$ be their output targets. The labeled part of X is represented with $X_l = \{x_1, ..., x_l\}$ with associated labels $Y_l = \{y_1, ..., y_l\}^T$. For ease of presentation, we concentrate on binary classification, where y_i can take on either of $\{-1, +1\}$. X has also unlabeled part, i.e., $X = X_u \cup X_l$. The aim is to predict labels for the remaining unlabeled points, $X_u = \{x_1, ..., x_u\}$.

In this section we review the most general graph-based semi-supervised learning method for binary classification problem. A graph is denoted with $g = (V, E)$, where $V = X_l \cup X_u$ is set of nodes. E represents the edges connecting two vertexes and is represented by the $n \times n$ symmetric affinity matrix W, where an edge $W(x_i, x_j) = e(i, j)$ represents the similarity between any vertexes x_i and x_j and 0 if there is no similarity between them. The most common similarity measure is the Gaussian kernel of width σ,

$$W_{ij} = K\left(x_i, x_j\right) = e^{-\|x_i - x_j\|^2/2\sigma^2} \tag{1}$$

Earlier research [1] suggests that the choice of σ can strongly influence the results (it is one of the imprecise parameters we use to characterize uncertainty interval of membership functions in our approach.) The diagonal degree matrix D is defined for g by D=$\sum_j W_{ij}$.

In graph-based SSL, a function over the graph is estimated such that it satisfies two conditions: 1) it should be close to the given labels (L), and 2) it should be smooth (S) on the whole graph. These two conditions are presented in a regularization form to minimize the following basic algorithm. Let **f**=$[f_1, ..., f_l, f_{l+1}, ..., f_n]$ denote the predicted labels of X where $f \in \Re^n$. According to the mathematical representation of graph based approach, the objective is to minimize

$$\arg min_f \; E_L(\mathbf{f}) + \lambda E_S(\mathbf{f}). \tag{2}$$

In (2) $E_L(\mathbf{f})$ is a loss function to penalize the deviation from the given labels,

$$E_L(\mathbf{f}) = \sum_{i \subset L} (f_i - y_i)^2 \tag{3}$$

and $E_S(\mathbf{f})$ is a regularizer to represent the label smoothness,

$$E_S(\mathbf{f}) = \frac{1}{2} \sum_{i,j \in L \cup U} W_{ij}(f_i - f_j)^2 = f^T \mathbf{L} f \tag{4}$$

In (4), $L = D - W$ is the graph Laplacian. To satisfy the local and global consistency [1], the normalized combinatorial Laplacian is used such that the $E_S(\mathbf{f})$ is replaced with normalized Laplacian, $\mathcal{L} = I - D^{-1/2} L D^{-1/2}$, as follows:

$$E_S(\mathbf{f}) = \sum_{i,j \in L \cup U} W_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2 = f^T \mathcal{L} f \tag{5}$$

The classification function **f** is learned through any of the $label\ propagation$(LP) algorithms, e.g., [1], [7] on the graph. From consistency approach [1] $f*$ can be found as :

$$f^* = (I - \lambda \left(D^{-1/2} W D^{-1/2}\right)^{-1} Y \tag{6}$$

Nearest neighbors are most commonly used method to identify the affinity matrix of the graph. In this work, we use the KFKNN method by capturing the uncertainty interval of similarities via perturbations on learning parameters.

3 Kernel Fuzzy k-Nearest Neighbor

The k-nearest neighbor approach (KNN) is commonly used in spectral learning to obtain the link weights of the graph Laplacian. It does not depend on the distribution of selected k objects, whereas fuzzy KNN [8] deals with the distribution of the selected k objects to determine the sum of similarities between labeled data points. They both use Euclidean distance function. To handle with more complex real systems, instead of Euclidean distance, we use kernel function to calculate distance between two objects. Through some non-linear mapping the input data is mapped onto a higher-dimensional feature space, i.e., $X \rightarrow \phi\left(X\right)$ to conduct the fuzzy KNN. With the kernel trick, instead of mapping the data point and calculating the distance in the feature space, we can use the kernel function and compute the dot product in some feature space, $K\left(x_i, x_j\right) = \left\langle \phi\left(x_i\right), \phi\left(x_j\right)\right\rangle$. The distances are calculated in this space as follows:

$$\left\|\phi\left(x\right) - \phi\left(x_j\right)\right\|^2 = 2 - 2K\left(x, x_j\right) \tag{7}$$

