Classification Methods on Heterogeneous Information Networks

Sagar Marthandan	1140715
Ankit Malhotra	1139941
Akshay Agrawal	1152254

July 17, 2020

Contents

List of Figures

1 Introduction

1.1 Heterogeneous Information Networks

Most real systems consist of a large number of interacting, multi-typed components, while most contemporary researches model them as homogeneous information networks, without distinguishing different types of objects and links in the networks. Recently, more and more researchers begin to consider these interconnected, multi-typed data as heterogeneous information networks, and develop structural analysis approaches by leveraging the rich semantic meaning of structural types of objects and links in the networks. Compared to widely studied homogeneous information network, the heterogeneous information network contains richer structure and semantic information, which provides plenty of opportunities as well as a lot of challenges for data mining. the heterogeneous information network can effectively fuse more information and contain richer semantics in nodes and links, and thus it forms a new development of data mining.

Difference between an Heterogeneous & Homogeneous network networks include different types of nodes or links, while homogeneous networks only have one type of objects and links. Homogeneous networks can be considered as a special case of heterogeneous networks. Moreover, a heterogeneous network can be converted into a homogeneous network through network projection or ignoring object heterogeneity, while it will make significant information loss. Traditional link mining is usually based on the homogeneous network, and many analysis techniques on homogeneous network cannot be directly applied to heterogeneous network.

However for classification in heterogeneous information networks we face several challenges like the complexity of network structure (which increases as the types of objects and links increases), Lack of labels (many times It is difficult to get labels for all types of objects as it is quite expensive in many real world applications) etc. We compare the performance of various algorithms with LLGC (which was discussed previously), wvRN and nLB algorithms. To recall, LLGC is homogenous version of GNetMine where there is single type of objects and links are treated as identical.
In this report, we cover three research papers which provide a survey of heterogeneous information network analysis by the means of classification.

- Learning with Local and Global Consistency.
- Graph Regularized Transductive Classification on Heterogeneous Information networks.
- Rank Based Classification of Heterogeneous Information Networks.

2 Abstract

In **Learning with Local Global Consistency (*LLGC*)**, information is processed from labelled and unlabelled data, design a classifying function, which is sufficiently smooth with respect to the inherent structure collectively shown by the the given labelled and unlabelled data. For this process, an algorithm has been developed, which yields promising results on a mix of classifying problems.

In **Graph Regularized Transductive Classification on Heterogeneous Information networks (*GNetMine*)**, the paper presents an approach for the transductive classification problem on heterogeneous networked data. To give an overview, in transductive classification, all data is observed beforehand i.e. both training and testing data as compared to the inductive learning where we only get to see the training one. Secondly, in transductive learning its a must to have the whole information network trained again for a new data point which is not done in inductive learning. A novel-graph based regularization framework, GnetMine, has been proposed to model the link structure in information networks.

Rank Based Classification of Heterogeneous Information Networks., is a new framework that groups objects into several pre-specified classes, while generating the ranking information for each type of object within each class simultaneously in a heterogeneous information network. According to the current ranking results the graph structure is adjusted to emphasize the sub-network related to a specific class while weakening rest of the network.It generates more accurate classes on networked data and provides meaningful ranking of objects within the class itself, which serves as a more informative view of the data.

Better ranking scores improve the performance of the classifier, by correctly identifying which objects are more important, and should therefore have a higher influence on the classifier's decisions. RankClass essentially integrates ranking and classification, allowing both approaches to mutually enhance each other.

3 Learning with Local and Global Consistency

3.1 Literature Review

Banking on the knowledge of how supervised and unsupervised learning works, we strideforward on how to learn from labelled and unlabelled data. This process is often formalized as unsupervised learning or transductive learning, an example being web categorization classified pages are small but unlabelled pages are large in contrast to the the entire web.

Though this is a hybrid way of clustering data, there are assumptions various testing that has to be performed for this to be a viable method. Some key assumptions for maintaining consistency are

- Nearby points are likely to have the same label.
- Points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same label.

By building an algorithm on the above concepts, various testing is performed on several data-sets and compared with other well known ML classifiers like Support Vector Machine (SVM) and RBF kernel. The above assumptions play a huge role, because of the the way it differentiates between other methods like, spectral, random walks, graph min cuts etc.

3.2 Methodology

For designing a classifier, few assumptions and definition of few entries are to be considered.