Gaussian kernel functions as in (1) is used. Let $C_i,\ i = 1, ...c$ represent each class c of distinct labels. The FKNN algorithm [8] assigns memberships as a function of vector distance from their nearest neighbors and those neighbor memberships in the possible classes. The membership values of x_i^u in each labeled class $c = 1, ..., k$ is calculated with

$$\mu_i(x^u) = \frac{\sum_{j=1}^{k} \mu_{ij}\left(1/\left(1 - K(x - x_j)\right)^{1/m-1}\right)}{\sum_{j=1}^{k}\left(1/\left(1 - K(x - x_j)\right)^{1/m-1}\right)} \tag{8}$$

Algorithm 1 Kernel Fuzzy κ-Nearest Neighbor

```
procedure KFKNN($k$, $X^l$, $X^u$, $m, \sigma$)
    for $j \leftarrow 1, n_u, x_j^u \in X^u$ do
        $w_{x_j^u, x_l^l} \leftarrow K(x_j, x_l^l)$ and $\sigma$ using (1).
        Select $k$ nearest neighbors among all labeled objects, $N_j = \left\{x_{1,j}^l, ...x_{k,j}^l\right\}$
        Calculate the membership values of $x_j^u$, $\mu_i(x_j^u)$, in each class $i = 1, ..., c$ using $N_j$ data points via (8)
        If required, classify $x_j^u$ into any of $C_i$ using criteria in (9)
    end for
end procedure
```

The initial class memberships of labeled points are calculated using the function in [2]. Class separation criteria is based on;

$$\hat{y}^u = \arg max_{i=1...c}\ \ \mu_i(x) \tag{9}$$

The input parameters of KFKNN are the k-nearest neighbors, the labeled X^l data points, and unlabeled X^u data points, the degree of fuzziness parameter, m and kernel distance variance parameter σ in (1). The parameter m determines how heavily the distance is weighted when calculating each neighbor contribution to the membership value. It has been recently shown [9], [10], [11] that changing the m parameter in fuzzy clustering methods [12] such as fuzzy c-means and fuzzy c-regression have an effect on the outcome of the partition matrix, as well as the classification results. With some perturbations on the fuzziness parameter, we can identify uncertainty interval of membership values distributions. In this work, we identify the parameter m as an interval valued parameter, $m := [m^l, m^u]$, $l : lower$, $u : upper$. Next we graduate the interval manually into crisp values, $m^r = \left\{m^1, ..., m^{nr}\right\}$. We do the same discretization for the Gaussian kernel hyperparameter, $\sigma^s = \left\{\sigma^1, ..., \sigma^{ns}\right\}$. Each pair of parameter corresponds to one discrete affinity matrix obtained from KFKNN. We obtain the degree of membership values of relations, degree of similarity of an edge, via fuzzy $max - min$ rule, to be explained in section 5.

4 Type-2 Fuzzy Numbers and Operations

Before we present the novel type-2 SSL method, we give a brief review of type-2 fuzzy numbers in the next. Introduced by Zadeh [13] in the trilogy of papers, a type-2 fuzzy set $\tilde{A}$ is characterized in [14] with a type-2 membership function, $\mu_{\tilde{A}}(x, u)$, which maps elements $x \subset X$ to their membership values $u \subset J_x \subseteq [0, 1]$ as follows:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u))\,|\forall x \subset X, \forall u \subset J_x \subseteq [0, 1]\} \tag{10}$$

where J_x is a type-1 fuzzy set, i.e., primary membership value, and $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ is called secondary membership grade. Any type-2 fuzzy set can be represented as a collection of embedded type-2 fuzzy sets [14], as follows:

$$\tilde{A}_e = \sum_{i=1}^{N} [f_{x_i}(u_i)/u_i]/x_i; u_i \subset J_{x_i} \subseteq [0, 1] \tag{11}$$

where u_i is the primary and $f_{x_i}(u_i)$ is the secondary membership grade. The total number of type-2 sets embedded in $\tilde{A}$ is given by $n_e = \prod_{i=1}^{N} M_i$, where M_i is the cardinality of each primary membership grade on $x_i \in X$. When X has a numeric domain, $\tilde{A}$ can be characterized as fuzzy number. Thus for $\tilde{A}_e^j = \left\{\left(u_i^j, f_{x_i}(u_i^j)\right), j = 1..N\right\}$ and $u_i^j \in \{u_{ik}, k = 1, ..., M\}$, we can define a fuzzy number with

$$\tilde{A} = \sum_{j=1}^{n_e} A_e^j \tag{12}$$

Zadeh [13] introduced the union and intersection of type-2 fuzzy sets. Mendel and John [14] then defined the type-2 operations with embedded type-2 fuzzy sets. The union of discrete [1] type-2 fuzzy sets $\tilde{A}$ and $\tilde{B}$ is given as:

$$\tilde{A} \cup \tilde{B} = \sum_{j=1}^{n_A} \tilde{A}_e^j \sum_{i=1}^{n_B} \tilde{B}_e^i = \sum_{j=1}^{n_A} \sum_{i=1}^{n_B} \tilde{A}_e^j \cup \tilde{B}_e^i \tag{13}$$

The union of two type-2 embedded sets in (13) is defined as:

$$\tilde{A}_e^j \cup \tilde{B}_e^i = \bigcup_{k=1}^{N} \left[F_{x_k}\left(u_k^j, w_k^j\right) / \left(u_k^j \vee w_k^i\right)\right] / x_k \tag{14}$$

where $F_{x_k}\left(u_k^j, w_k^j\right)$ is the flag that is computed by the t-norm of the secondary membership grades of any kth embedded set, $\mu_{A_e^j}\left(x_k, f_{x_k}\left(u_k^j\right)\right)$ and $\mu_{B_e^i}\left(x_k, f_{x_k}\left(w_k^i\right)\right)$ [3]. When the secondary membership grades are unity, i.e., interval type-2 fuzzy sets, the union of type-2 embedded sets are defined as:

$$\tilde{A}_e^j \cup \tilde{B}_e^i = \bigcup_{k=1}^{N} \left[1/\left(u_k^j \vee w_k^i\right)\right] / x_k \tag{15}$$

In [15] the arithmetic operations with type-2 fuzzy numbers such as addition of $\widetilde{about}\ 2$ and $\widetilde{about}\ 5$ is defined based on (14) in series of steps. Later in [3], computationally less expensive arithmetic operations, e.g., addition and multiplication with type-2 fuzzy numbers, is defined, which will be used in this paper to define the similarity between the sentences to construct the type-2 affinity matrix. with unity secondary membership grades.

Here we will only give a brief description of addition between two interval type-2 fuzzy numbers,

$$\widetilde{about}\ 3 = \{0.3, 0.6, 0.7\}/2, \{1\}/3, \{0.3, 0.6, 0.7\}/4$$
$$\widetilde{about}\ 5 = \{0.6, 0.7\}/4, \{1\}/5, \{0.6, 0.7\}/6$$

Since the secondary membership grades are unity, we obtain for each discrete primary membership value:

$$\mu_{aboutA_e^j + \widetilde{about}B_e^i}(z) = sup\left(u_x^j \wedge w_y^i\right) \tag{16}$$

where $z = x \otimes y$ is performed for any discrete number defined in $\widetilde{about}\ A$ and $about\ B$, on all embedded numbers $u_x^j = \mu_{about\ A_e^j}$, $j = 1..n_A$ and $w_x^i = \mu_{about\ B_e^i}, i = 1..n_B$. In (16), $\wedge$ is t-norm(minimum). Then the union of all the pairs are captured along domains of $z \in Z = X \otimes Y$. For potential duplicate values from the resultant of the $supremum$ operation in (16), the best practice is to select pair with the highest