- **Point Set** $\mathcal{X} = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\} \subset R^m$ and **Label set** $\mathcal{L} = \{1, \ldots, c\}$,.
- first l points $x_i (i \leq l)$ are labeled as $y_i \in \mathcal{L}$ and remaining points $x_u (l+1 \leq u \leq n)$ are unlabeled.

For complexity purposes, the algorithm mentioned below is explained as much as possible in layman form, reason to keep it abstract. The algorithm runs through a four step process.

1. Form the affinity matrix W **defined by** $W_{ij} = \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$ **if** $i \neq j$ **and** $W_{ii} = 0$
We define a pairwise relationship matrix W on the dataset $\mathcal{X}$ with the diagonal elements being zero. We can think that a graph G = ¡V,E¿ is defined on data set $\mathcal{X}$, where the the vertex set V is just data points from $\mathcal{X}$ and the edges E are weighted by W.

2. Construct the matrix $S = D^{-1/2} W D^{-1/2}$ **in which** D **is a diagonal matrix with its** (i, i) **-element equal to the sum of the** i **-th row of** W**.**
In the second step, the weight matrix W of graph G is normalized symmetrically, which is necessary for the convergence of the following iteration. This method can be considered as the spectral clustering method.

3. Iterate $F(t+1) = \alpha S F(t) + (1-\alpha) Y$ **until convergence, where** α **is a parameter** in$(0, 1)$
During the iteration, the label of each unlabelled point is set to be the class of which it has received the most information. The parameter α specifies the relative amount of the information from its neighbors and its initial label information. The convergence in the iteration process is illustrated in Figure 1.

4. Let F^* **denote the limit of the sequence** $\{F(t)\}$**. Label each point** x_i **as a label** $y_i = \arg\max_{j \leq c} F^*_{ij}$
By proving convergence a final iteration sequence can be achieved (referenced from STEP 3) as

$$F^* = (I - \alpha S)^{-1} Y$$

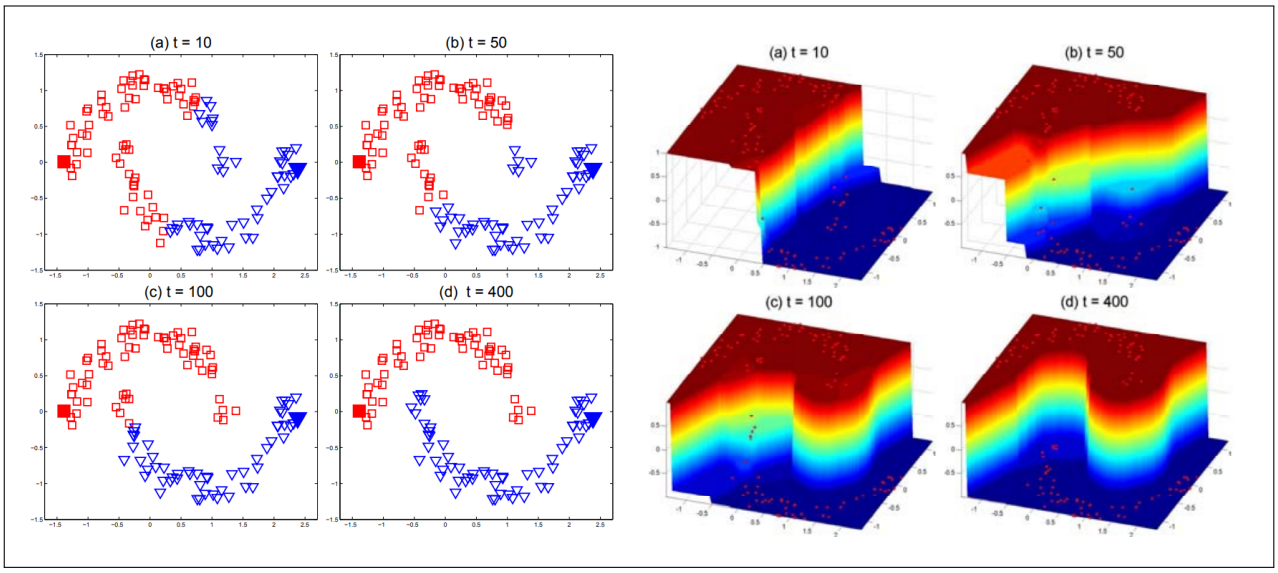

Figure 1: The convergence process of our iteration algorithm with t increasing from 1 to 400 is shown from (a) to (d). Note that the initial label information are diffused along the moons.