[1] In this paper we deal with discrete fuzzy sets as naturally the fuzzy sets are discretized to do inference.

membership value to form the resultant interval type-2 fuzzy number. From the addition operation on two type-2 fuzzy numbers, the resulting type-2 fuzzy number would yield:

$$\tilde{about}\ 8 = \{0.3, 0.6, 0.7\}/6, \{0.6, 0.7\}/7, \{1\}/8, \{0.6, 0.7\}/9, \{0.3, 0.6, 0.7\}/10$$

5 Novel Type-2 Semi-Supervised Learning

5.1 Motivation

In graph based SSL methods, one of the main problems is to construct the graph. Typically, k-nearest neighbor of each data point is identified and then each node is connected to their k nearest neighbors according to the edge weights using a Gaussian distance function as in (1). In cases where it is difficult to identify numerical characterization of similarities between two data points, one could identify imprecise similarities as type-2 fuzzy numbers (as explained in the previous section) and characterize these fuzzy numbers based on small perturbations on the learning parameters. Applications of Type-2 fuzzy modeling tools, e.g., [16], [17], [18] has shown to be effective methods for such cases.

In this work, we mainly focus on the imprecision of model parameters to define the edge weights. It is in this sense that we characterized the edge weights of a graph with type-2 fuzzy numbers, to be explained next.

5.2 The Learning Algorithm

We assume that the dataset $X = \{x_1, ..., x_n\}$ contain set of labeled and unlabeled data points which are represented as $X = \{X^u \cup X^l\}$. There are c number of classes based on the class labels of the labeled data and k represents the nearest neighbor count. For the labeled training data points the degree of memberships to any of classes is either 0 or 1 depending on the label of the class. After we obtain the partition matrix of the overall dataset X, we compute type-2 link membership values of each pair in the dataset.

We construct one KFKNN model and obtain one partition matrix $U^{sr} \subset \Re^{n\times k}$ for each pair of imprecise parameters, $\{\sigma^s, m^r\}$, $s = 1..ns$, $r = 1..nr$, where $\mu_{ij}^{sr} \in U^{sr}$ represents the degree of membership of data point x_j to any class i obtained using parameter set of $\{\sigma^s, m^r\}$. The next step is to merge all these partition matrices to construct type-2 fuzzy graph. Each vertex is connected to its k-nearest neighbor obtained from each KFKNN model. First we measure the similarities between each connected node, $w_{pq}^s \in W^s$ in each graph with Gaussian weighted distance measure as in (1). Note that the edge similarities are effected by the Gaussian variance parameter, σ^s. As a result, we characterize each edge weight, $\tilde{w}_{pq} = \{w_{pq}^1, ..., w_{pq}^{ns}\}$, $s = 1, .., ns$, with a discrete interval valued number based on σ^s.

The next step is to associate degree of similarity to each edge weight. Any vertex v_j, representing each data point x_j holds a degree of membership μ_{ij}^{sr} to each class label $i = 1...c$ (conditional degree of possibility given class labels). To construct the fuzzy graph, μ_i^{sr} we calculate the degree of relationship between each node using conditional possibilities via min-max rule. Let $\mu_{R(x_p,x_q)}$ be the degree of relationship (similarity) between paired data points, x_p and x_q, in the graph which is measured from the conditional possibility values of each vertex. Using the fuzzy $max-min$ rule of relations [13], we calculate the joint degree of fuzzy relations as follows:

$$\mu_{pq}(x_p, x_q) = \vee_{i=1,...,c} \left(\mu_i(x_p) \wedge \mu_i(x_q)\right) \quad (17)$$

where $\vee$ and $\wedge$ denote max and min operations, respectively. Using (17) we define as many degree of similarity values, μ_{pq}^{sr}, as the number of discrete m^r, $r = 1..nr$, to characterize each discrete edge weight, w_{pq}^s. This would identify the edge weights as type-2 fuzzy numbers, see Figure 2.