This is also used to prove that that the result does not depend in the initial value of the iteration. There exists other two variants for the Iteration Sequence : $F^* = (I - \alpha P)^{-1} Y$. and $F^* = \left(I - \alpha P^T\right)^{-1} Y$. Further a regularization function is developed for the above final iteration to keep the cost error in check. It is formulized as below.

$$\mathcal{Q}(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \right)$$

Where $\mu > 0$ is the regularization parameter and the classification function is as follows.

$$F^* = \arg\min_{F \in \mathcal{F}} \mathcal{Q}(F)$$

1) The first term of the right-hand side in the cost function is the smoothness constraint, which means that a good classifying function should not change too much between nearby points.
2) The second term is the fitting constraint, which means a good classifying function should not change too much from the initial label assignment. The trade-off between these two competing constraints is captured by a positive parameter. The fitting constraint contains labeled as well as unlabeled data.

Smoothness is the sum of local changes of the function between nearby points. Consider these points as undirected weighted graphs, whose weights denote pairwise relationships. The smoothness term essentially splits the function value at each point among the edges attached to it before computing the local changes, and the value assigned to each edge is proportional to its weight.

Differentiating $\mathcal{Q}(F)$ with respect to F, we arrive at the closed form expression of the iteration algorithm i.e.

$$F^* = \beta(I - \alpha S)^{-1} Y$$

4 Graph Regularized Transductive Classification on Heterogeneous Information networks

4.1 Literature Review

The table below represents the various transductive classification methods where one dimension represents whether the data has features/attributes or not, and the other dimension represents different kinds of network structure.

	Non-networked data	Homogenous networked data	Heterogeneous networked data
Attributed data	SVM, Logistic Regression, etc.	Statistical Relational Learning (Relational Dependency Networks, etc.)	
Non-attributed data	/	Network-only Link-based classifier, Relational Neighbor, etc.	***GNetMine***

Figure 2: Summary of related work about transductive classification.

4.2 Methodology

4.2.1 Notations Used

- **Heterogeneous Information Networks :** For given m types of data objects, denoted by $\mathcal{X}_1 = \{x_{11}, \ldots, x_{1n_1}\}, \ldots, \mathcal{X}_m = \{x_{m1}, \ldots, x_{mn_m}\}$, a graph $G = \langle V, E, W \rangle$ is called a heterogeneous information network if $V = \bigcup_{i=1}^{m} \mathcal{X}_i$ and $m \geq 2, E$
- **Class :**A class is defined as $G' = \langle V', E', W' \rangle$, where $V' \subseteq V, E' \subseteq E \ \forall e = \langle x_{ip}, x_{jq} \rangle \in E', W'_{x_{ip}x_{jq}} = W_{x_{ip}x_{jq}}$.
- **Transductive Classification on HIN :**
  1) A subset of data objects $V' \subseteq V = \bigcup_{i=1}^{m} \mathcal{X}_i$, which are labeled with values $\mathcal{Y}$ denoting which class each object belongs to, predict the class labels for all the unlabeled objects $V - V'$.

  2) Suppose the number of classes is K. For any object type $\mathcal{X}_i, i \in \{1, \ldots, m\}$ we try to compute a class indicator matrix $\mathbf{F}_i = \left[\boldsymbol{f}_i^{(1)}, \ldots, \boldsymbol{f}_i^{(K)}\right] \in R^{n_i \times K}$, where each $f_i^{(k)} = \left[f_{i1}^{(k)}, \ldots, f_{in_i}^{(k)}\right]^T$ measures the confidence that each object $x_{ip} \in \mathcal{X}_i$ belongs to class k. Then we can assign the p -th object in type $\mathcal{X}_i$ to class c_{ip} by finding the maximum value in the p -th row of $\mathbf{F}_i : c_{ip} = \arg\max_{1 \leq k \leq K} f_{ip}^{(k)}$

The below mentioned function is optimized through the iterative or minimization process.

$$J\left(\boldsymbol{f}_1^{(k)}, \ldots, \boldsymbol{f}_m^{(k)}\right) = \sum_{i,j=1}^{m} \lambda_{ij} \left(\left(\boldsymbol{f}_i^{(k)}\right)^T \boldsymbol{f}_i^{(k)} + \left(\boldsymbol{f}_j^{(k)}\right)^T \boldsymbol{f}_j^{(k)} - 2\left(\boldsymbol{f}_i^{(k)}\right)^T \mathbf{S}_{ij} \boldsymbol{f}_j^{(k)} \right) + \sum_{i=1}^{m} \alpha_i \left(\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)}\right)^T \left(\boldsymbol{f}_i^{(k)} - \mathbf{y}_i^{(k)}\right) \tag{1}$$

In the above equation,

λ_{ij} and α_i, are the regularization parameters that controls the trade off among different terms, where λ_{ij} and α_i, where $0 \leq \lambda_{ij} < 1, 0 < \alpha_i < 1$. For $\forall i, j \in \{1, \ldots, m\}, \lambda_{ij} > 0$ indicates that object types $\mathcal{X}_i$ and $\mathcal{X}_j$ are linked together and this relationship is taken into consideration and

$$\mathbf{S}_{ij} = \mathbf{D}_{ij}^{(-1/2)} \mathbf{R}_{ij} \mathbf{D}_{ji}^{(-1/2)}, i, j \in \{1, \ldots, m\} \tag{2}$$

where $\mathbf{S}_{ij}$ is the normalized relational matrix of graph $\mathcal{G}_{ij}$ with dimensions $n_i \times n_j$

and where $\mathbf{D}_{ij}$ $(n_i \times n_i.)$ defined for each relation matrix $\mathbf{R}_{ij}$. The (p, p), element of $\mathbf{D}_{ij}$ is the sum of the p -th row of $\mathbf{R}_{ij}$..

5 Rank Based Classification of Heterogeneous Information Networks

5.1 Literature Review

There have been some research and development in the field of Classification and Ranking. Findings from these and the applications of them are on individual basis, and not a combined approach as shown in RankClass. They deal with only homogeneous set of data objects and not heterogeneous, because of which they cannot compute the type differences amongst objects in a heterogeneous information network. Even if applied to heterogeneous objects, these objects are first converted into homogeneous and later the algorithms are applied to them for getting the best of results.
This transformation of the network from homogeneous to heterogeneous can be done in one of the two ways i.e,
1. Treating all objects as the same type by disregarding the type differences between them.
2. Extracting a homogeneous sub-network on a single type of object.

Ranking of objects that are networked have been of great interest in the past few years. In particular two representative algorithms:

1. PageRank
2. HITS

Both of these algorithms are used to calculate the ranking scores of every object in the network using various propagation methods.

For Heterogeneous web, PopRank was introduced to rank popularity. It considers that different types of links in a network have different propagation factors, which are trained according to partial ranks given by experts. On the contrary RankClass ranks objects w.r.t. their importance within each class.

Methods such as boosting, bagging have been seen in multiple studies to improve the quality of classification. One specific boosting method AdaBoost learns from its classification mistakes iteratively, by assigning higher weights to objects that had been misclassified in the previous iteration. The process continues until stable state is reached.
Similar to boosting, RankClass also deals with the importance of the objects in different stages of classification. However, boosting estimates the global ranking/importance of each objects w.r.t. their mistakes, while RankClass makes use of within-class ranking for measuring the importance of each object w.r.t. each class.

5.2 Methodology

RankClass integrates classification and ranking simultaneously in a mutually enhancing process. It is a ranking model which iteratively computes the ranking distribution of objects within each class.
Due to the ranking results of the n-th iteration, the graph structure that is used is adjusted so that the sub-network corresponding to the class is focused while weakening the rest of the network.

A few definitions should be known in order to state the framework for RankClass.
1. Heterogeneous Information Network
2. Class
3. Ranking distribution of objects within each class k, denoted as $P(x|T(x),k)$,
where $k = 1, ..., K$ and $T(x)$ denotes the type of object x

5.2.1 Framework for RankClass

Step 1: Initialize the ranking distribution within each class according to the labeled data, i.e. $(P(x|T(x),k)^0)_{k=1}^K$. Initialize the set of network structures employed in the ranking model, $(G_k^0)_{k=1}^K., asG_k^0 = G, k = 1, ..., K$ as well as initialize t = 1

Step 2: Objects are ranked within their own type and within a specific class. The higher an object x is ranked within class k, the more important x is for class k, and the more likely it is that x will be visited in class k. Using the graph-based ranking model and the current set of network structures $(G_k^t - 1)_{k=1}^K$, update the ranking distribution within each class k, i.e., $(P(x|T(x),k)^t)_{k=1}^K$.