In summary, we defined ns, $s = 1, ..., ns$ different variance values, σ^s to calculate ns different similarity values, w_{pq}^s, $s = 1.., ns$. for any pair of connected nodes. These form the soft weight, $\tilde{w}_{pq} = w_{pq}^{s=1..ns}$ of the corresponding edge $e_{pq}, p, q = 1, ..., n$ of the graph. In addition, each discrete similarity value w_{pq}^s is represented with a degree of similarity from the $max-min$ operation, $\mu_{pq}^{sr}(x_p, x_q)$ using (17). These stretch out the degree of membership of each w_{pq}^s, and we identify discrete membership values for each weight, w_{pq}^s.

The fuzzy graph where the edges are represented with the soft type-2 weights is shown in Fig. 1, and Fig. 2 displays a sample edge weight that is in the form of type-2 fuzzy number.

The pseudo code for the type-2 fuzzy semi-supervised graph learning method is shown in Algorithm 2. Initially the partition matrix U is computed for unlabeled data points using the labeled data (Step-3). Note that the membership values of the labeled data points $\mu_i(x_j^l)$ to any of the c clusters are either 1 or 0 so they are merged to the partition matrix (Step-7).

After the conditional possibility values of each data point x_j to each class $\mu_i^{sr}(x_j)$ is obtained from every pair of parameter values, $\{\sigma^s, m^r\}$ we measure the joint possibilities of paired data points in step 9-10. We use the $max-min$ rule to obtain single degree of similarity between two nodes, x_p and x_q. In

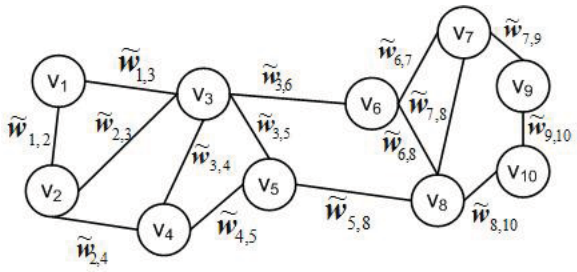

Figure 1: A type-2 fuzzy graph.

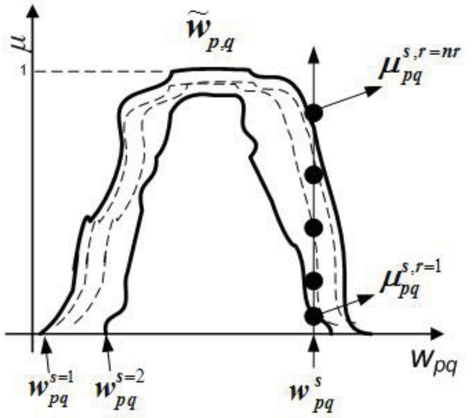

Figure 2: The interval type-2 fuzzy edge weight.

Algorithm 2 Forming Type-2 SSL Graph

```
1: procedure LearnGraph($k$, $X^l$, $X^u$, $X = \{X^l + X^u\}$)
2:     For $\sigma^s = \{\sigma^1, ..., \sigma^{ns}\}$ and $m^r = \{m^1, ..., m^r\}$
3:         -Execute Algorithm 1 using $\{\sigma^s, m^r\}$ to obtain
4:         each partition matrix $U^{sr} \subset \Re^{n \times c}$
5:         -Calculate similarity matrix $W^s = \{w^s_{pq}\}$ using $\sigma^s$
6:     EndFor
7:     $\mu^{sr}_j \in U^{sr} = U^{sr} + \{\mu_{i=1..c}(x^l_{i=1..n_l})\}$ where
8:     For $p, q = 1, ..., n,\ p \neq q$
9:         -Calculate degree of fuzzy relation, $\mu^{sr}_{pq}(x_p, x_q)$
10:        using (17)
11:        -Characterize each edge weight as type-2 fuzzy
12:        number $\tilde{w}_{pq} = \{w^{s=1..ns}_{pq}\} \in \tilde{W} \subset \Re^{n \times n}$
13:        using each $\mu^{sr}_{pq}$.
14:        $\tilde{d}_p \in \tilde{D} \subset \Re^{n \times n} \leftarrow \sum_{q=1}^{n} \tilde{w}_{pq}$
15:    EndFor
16:    $\tilde{Ln} \leftarrow \tilde{D}^{-1/2}\tilde{W}\tilde{D}^{-1/2}$
17: end procedure
```