- The initial ranking distribution can be set as a uniform distribution

$$P\left(x_{ip} \mid \mathcal{X}_i, k\right)^0 = \begin{cases} 1/l_{ik} & \text{if } x_{ip} \text{ is labeled to class } k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- To reduce the impact of node popularity Normalized Relation matrix is used in place of traditional relation matrix.

$$\mathbf{S}_{ij} = \mathbf{D}_{ij}^{(-1/2)} \mathbf{R}_{ij} \mathbf{D}_{ji}^{(-1/2)}, i, j \in \{1, \ldots, m\} \tag{4}$$

- The t-th iteration of ranking distribution can be computed by the given equation:

$$\begin{aligned} & P\left(x_{ip} \mid \mathcal{X}_i, k\right)^t \\ & \propto \frac{\sum_{j=1}^m \lambda_{ij} S_{ij,pq} P(x_{jq}|\mathcal{X}_j,k)^{t-1} + \alpha_i P(x_{ip}|\mathcal{X}_i,k)^0}{\sum_{j=1}^m \lambda_{ij} + \alpha_i} \end{aligned} \tag{5}$$

The first term is used to update the ranking score of object x_ip. The relative importance of neighbors of different types is controlled by $\lambda_{ij}\epsilon[0,1]$. The larger the value of λ_{ij} , the more value is placed on the relationship between object types X_i and X_j. The parameters λ_{ij} are used for selecting which types of links are important in the ranking process. The second term learns from the initial ranking distribution encoded in the labels, whose contribution is weighted by $\alpha_i\epsilon[0,1]$.

Step 3: Graph based ranking ranks all object types in the global network. A better way would be ranking within class over the subnetwork corresponding to each class.
Based on $(P(x|T(x),k)^t)_{k=1}^K$, adjust the network structure to favor within-class ranking, i.e., $(G_k^0)_{k=1}^K$
Current ranking distribution $(P(x|T(x),k)^t)$:

$$\begin{aligned} & R_{ij,pq}^t(k) = R_{ij,pq} \times \\ & (r(t) + \sqrt{\frac{P(x_{ip}|\mathcal{X}_i,k)^t}{\max_p P(x_{ip}|\mathcal{X}_i,k)^t} \frac{P(x_{jq}|\mathcal{X}_j,k)^t}{\max_q P(x_{jq}|\mathcal{X}_j,k)^t}}) \end{aligned} \tag{6}$$

Where, $r(t) = \frac{1^t}{2}$
$r(t)$ is a positive parameter that helps the weights of the links to stoop to zero. An alternate equation can be formulated by taking the arithmetic mean instead of the geometric mean, and setting r as a positive function that decreases with t.

Step 4: Repeat steps 1 and 2, setting t = t + 1 until convergence, i.e., until $(P(x|T(x),k)^*)_{k=1}^K = (P(x|T(x),k)^t)_{k=1}^K$ do not change much for all $x\epsilon V$.

Step 5: Based on$(P(x|T(x),k)^*)_{k=1}^K$, calculate the posterior probability for each object, i.e., $(P(k|x,T(x))^*)_{k=1}^K$
Assign the class label to object x as: $C(x) = argmaxP(k|x,T(x))$

6 Experiments and Results

6.1 Learning with Local and Global Consistency

The k-NN and one-vs-rest SVM's are used as baselines and compared to the two other variants of the iteration sequence. The Harmonic Gaussian method is also used, which is relatively close to this method. Since the labeled points are less, there is no reliable approach to select a model, all the algorithms use their respective optimum parameters. The α in this method is simply fixed to 0.99.

Scenario 1 : Toy Problem

The basic idea is to construct a smooth function and use this method to improve a supervised classifier by smoothing its classifying result. The classifying result given by a supervised classifier i used as the input of this algorithm. This conjecture is demonstrated by the toy data-set. Figure 3(a) is the classification result given by the SVM with a RBF kernel. This result is then assigned to Y in our method. The output of our method is shown in Figure 3(b). Note that the points classified incorrectly by the SVM are successfully smoothed by the consistency method.

A function $f(x_i) = (F^*_{i1} - F^*_{i2}) / (F^*_{i1} + F^*_{i2})$ is defined and used as a smoothing function. From the

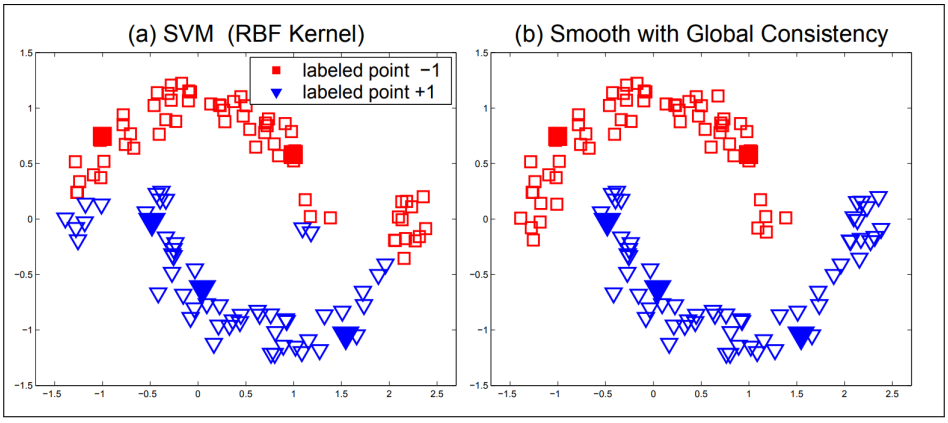

Figure 3: Smooth classification results given by supervised classifiers with the global consistency: (a) the classification result given by the SVM with a RBF kernel; (b) smooth the result of the SVM using the consistency method

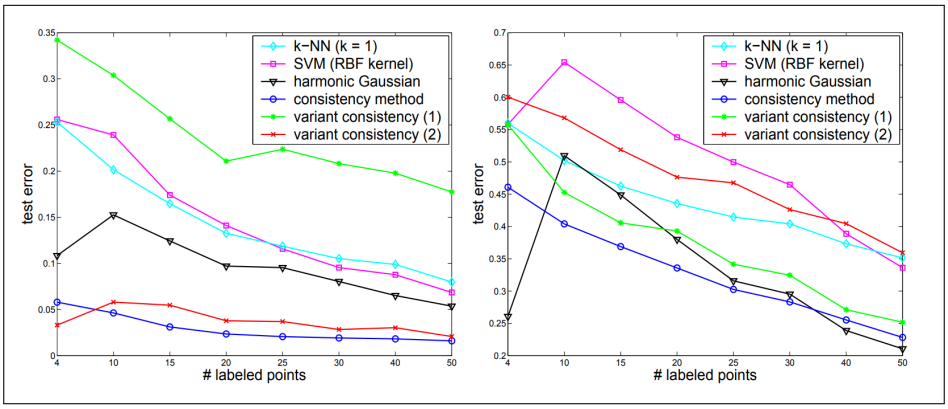

Figure 4: Left panel: Output for Digit Recognition, Right panel: Output for Text Classification. Samples are chosen so that they contain at least one labeled point for each class.

observation in the image provided, its evident on how the the output from the SVM is smoothened.

Scenario 2 : Digit Recognition

Here, classification is performed on USPS handwritten 16x16 digits dataset. Digits 1, 2, 3, 4 are the four classes used. Of 3874 samples, there are 1269 samples of digit 1, 929 for 2, 824 for 3 and 852 for 4.

The k in k-NN is set to 1, RBF kernel width to 5 and for the harmonic gaussian it is set to 1.25. The affinity matrix is constructed by the RBF kernel and diagonal elements is set to 0. Results for over 100 averages is noted below. The consistency method and one of its variants is clearly superiors to the the other methods as seen in Figure 4, Left Panel. A jittering kernel is used in this process, so that it incorporates prior knowledge about digit variance.

Scenario 3 : Text Classification

Classification is perfomed on text using 20-newsgroups dataset. Topics chosen were s autos, motorcycles, baseball, and hockey from the version 20-news-18828. The Rainbow software package processed the artickles with the options
(1) passing all words through the Porter stemmer before counting them.
(2) tossing out any token which is on the stoplist of the SMART system.
(3) skipping any headers and ignoring words that occur in 5 or fewer documents.

The distance between points x_i and x_j was defined to be $d(x_i, x_j) = 1 - \langle x_i, x_j \rangle / \|x_i\| \|x_j\|$. The k in k-NN was set to 1, RBF kernel width for SVM was set to 1.5, and for the harmonic Gaussian method it was set to 0.15. The affinity matrix constructed by the RBF kernel used the same width used as in the harmonic Gaussian method, but the diagonal elements are set to 0. The test errors averaged over 100 trials are summarized in the right panel of Figure 4, Right Panel. Samples were chosen so that they contain at least one labeled point for each class.

The harmonic gaussian method has a starts good with less labelled points, estimates poorly with more labelled points and as it increases it works well again. It works better than the consistency method. The bug advantage that the consistency method has over the Guassian method is the desicion rule is simpler i.e. the naive threshold.