this way we represent the discrete membership values of the type-2 fuzzy number of the corresponding edge similarity.

Using the type-2 fuzzy number operations from section 4 [3], we obtain the fuzzy degree of each connected edge. Note that, instead of crisp edge weights, W, and graph node degrees, D, we characterized type-2 fuzzy edge matrix $\tilde{W}$ and degrees, $\tilde{D}$. To obtain the soft normalized graph Laplace, we compute the Step-16.

5.3 Inference with Soft Graphs

In the previous section we presented the algorithm to form a soft graph Laplacian matrix $\tilde{L}_n$, the members of which are represented with interval type-2 fuzzy numbers. It has been shown in recent literature [**?**, 9, 11, 10] that some of the uncertainties in system models can be captured during learning when we define interval values for the learning parameters instead of crisp values. For instance when fuzzy clustering methods are used to construct membership functions (fuzzy sets) defining an interval valued fuzziness parameter, m, instead of crisp value, we can identify the uncertainty interval of the membership functions as well as the perturbation effects on the local functions. The same is also true for the parameters of similarity functions. The variance parameter of a Gaussian kernel has a significant effect on the outcome [7]. By the analogy of these approaches, we use the interval valued Gaussian variance distance measures and fuzziness parameter of the KFKNN for identification of the soft graph Laplacian as explained in the previous section. The classifier function is learnt using the graph Laplacian. In standard SSL methods, briefly explained in section 2, one can use a label prorogation method to find the classifier function, f^*. In Type-2 SSL algorithm of this paper, we learn classifier by first type-reducing the type-2 graph Laplacian and then defuzzifying its values to obtain a single graph Laplace.

The graph Laplacian is a sparse matrix $L := (l^{n \times n}_{pq})$ representing the difference between the degree matrix and weight matrix. The matrix elements are zero if there is no edge link, and 1 for the diagonal elements. Hence the matrix is sparse representing the graph where there are edge links. We consider soft normalized graph Laplacian $\tilde{\mathcal{L}} = I - \tilde{D}^{-1/2}\tilde{W}\tilde{D}^{-1/2}$. Each matrix element $\tilde{l}_{pq}$ is an interval type-2 fuzzy number, represented as:

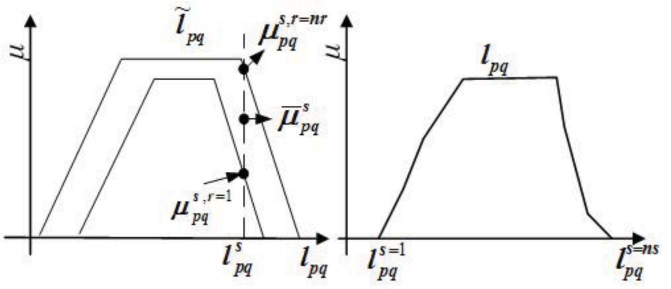

Figure 3: (Left Graph)Interval type-2 fuzzy laplace value. (Right Graph)Type-1 fuzzy laplace value.

$$\tilde{l}_{pq} = \{\mu^{s=1,r=1}_{pq}, .., \mu^{s,r=nr}_{pq}\} / l^{s=1}_{pq} + \cdots + \{\mu^{s=ns,r=1}_{pq}, .., \mu^{s=ns,r=nr}_{pq}\} / l^{s=ns}_{pq}$$

We consider type-reducing the Laplace values via center of gravity (COG) rule as shown in Figure 3 as

$$COG(\mu^s_{pq}) = \sum_{j=1}^{nr} \mu^{s,r=j}_{pq} / nr \tag{18}$$

Then we defuzzify to obtain a single Laplace values:

$$l_{pq} = \frac{\sum_{t=1}^{ns} COG\left(\mu^s_{pq}\right) l^{s=t}_{pq}}{\sum_{i=1}^{ns} COG\left(\mu^s_{pq}\right)} \tag{19}$$

Using the defuzzified graph Laplacian, we obtain the classifier function using (6).

6 Textual Entailment Model for Question Answering

In this section we use a real question and answering (Q/A) dataset to test the performance of the type-2 SSL in comparison to the SSL method [7], and well-known SVM [19]. We used the TREC2001-2003 factoid questions for training and TREC04 questions for testing. The training and testing datasets comprise of 20 relevant sentences for each question that may or may not contain the true answer. We build the training and testing datasets using the relevant documents provided by NIST [2] and extracted 20 relevant sentences from TREC corpus using search engine [3]. We manually classified each retrieved sentence as true/false based on the provided gold-factoid-answers. In addition, we used additional question-sentence pairs from RTE-3, and RTE-4 datasets of RTE challenges [4]. We used only Q/A type sentence pairs and converted the hypothesis sentences into question forms to build the Q/A pairs. We sustained the true/false distributions in the training and testing datasets around 25% true-75% false.

Each of the Q/A pairs, i.e., from TREC and RTE, are used to extract features to indicate true/false entailment. Thus the

[2] available at http://trec.nist.gov/data/qa.html

[3] available at http://lucene.apache.org/java/docs/

[4] Recognizing Textual Entailment Challenge, datasets available at http://www.nist.gov/tac/tracks/2008/rte/index.html

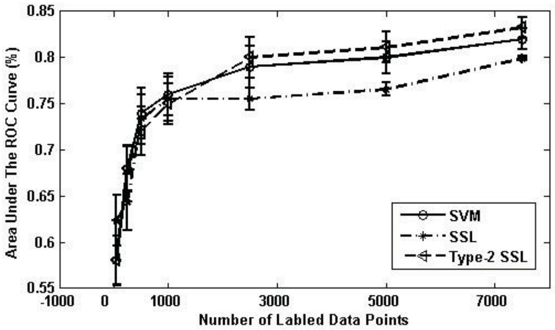

Figure 4: AUC performance on Testing Dataset.

aim of the model is to identify if the candidate sentence entails the question. To measure entailment we extracted 20 different features based on syntactic and semantic match between the question and answer sentences. Some of our features include syntactic matching (keyword and phrase match), named-entity match, headword match using hypernym, hyponym extensions of words, answer type match, which is determined from the question. An automatic hypernym relation extraction algorithm is implemented to capture semantic structures in sentences, such as 'a.k.a', 'such as', apposition relations, to list a few. In addition, we used semantic features such as sentence structure match, i.e., subject, object, headword match, named entity match based on our named entity module to match 50 fine objects in sentence pairs such as time, place, location, person, descriptions, reason, numbers, etc.

In this experiment we used around 600 training questions to compile around 7500 training data points. Similarly, using the entire 202 TREC04 factoid questions as testing, we compiled around 4000 testing data points. Among the training datasets, we used different number of labeled data points to compile series of experiments. Since the data points are randomly selected from the pool, we repeated the experiments 10 times and captured the standard errors to measure robustness. We measured the accuracy of the true/false entailments based on **A**rea **U**nder the Receiver Operating **C**urve statistic. We repeated the experiment using different training sets of different sizes. We built type-2 SSL models with parameters, i.e., $k = \{3, 5, 10, 50\}$, $m = \{1.4, 1.5, ..., 2.6\}$, $\sigma = \{10^{-2}, ..10^{2}\}$ and tested the AUC performance on the same testing dataset. We compared the performance of our models with the ones obtained from application of the same experiments using standard SSL method [20] with the same learning parameters as well as the SVM methods using learning parameters $C = \{2^{-1}, ..., 2^{8}\}$ and $\gamma = \{1 - default\}$. The AUC performance comparison with the well-known SSL and SVM method is shown Figure 4. The results of the experiment on QA datasets reveals that the new soft SSL outperforms the standard SSL method. The results are comparable with the state-of the art SVM tool. Based on the current experiment we can suggest that the presented method is an alternative method for linguistic analysis when there are not enough labeled data but more unlabeled data present.

7 Conclusions

We presented a new uncertainty modeling tool for graph-based semi-supervised learning methods using type2 fuzzy number arithmetic. We learn the interval valued degree of similarity values based on fuzzy k-nearest approach. Each edge linking weight is represented with a type-2 fuzzy number and arithmetic operations on type-2 fuzzy numbers are implemented to build the type-2 fuzzy graph Laplacian and learn a classifier function. Experiments on information extraction from unstructured text for question/answering task have shown promising results.

References

[1] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–328, 2004.

[2] X.-H. Wu and J.-J. Zhou. Kernel-based fuzzy k-nearest neigbor algorithm. 2:159–162, 2005.

[3] W. Blewitt, S.M. Zhou, and S. Coupland. A novel approach to type-2 fuzzy addition. *Proc. Intern. Conference on Fuzzy Systems*, pages 1–6, 2007.

[4] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[5] E.-M. Voorhees. The trec question answering track. *Natural Language Engineering*, 7(4):361–378, 2001.

[6] D. Zhou and B. Scholkopf. Learning from labeled and unlabeled data using random walks. pages 237–244, 2004.

[7] X. Zhu, J. Lafferty, and Z. Ghahramani. *Semi-supervised learning: From Gaussian Fields to Gaussian processes*. Technical Report, Carnegie Mellon University, 2003.

[8] J.M. Keller, M.R. Gray, and J.A. Givens Jr. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Systems Man and Cybernetics*, 15:580–585, 1985.

[9] A. Celikyilmaz and I.B. Turksen. Uncertainty bounds of fuzzy c-regression method. *IEEE-FUZZ Int. Conf. on Fuzzy Systems, Hong Kong*, pages 1098–7584, June 2008.

[10] A. Celikyilmaz and I. B. Turksen. Uncertainty modeling with evolutionary improved fuzzy functions approach. *IEEE Systems, Man, and Cybernetics- Part B*, 38(4):1098–1110, 2008.

[11] C. Hwang and F. C.-H. Rhee. Uncertainty fuzzy clustering: Interval type-2 fuzzy approach to c-means. *IEEE Trans. On Fuzzy Systems*, 15(1):107–120, 2007.

[12] J.C. Bezdek. *Pattern recognition with fuzzy objective functions*. Plenum Press, New York, 1981.

[13] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - (i-ii-iii). *Information Sciences*, 8-9, 1975.

[14] J.M. Mendel and R.I. John. Type-2 fuzzy sets made simple. 10(2):117–127, 2002.

[15] S. Coupland and R. John. An approach to type-2 fuzzy arithmetic. *Proc. UK Workshop on Computational Intelligence*, pages 107–114, 2003.

[16] A. Celikyilmaz and I.B. Turksen. Uncertainty modeling with evolutionary improved fuzzy functions approach. *IEEE Systems, Man, and Cybernetics- Part B*, 38(4):1098–1110, 2008.

[17] O. Castillo and P. Melin. *Type-2 Fuzzy Logic: Theory and Applications*. Springer-Verlag, Heidelberg, Germany, 2008.

[18] M. B. Begian, W. Melek, and J.M. Mendel. Parametric design of stable type-2 tsk fuzzy systems. 2008.

[19] B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proc. of Fifth Annual Workshop on Computational Learning Theory*, 3:144–152, 1992.

[20] X. Zhu and Z. Ghahramani. *Learning from labeled and unlabeled data with label propagation*. Technical Report CMU-CALD-02-107, Carnegie Mellon University, Pittsburgh, 2002.