6.2 GNetMine & RankClass method

For both GNetmine and RankClass method, the data set preparation is identical. DBLP data-set which contains 14376 papers, 20 conferences, 14475 authors and 8920 terms with a total number of 170794 links is taken as input. The segregation is done on the data-set into four types of objects namely paper, conference, author and term. Within the network, we also have three types of link relationships namely paper-author, paper-conference and paper-term. The results for both are referenced in the table references from Figure 5, for which the accuracy results are provided.

6.2.1 GNetMine

From Figure , we infer that the performance of wvRN and nLB algorithms are better on the author-author and paper-paper sub networks as compared to working on whole heterogeneous information networks. But when the entire Heterogeneous network is taken into consideration (i.e author-conference-paper-term), the perfomrance of wvRN and nLB reduces, because the complexity of the network increases. GNetMine even outclasses LLGC even though parameters for all types of objects and links are set to the same values.

6.2.2 Rank Class

Note that LLGC, wvRN and nLB are classifiers which work with homogeneous networks, and cannot be directly applied to heterogeneous information networks. So In order to compare all of the above algorithms, the heterogeneous DBLP network can be transformed into a homogeneous network.

Table 3: Comparison of classification accuracy on authors (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (A-A)	nLB (A-C-P-T)	wvRN (A-A)	wvRN (A-C-P-T)	LLGC (A-A)	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	RankClass (A-C-P-T)
(0.1%, 0.1%)	25.4	26.0	40.8	34.1	41.4	61.3	82.9	**85.4**
(0.2%, 0.2%)	28.3	26.0	46.0	41.2	44.7	62.2	83.4	**88.0**
(0.3%, 0.3%)	28.4	27.4	48.6	42.5	48.8	65.7	86.7	**88.5**
(0.4%, 0.4%)	30.7	26.7	46.3	45.6	48.7	66.0	87.2	**88.4**
(0.5%, 0.5%)	29.8	27.3	49.0	51.4	50.6	68.9	87.5	**89.2**
average	28.5	26.7	46.3	43.0	46.8	64.8	85.5	**87.9**

Table 4: Comparison of classification accuracy on papers (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (P-P)	nLB (A-C-P-T)	wvRN (P-P)	wvRN (A-C-P-T)	LLGC (P-P)	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	RankClass (A-C-P-T)
(0.1%, 0.1%)	49.8	31.5	62.0	42.0	67.2	62.7	**79.2**	77.7
(0.2%, 0.2%)	73.1	40.3	71.7	49.7	72.8	65.5	**83.5**	83.0
(0.3%, 0.3%)	77.9	35.4	77.9	54.3	76.8	66.6	83.2	**83.6**
(0.4%, 0.4%)	79.1	38.6	78.1	54.4	77.9	70.5	83.7	**84.7**
(0.5%, 0.5%)	80.7	39.3	77.9	53.5	79.0	73.5	84.1	**84.8**
average	72.1	37.0	73.5	50.8	74.7	67.8	82.7	**82.8**

Table 5: Comparison of classification accuracy on conferences (%)

$(a\%, p\%)$ of authors and papers labeled	nLB (A-C-P-T)	wvRN (A-C-P-T)	LLGC (A-C-P-T)	GNetMine (A-C-P-T)	RankClass (A-C-P-T)
(0.1%, 0.1%)	25.5	43.5	79.0	81.0	**85.0**
(0.2%, 0.2%)	22.5	56.0	83.5	85.0	**85.5**
(0.3%, 0.3%)	25.0	59.0	87.0	87.0	**90.0**
(0.4%, 0.4%)	25.0	57.0	86.5	89.5	**92.0**
(0.5%, 0.5%)	25.0	68.0	90.0	94.0	**95.0**
average	24.6	56.7	85.2	87.3	**89.5**

Figure 5: Comparison of classification accuracy on authors in percentage

While classifying authors and papers, homogeneous A-A and P-P sub-network were also constructed in various ways. Here for nLB and weRN, for authors the best result is given by co-author networks and for the papers it is given by linking two papers if they're published in the same conference. The table proves that the homogeneous classifiers are more suitable for working with homogeneous data and transforming them into a heterogeneous sub network results in data loss.
Compared to GNetMine, RankClass achieves 16.6%, 0.58% and 17.3% relative error reduction in the average classification accuracy when classifying authors, papers and conferences, respectively. The table also proves that the RankCLass Algorithm outperforms all others.

Convergence Study

This experiment shows the relative changes in the weight links within the network. All the links that are connected to object type X_i in the network can be denoted by two categories: first one contains the links that connect to atleast one object of class k which is denoted as G_{in}, while the second category contains the remaining objects which are not of class k (denoted as G_{out}.

The experiment results show that initially the average link weight of G_{in} and G_{out} were same and decrease in later iterations. After a few iterations, the average link weights in Gin and Gout converge to relatively stable values.

The exponential drop of many weights is due to the multiplication of $r(t)$. The experimental graph shows a clear gap between the average link weights of Gin and Gout, making it easier to relate that the sub-network corresponding to each class k is well-separated from the rest of the network so that the ranking can be

accurately performed withing the subnetwork rather than the global network.

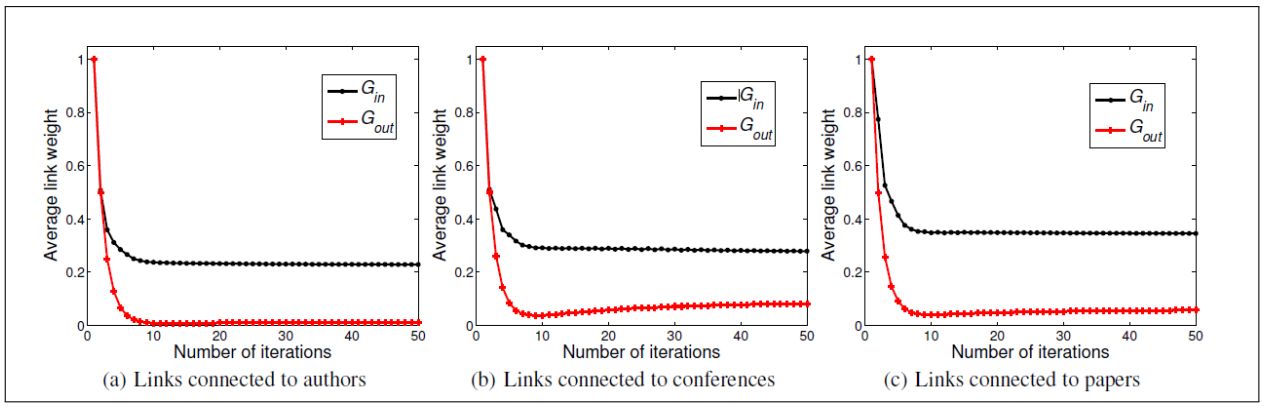

Figure 6: Link weight change in 50 iterations

Time Complexity Study

The running time of the algorithm is computed by varying the size of the algorithm by randomly selecting connected sub-networks from the original network. The graph shows that the time complexity of the method used is generally linear w.r.t. the size of the database, which is consistent with the analysis done.

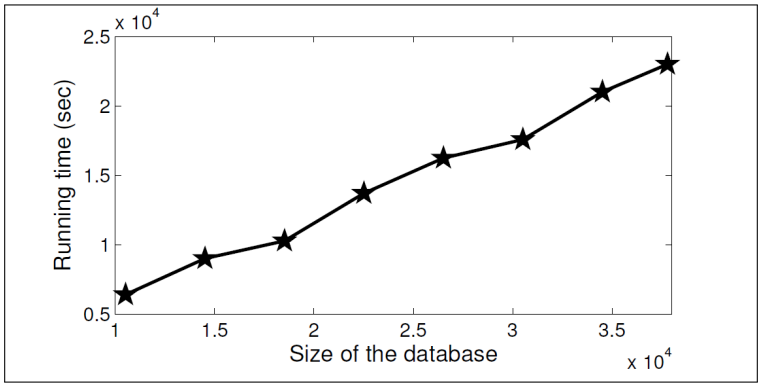

Figure 7: Running time w.r.t. database size

7 Conclusion

Though **LLGC** compares well when pitted against nLB & wvRB, it falls short against methods like GNetMine & Rank Class methods.

The application of **GNetMine** algorithm on heterogeneous information network shows that different types of objects and links should be treated separately due to different semantic meanings. This preserves consistency over each relation graph corresponding to each type of links separately and minimize the aggregated error.

The presented framework randomly classifies the unlabeled data by labeling some randomly selected objects. However, the quality of labels can significantly influence the classification results.

Through through analysis of **Rank Class** method, it is evident of this method is the best of the three. Considering the experiments and analysis, the above statement holds to this committed report.

References

[1] Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Scholkopf. *Learning with local and global consistency.* . In: NIPS 16 (2003).

[2] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. *Graph Regularized Transductive Classification on Heterogeneous Information Networks.* . Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL.

[3] Ming Ji, Marina Danilevsky and Jiawei Han. *Ranking-Based Classification of Heterogeneous Information Networks.* . KDD '11: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining