The Relational Vector-space Model and Industry Classification

Abraham Bernstein
University of Zurich
Winterthurerstrasse 190
8057 Zurich, Switzerland
bernstein@ifi.unizh.ch

Scott Clearwater
Clearwater Ways
P.O. Box 620513
Woodside, CA 94062, U. S. A.
clearway@ix.netcom.com

Foster Provost
NYU Stern School of Business
44 West 4th Street, Room 8-86
New York, NY 10012, U.S.A.
fprovost@stern.nyu.edu

ABSTRACT

This paper addresses the classification of linked entities. We introduce a relational vector-space (VS) model (in analogy to the VS model used in information retrieval) that abstracts the linked structure, representing entities by vectors of weights. Given labeled data as background knowledge/training data, classification procedures can be defined for this model, including a straightforward, "direct" model using weighted adjacency vectors. Using a large set of tasks from the domain of company affiliation identification, we demonstrate that such classification procedures can be effective. We then examine the method in more detail, showing that as expected the classification performance correlates with the relational autocorrelation of the data set. We then turn the tables and use the relational VS scores as a way to analyze/visualize the relational autocorrelation present in a complex linked structure. The main contribution of the paper is to introduce the relational VS model as a potentially useful addition to the toolkit for relational data mining. It could provide useful constructed features for domains with low to moderate relational autocorrelation; it may be effective by itself for domains with high levels of relational autocorrelation, and it provides a useful abstraction for analyzing the properties of linked data.

Keywords

relational data mining, vector-space models, industry classification, homophily, relational autocorrelation, relational-neighbor classifier

1. INTRODUCTION

The analysis of linked data differs from the traditional data-mining scenario: the data items, instead of being statistically independent, have relationships to each other. Linked data are ubiquitous, and relational data mining is receiving increasing attention with the explicit linking of web sites, and with the need to analyze social networks for applications such as counterterrorism [1, 2, 3]. We address a particular relational data mining application: identifying the group membership of linked entities. We address company-industry affiliation, but the framework and methods we describe are intended to be general.

Figure 1 shows a link diagram of companies and their relationships, as extracted from the business news. Colors indicate industry-sector affiliation. The diagram suggests that relationships may play a useful role in identifying the (unknown) affiliation of a company, because linked companies often have the same affiliation.

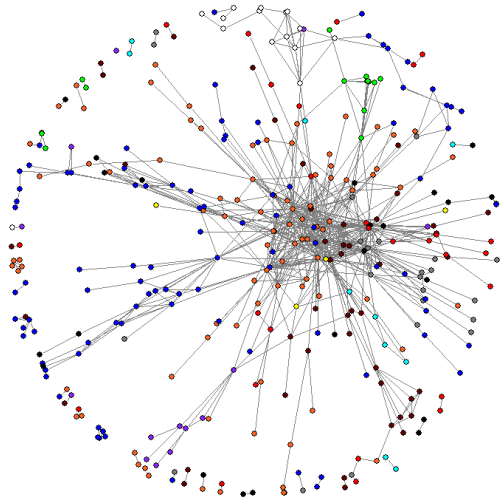

Figure 1. Link diagram of firms. Only links with strength > 4 are shown (but proximity also indicates relatedness). Colors indicate industry-sector membership.

The key contribution of this paper is the presentation and demonstration of a simple, but useful, method for producing classification models from linked data. In analogy to information retrieval [4], we represent entities using a vector-space model. The *relational vector-space* (RVS) model abstracts away much of the graph structure, representing entities by adjacency vectors. Various classification procedures can be defined on the RVS model.

The main attraction of the RVS model is its simplicity. We argue that RVS class-membership scores could be useful constructed features for more complex (relational) data-mining approaches, such as ILP [5] that do not naturally summarize the class membership of local neighborhoods. We also believe that for certain tasks, the RVS model may be appropriate by itself.

The rest of the paper is organized as follows. We present the RVS model formally, and use it to define several classification scoring functions. Next we introduce the domain of company affiliation identification, from which we will take a set of classification tasks. Then we present the results of an experimental case study, examining the effectiveness of the RVS model for classification in this domain. Finally, we show how the model's scores can be used to analyze and visualize certain class-related information about the original, complex graph.

2. THE RVS MODEL

We make a direct analogy to the "vector-space model" used for information retrieval, in which all textual and linguistic structure is ignored and documents are represented by vectors of weights on words. The relational vector-space model is a similarly limited abstraction of the graph structure, into a representation on which straightforward classification techniques can be built. Specifically, each dimension in the vector space corresponds to another entity; each entity is represented by a (weighted) adjacency vector (i.e., the magnitude along each dimension is some measure of the strength of the relationship).

2.1 General Model

Formally, we consider a set of entities E and a set $B \subseteq E$ of "background knowledge" entities. Later in our company affiliation domain, the entities will be companies and the background knowledge will be companies for which the classification is known. We place an (arbitrary) ordering on B, resulting in b_i, i = 1, .. |B|. These define the dimensions of the vector space, and thereby the dimensions along which any entity can be described.

Definition: An entity e is described by an *entity vector* $\mathbf{w} = (w_1, w_{2,} \ldots)$, where w_i is the strength of the relationship between entity e and background entity b_i. Ignoring strengths gives a *simple entity vector*, **w,** where the w_i are binary (presence/absence of a link).

This relational vector-space representation can be used for classification and clustering of entities, and other tasks that rely on entity similarity. In this paper, we will consider entity classification. In particular, consider a discrete, finite set of classes **C**, such that for each $C_i \in \mathbf{C}$, $C_i \subseteq E$. If $e \in C_i$, e is considered to be a member of class i. In principle, the classes need not be mutually exclusive, but we will consider them to be for this paper, so the class can be considered to be a single-valued attribute of an entity and (later) we can adapt previous notions of relational autocorrelation directly. By definition, for $e \in B$, class membership is known. We would like to determine (estimate) class membership for at least one entity $e \notin B$.

Definition: Each class $C_i \in \mathbf{C}$ is described by a *class vector* $\mathbf{c}_i = (c_{i,1}, c_{i,2,} \ldots)$, where $c_{i,j}$ is the strength of the relationship between class C_i and background entity b_i.

In order to classify an entity, we will consider how similar the entity vector is to each class vector, using a similarity-based scoring function. First, let us define a generalized scoring function.

Definition: The *generalized RVS score* of entity e for class i is the normalized inner product of **w** and $\mathbf{c_i}$ (the normalizing function $g(\mathbf{w}, \mathbf{c}_i)$ is discussed below):

$$d(e,i) = \frac{\mathbf{w} \cdot \mathbf{c}_i}{g(\mathbf{w}, \mathbf{c}_i)}$$

RVS scores may be used for classification and other class-based scoring (e.g., for ranking) directly. They also could provide generally useful constructed features to be used by other methods (for example, more complex relational data mining methods [1,2,3]).

2.2 Instantiating the RVS Model

To define specific RVS scores we must answer three questions, which we now will address in turn.

1. How exactly are the entity vectors, **w**, defined?
2. How exactly are the class vectors, $\mathbf{c_i}$, defined?
3. What normalizing function, $g(\mathbf{w}, \mathbf{c}_i)$ is used?

Entity vectors. Recall that an entity vector is composed of the strengths of the relationships between the entity e and the background entities b_i. Of course, the definition of strength is domain dependent, but there are some general issues worth highlighting. In all cases, we will consider $w_i = 0$ to indicate the lack of a relationship between e and b_i. A simple way of defining entity vectors is to ignore strengths, creating a vector of binary indicators. If there is a natural notion of strength, such as the number of links between entities, this gives an obvious way of defining the w_i. However, in analogy to how the vector-space model is used in text classification, a TFIDF-like weighting scheme [4] may be provide added discrimination power.

Class vectors. Defining class vectors is somewhat more involved. One general *direct* method is to give non-zero weights to the background entities that are members of the class. The distribution of weights places an *a priori* directionality on the class vector, which ideally maximizes discriminability. Using uniform weights defines a set of simple, "canonical" vectors for each class.

Definition: The *canonical class vector*, $\mathbf{c_i}$, for class i has non-zero components:

$$c_{i,j} = 1 \Leftrightarrow b_j \in C_i$$

Other distributions of direct weights may be natural for a particular domain, based on background knowledge or statistics summarized from the corpus of background entities. For company affiliation classification, companies in an industry (class) may be weighted by

market capitalization or by a measure of marginal probability of linkage to same-class companies.

These direct methods assume that linkage to members of the same class is sufficient for discrimination. It may be that members of the same class are not linked to each other, but are linked to the same other entities (or other classes). Short of abandoning the RVS approach for a more complex graph-based approach, an *indirect* method for defining class vectors may be beneficial.

Definition: The *simple indirect class vector*, $\mathbf{sic_i}$, for class i is the vector sum of the entity vectors for the background entities belonging to the class:

$$\mathbf{sic_i} = \sum_{e \in C_i \cap B} \mathbf{w}$$

One can define more complicated indirect class vectors. For example, a class centroid would be slightly more complicated. An even more complicated indirect method would be to redefine the b_i, one per class, as "super-entities." Then an indirect method could compare an entity's distribution of links to the various super-entities to the average distributions for those classes. For this paper, we do not consider complicated variations further.

Normalization functions. Generally, $g(\mathbf{w}, \mathbf{c}_i)$ defines the semantics of the similarity represented by the score. For example, the familiar "cosine similarity" between the entity vector and the class vector is d(e,i) with the following normalization function:

$$g(\mathbf{w}, \mathbf{c}_i) = \|\mathbf{w}\| \left\|\mathbf{c}_i\right\|,$$

where $\| \ \|$ is the Euclidean (L2) norm. Whether the exact cosine distance, or some other normalization, is appropriate is domain dependent, but also depends on the definitions of $\mathbf{w}$ and $\mathbf{c_i}$. For the experiments below, we will look at several scoring functions representing different similarities. These scoring functions are defined by different instantiations of $\mathbf{w}$, $\mathbf{c_i}$, and $g(\mathbf{w}, \mathbf{c}_i)$.

2.3 Five RVS scoring functions

The RVS model gives a convenient design space of classification scoring functions. We concentrate on the canonical class vector, because it is easy to define, and creates intuitively attractive scores (that perform well in our domain).

Definition: The *class-normalized direct RVS score* of entity e for class i is the inner product of $\hat{\mathbf{w}}$ and the canonical class vector $\mathbf{c_i}$, normalized by the L1 norm of $\mathbf{c}$.

$$s_{cnd}(e,i) = \frac{\hat{\mathbf{w}} \cdot \mathbf{c}_i}{\sum c_{i,j}}$$

The class-normalized direct RVS score counts up the connected entities belonging to the class, and then normalizes by the size of the class,[1] so that certain classes do not get higher scores simply because they are larger.

Definition: The *entity-normalized direct RVS score* of entity e for class i is the inner product of $\hat{\mathbf{w}}$ and the canonical class vector $\mathbf{c_i}$, normalized by the L1 norm of $\hat{\mathbf{w}}$.

$$s_{end}(e,i) = \frac{\hat{\mathbf{w}} \cdot \mathbf{c}_i}{\sum \hat{\mathrm{w}}_j}$$

The entity-normalized direct RVS score is attractive intuitively: it represents the proportion of connected entities that are members of C_i. This normalizes so that certain entities do not get higher scores simply by being more highly connected.

Definition: The *weighted, entity-normalized direct (wend) RVS score* of entity e for class i is the inner product of $\mathbf{w}$ and the canonical class vector $\mathbf{c_i}$, normalized by the L1 norm of $\mathbf{w}$.

$$s_{wend}(e,i) = \frac{\mathbf{w} \cdot \mathbf{c}_i}{\sum w_j}$$

Using a weighted entity vector inherently deals with noise (spurious, low-weight links) in the data. Using the L1 norm of the weight vector gives the intuitively appealing weighted proportion of links that are to members of the class of interest.

All three of these methods directly relate the entity vectors $\mathbf{w}$ with the respective canonical class vectors $\mathbf{c_i}$. A second group of scoring functions relates the entity vector $\mathbf{w}$ with the simple indirect class vector $\mathbf{sic_i}$ of a class.

Definition: The *(simple) indirect RVS score* of entity e for class i is the cosine similarity between $\mathbf{w}$ and $\mathbf{sic_i}$,

$$d(e,i) = \frac{\mathbf{w} \cdot \mathbf{sic}_i}{\|\mathbf{w}\| \|\mathbf{sic}_i\|}$$

We define *efigf* weights (entity frequency inverse graph frequency) analogously to the TFIDF (text frequency inverse document frequency) weights used in Information Retrieval [4].

Definition: The *efigf-based indirect RVS score* of entity e for class i is the cosine between the efigf-normalized vector $\mathbf{w'}$ and the analogously normalized vector $\mathbf{sic_i'}$, where

$$\mathbf{ef} = w \frac{1}{\max_l(w_l)}, \ igf_i = \log\left(\frac{N}{n_i}\right), \text{ and}$$

$$\mathbf{w'} = \mathbf{ef} \ ? \ \mathbf{igf} \ (\mathrm{sic_i'} \text{ analogously})$$

$$\text{hence, } d_{efigf}(e,i) = \frac{\mathbf{w'} \cdot \mathbf{sic}_i'}{\|\mathbf{w'}\| \|\mathbf{sic}_i'\|}$$

[1] For the canonical class vector, the semantics of the cosine of the angle between it and a weighted entity vector is dubious.

3. DOMAIN & TASKS

To demonstrate the RVS model, we report a case study involving several classification tasks from the domain of company affiliation identification. Identifying the group membership of companies is a prerequisite for solving various problems. Consider industry membership. Determining which companies belong to a particular industry is essential for intellectual property (e.g., patent) litigation, financial analysis (e.g., balancing a portfolio, constructing sector funds), making/improving government economic projections, and so on.

Traditionally, industry membership has been determined by a manual process, and there are various existing classifications. For example, the US Government's Office of Management and Budget has developed a framework for how to assign SIC codes ("Standard Industry Classification" codes–hierarchical, four digit codes used as industry identifiers for firms). Business information companies, such as Hoover's and Yahoo, have different industry classifications (which often do not have a high degree of correspondence with the assigned SIC codes). There are known problems with industry classifications. For example, one study showed that two common SIC-code sources for the same companies disagreed on more than 36% of the codes at the 2-digit code level, and on more then 80% at the 4-digit level [6].

The RVS model can take as background knowledge any industry classification, and (attempt to) classify companies based on it. This gives the additional flexibility to adjust the classification of some background companies, and have the model adjust the rest accordingly, or start from scratch with a new scheme.

The quality of the generalization performance is an empirical question, which we address next for Yahoo's classification. Thus, for the RVS model, E is the set of companies, **C** comprises the Yahoo classifications (industry sector, unless otherwise noted), and B contains the companies for which the Yahoo classification is (deemed to be) known. We chose Yahoo because the granularity of the classifications (12 sectors) was attractive for a conference-paper study and because of ease of access to the data.

For the RVS model we also need a source for links between companies. For this study we chose a generic, but easily accessible link: two companies are linked if they cooccur in a business news story, with the strength of the relationship being the number of such links. Note that cooccurrence lumps together a wide variety of relationships, including joint ventures, mergers/acquisitions, product-related, market related, and so on. Some have nothing to do with industry membership (e.g., two companies happen to announce earnings on the same day). We based the cooccurrences on a collection of news stories from the period December 1999 to September 2002, for which the news provider had assigned at least two ticker symbols and for which the symbols appeared in the Yahoo classification.

4. RESULTS

To compare the various RVS scoring methods, we take each affiliation (the 12 Yahoo sectors) and ask how well the companies can be separated into those belonging to the affiliation and those not. We examine the five scoring functions listed in Section 2.2. and two extensions (described later). We also examined the methods using as the affiliations 97 Yahoo industries, with similar results (which we also use for illustration).

4.1 ROC Analysis for Sectors

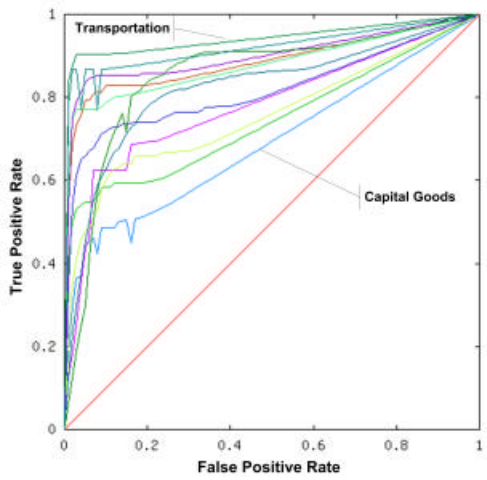

Figure 2: ROC curve for weighted, entity-normalized method (averaged over 10 runs)

We use ROC analysis [7, 8] to assess the model's ability to separate class members from non-members. For a given scoring of companies, ROC curves plot all the possible tradeoffs between correctly classifying the members of the class (the true positive rate, on the y-axis) and incorrectly identifying non-members of the class (false-positive rate, on the x-axis). The area under the ROC curve (AUC), equivalent to the Wilcoxon-Mann-Whitney statistic, is the probability that a member of the class will be scored higher than a non-member [9]. Error is calculated as 1 – AUC, and since the AUCs often are close to 1, relative error reduction[2] is reported for comparisons.

Figure 2 shows the ROC curves for the best method, the weighted, entity-normalized direct score (s_{wend}). Generalization performance ranges from moderate class separability (AUC=0.68 for Capital Goods) to excellent class separability (0.93 for Transportation). Referring back to Figure 1, Transportation is green, and we can see

[2] Relative error reduction of method2 over method1 = (AUC2 – AUC1)/(1-AUC1).

that green nodes are very well interlinked. (Capital Goods, cyan, are interlinked not nearly as well.)

Table 1 reports the AUCs of all 5 scoring functions for the 12 classification tasks. In most cases all the scoring methods classify considerably better than random (represented by the diagonal in ROC space). s_{wend} consistently performs better than the other scores (with only a few exceptions).. Table 2 shows the relative error reduction of s_{wend} over the other methods. s_{wend} has lower error than its closest competitor, the simple s_{end}, on 10 of 12 classification tasks, but achieves only a 2.3% error reduction on average.

	area under curve				
Sector	s_{end}	s_{cnd}	s_{wend}	d_{si}	d_{efigf}
BasicMaterials	0.7318	0.6644	0.7339	0.6218	0.6494
CapitalGoods	0.6781	0.6635	0.6810	0.5274	0.5476
Conglomerates	0.7563	0.5318	0.7697	0.6236	0.6281
ConsumerCyclical	0.7379	0.6087	0.7463	0.5845	0.6073
ConsumerNonCyclical	0.8704	0.6530	0.8753	0.7227	0.7285
Energy	0.8685	0.7701	0.8682	0.8083	0.8520
Financial	0.8002	0.6619	0.8067	0.5566	0.6238
Healthcare	0.8890	0.6918	0.8898	0.7652	0.8142
Services	0.7966	0.6035	0.8124	0.5823	0.6031
Technology	0.8378	0.6785	0.8427	0.7146	0.7294
Transportation	0.9306	0.7325	0.9307	0.8406	0.8825
Utilities	0.9103	0.7982	0.9096	0.8841	0.8924
Average	***0.8173***	***0.6715***	***0.8222***	***0.6860***	***0.7132***

Table 1: Area under curve (AUC) for all scoring methods

	error reduction			
Sector	s_{end}	s_{cnd}	d_{si}	d_{efigf}
BasicMaterials	0.0080	0.2072	0.2966	0.2411
CapitalGoods	0.0090	0.0520	0.3250	0.2948
Conglomerates	0.0550	0.5081	0.3881	0.3807
ConsumerCyclical	0.0322	0.3517	0.3895	0.3540
ConsumerNonCyclical	0.0382	0.6407	0.5503	0.5408
Energy	-0.0028	0.4267	0.3122	0.1092
Financial	0.0327	0.4283	0.5642	0.4863
Healthcare	0.0068	0.6423	0.5305	0.4066
Services	0.0778	0.5268	0.5508	0.5274
Technology	0.0303	0.5106	0.4489	0.4186
Transportation	0.0007	0.7409	0.5653	0.4101
Utilities	-0.0073	0.5520	0.2201	0.1600
Average	***0.0234***	***0.4656***	***0.4285***	***0.3608***

Table 2: Relative error reductions for s_{wend} over other methods

Notice the curious shape of the ROC curves in Figure 2: rather than having smoothly decreasing slopes (for ROC curves the slope corresponds to the class-membership likelihood ratio), after a certain point the slope is constant (to (1,1)). This is an indication that s_{wend} is giving equal (low) scores to a large number of entities. Examining the scores we see that, indeed, the direct method is giving scores of zero to many entities.[3]

s_{wend}=0 means that the entity is not linked to any (background) members of the class. This may largely be due to our limited data sample. A larger sample would contain (i) many more links and perhaps (ii) many more labeled background companies. Moreover, comparing different direct scores on these data obscures their differences, because (as is evident in Figure 2) due to the large number of zeros, for a given industry the AUCs cannot be very different for different direct scorings (which would correspond only to different slopes of the already-very-steep initial rise). By definition, on the cases with no links to background class members, all of the direct methods give zero scores.

Therefore, to assess the potential of the scores with more data, and to compare different direct scores on those cases where they *can* differ, we magnify the far-left part of the curves by looking only at those cases with at least one link to a background member of the class (i.e., ignoring the zero scores). The resultant ROC curves for s_{wend} are shown in Figure 3.

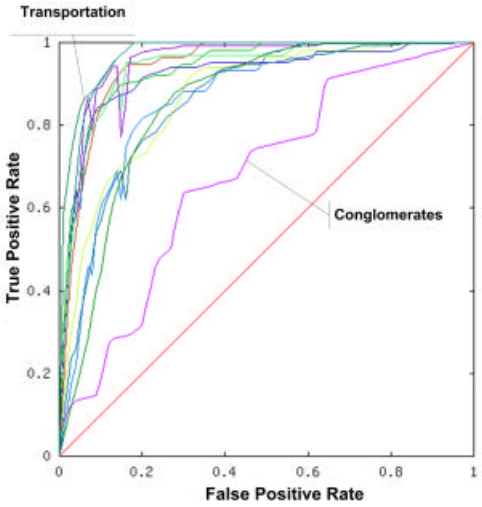

Figure 3: ROC curve for weighted, entity-normalized method, ignoring non-linked entities (averaged over 10 runs)

In Figure 3, most of the AUCs are 0.9 or better, and only one (Conglomerates, AUC=0.67) is less than 0.8. This demonstrates that s_{wend} can separate the entities by class remarkably well, in cases where it has a chance—i.e.,

[3] Giving scores of zero to entities *not* in the class is of course desirable. The problem here is that members of the class are receiving scores of zero. The percentage varies from sector to sector, and can be estimated by (one minus) the TP rate at the beginning of the final linear segment of the ROC curve. E.g., for Transportation approximately 10% of the members of the class receive zeros. For Capital Goods, approximately 50% receive zeros.

where there is at least one link to a known member of the class.

	area under curve (no zeros)			
Sector	s_{end}	s_{wend}	d_{si}	d_{efigf}
BasicMaterials	0.9106	0.9286	0.6442	0.6685
CapitalGoods	0.8321	0.8574	0.5299	0.5676
Conglomerates	0.5755	0.6668	0.7079	0.7169
ConsumerCyclical	0.8205	0.8602	0.5853	0.6107
ConsumerNonCyclical	0.9079	0.9317	0.7482	0.7578
Energy	0.9291	0.9281	0.8283	0.8522
Financial	0.8892	0.9107	0.6243	0.6646
Healthcare	0.9397	0.9405	0.7599	0.8078
Services	0.8143	0.8462	0.5712	0.5970
Technology	0.8373	0.8446	0.7051	0.7195
Transportation	0.9567	0.9624	0.8551	0.9124
Utilities	0.9397	0.9518	0.9076	0.9225
Average	***0.8627***	***0.8857***	***0.7056***	***0.7331***

Table 3: Area under curve (AUC) for all scoring methods ignoring non-linked entities

Table 3 reports the AUCs of all 5 scoring functions for the 12 classification tasks for this task. In most cases all the scoring methods classify considerably better than random (represented by the diagonal in ROC space), but again s_{end} and s_{wend} perform the best. The wend score consistently performs better than the other scores (with only a few exceptions). Table 4 shows the relative error reduction of the s_{wend} over the other methods. Even over s_{end}, it achieves a 15% error reduction on average.

	error reduction (no zeros)		
Sector	s_{end}	d_{si}	d_{efigf}
BasicMaterials	0.2019	0.7994	0.7846
CapitalGoods	0.1506	0.6966	0.6701
Conglomerates	0.2152	-0.1406	-0.1768
ConsumerCyclical	0.2209	0.6628	0.6408
ConsumerNonCyclical	0.2586	0.7290	0.7182
Energy	-0.0152	0.5810	0.5132
Financial	0.1945	0.7624	0.7339
Healthcare	0.0133	0.7521	0.6904
Services	0.1716	0.6413	0.6183
Technology	0.0444	0.4729	0.4458
Transportation	0.1298	0.7402	0.5702
Utilities	0.1994	0.4779	0.3777
Average	***0.1487***	***0.5979***	***0.5489***

Table 4: Relative error reductions for s_{wend} over other methods ignoring non-linked entities

It is important to emphasize that we are not claiming that these results show that s_{wend} is generally preferable. This will be domain and task dependent. For this particular domain, s_{wend} seems to be the better score. This general result is reinforced by examining the results on the finer-grained industry (rather than sector) affiliations. For 34 of the 97 industries the two methods produce identical generalization performance.[4] For the remaining 63 industries, s_{end} is superior for 11 and s_{wend} for 52. Figure 4 plots the AUCs of s_{wend} (vertical axis) and s_{end} (horizontal axis). Points above the diagonal indicate that s_{wend} has a higher AUC than s_{end}. Clearly, s_{wend} is the better performer on these finer-grained classification tasks, sometimes by a large margin.

Returning to the zero scores, the direct RVS method does not stand a chance when there are no links to a known member of the class. The indirect method is not so limited—the only time it will give a non-zero score for a class is if the entity in question is not linked to anything that a known member is linked to. Scoring all the companies with the indirect method indeed produces few zeros. Unfortunately (as shown in Table 1), the classification performance is not nearly as strong with the indirect methods. The indirect methods show a much wider range of performance, from Utilities (almost as good as with the direct score) down to Capital Goods (apparently random).

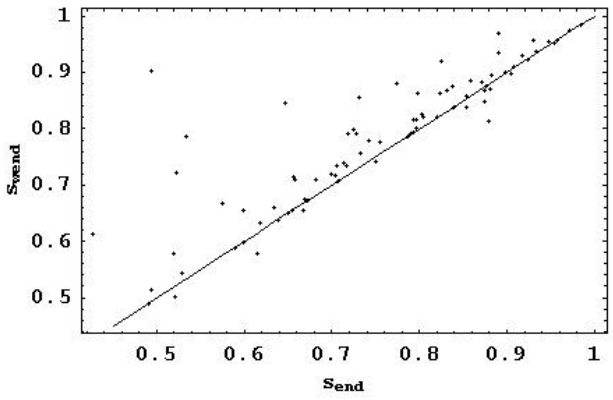

Figure 4. AUC of s_{wend} vs. AUC of s_{end} on the 97 industries

4.2 Hybrid methods

In order to improve the direct methods' performance on entities with no direct links to the class, it is possible to combine the direct and indirect methods, using the latter only when the former returns a zero.

Definition: The *weighted, efigf combined score* of an entity is:

$$s'(e,i) = d_{efigf}(e,i) * \min_k (s_{wend}(e,k))$$

$$cs\ (e,i) = \begin{cases} s_{wend}(e,i) \\ s'(e,i), if\ s_{wend}(e,i) = 0 \end{cases}$$

Thus, we use the weighted, entity normalized direct score s_{wend}, unless s_{wend} is zero, in which case we scale the efigf-score by the minimal, greater-than-zero s_{wend} to fit the d_{efigf}'s below the true weighted, entity normalized scores.

[4] For sparser data the two methods' scorings will become more similar—and exactly identical scorings are not necessary to produce identical ROC curves.

Using this approach, we see a modest improvement. On average we see 4% additional error reduction over s_{wend} (see Table 5). However, there are certain cases where additional error reduction is very large (Transportation, Energy error reduction >20%), and three cases where it increases error (on average 9% relative increase). This illustrates the need for a flexible framework within which a variety of RVS methods can be defined and tested.

Another approach to address the scoring of entities with no links to a known member of the class in question is to investigate degree-2 links (links to entities "two hops" away). Redefining the links in the direct RVS model results in a score, which is analogous to s_{end}, the simple entity-normalized direct RVS score, but follows links of degree two. Consider **w?** to be the analogue to **w,** except with two-hop links.

Definition: An entity e can be described by an *simple second-degree entity vector* $\mathbf{w?_j} = (w?_{j,1}, w?_{j,2}, \ldots)$, where:

$$w?_{j,k} = 1 \text{ if } w_{j,i} * w_{i,k} = 1 \text{ for any } e_i, e_k \text{ in } E$$

Definition: The *second-degree class-normalized direct RVS score* of entity e for class i is the inner product of $\hat{\mathbf{w}}''$ and the canonical class vector $\mathbf{c_i}$, normalized by the L1 norm of **c**.

$$s''_{cnd}(e,i) = \frac{\hat{\mathbf{w}}'' \cdot \mathbf{c}_i}{\sum c_j}$$

Again we can define a combined score:

Definition: The *weighted, second degree class-normalized combined score* of an entity is:

$$s''(e,i) = s''_{cnd}(e,i) * \min_k (s_{wend}(e,k))$$

$$cs''(e,i) = \begin{cases} s_{wend}(e,i) \\ s''(e,i), if\ s_{wend}(e,i) = 0 \end{cases}$$

	area under curve			rel. error red.	
Sector	s_{wend}	***cs***	***cs"***	***cs***	***cs"***
BasicMaterials	0.7339	0.7313	0.7677	-0.0098	0.1270
CapitalGoods	0.6810	0.6525	0.7187	-0.0891	0.1183
Conglomerates	0.7697	0.7702	0.7232	0.0024	-0.2019
ConsumerCyclical	0.7463	0.7178	0.7682	-0.1126	0.0862
ConsumerNonCyclical	0.8753	0.8859	0.8726	0.0850	-0.0215
Energy	0.8682	0.8981	0.9078	0.2267	0.3003
Financial	0.8067	0.7938	0.8129	-0.0671	0.0319
Healthcare	0.8898	0.8945	0.9136	0.0425	0.2163
Services	0.8124	0.8150	0.8234	0.0137	0.0586
Technology	0.8427	0.8458	0.8496	0.0200	0.0437
Transportation	0.9307	0.9470	0.9458	0.2347	0.2177
Utilities	0.9096	0.9185	0.9187	0.0979	0.1011
Average	*0.8222*	*0.8225*	*0.8352*	*0.0370*	*0.0898*

Table 5: AUC and relative error reduction with combined methods

As Table 5 shows this method improves further over s_{wend}. On average we get 9% relative error reduction with some reductions going up to 30% (for energy) and two additional being higher than 20% (Healthcare and Technology). Like with the weighted, efigf combined score cs, however, some sectors have an error increase, the largest being Conglomerates with 20%. (NB: by its nature, Conglomerates is the one sector for which we would not expect members to be linked to each other.) This illustrates that even in a domain where simple scores perform very well, more-complex scores can add value.

4.3 Comparing scores across sectors

The ROC analysis above evaluates the problem: given a sector, how well can companies be separated into those in the sector and those not. More specifically, it evaluates the scoring function's ability to rank the companies by probability of class membership. The dual question is: given a company, how accurately can it be placed into the "correct" sector?

The base rate for this classification problem will be the marginal probability of the most common class: in our data, 0.29 (Technology). The accuracy of s_{wend} for classifying companies into the correct sector was 0.68. Table 6 shows the accuracy for the companies in each sector. For only one sector (Conglomerates) was the classification accuracy worse than the base rate (0.15) and this sector also had the smallest number of members (recall that s_{wend} does not normalize for the size of the class). Classification is one (important) case where comparing scores across sectors is necessary. We will return to this in the follow-up analysis below.

Sector	Correct	Total	Accuracy
Technology	392	505	0.78
Energy	54	71	0.76
Transportation	28	38	0.74
Healthcare	131	180	0.73
Utilities	21	30	0.70
Financial	111	170	0.65
Services	286	444	0.64
ConsumerNonCyclical	38	60	0.63
BasicMaterials	47	104	0.45
ConsumerCyclical	36	99	0.36
CapitalGoods	17	73	0.23
Conglomerates	3	14	0.21
Overall	**1164**	**1788**	**0.65**
base rate (Technology)			*0.28*

Table 6: Accuracy for classifying companies in each sector

4.4 Other methods

How good are these results, with respect to other methods of company-affiliation classification? Our goal in this paper was to demonstrate the RVS model, and not to assess what is the best method for company affiliation identification. Nevertheless, for completeness we address this question briefly.

Running the relational learning program FOIL [10] on these data failed completely, returning a single clause for each company. We modified FOIL to search for more general theories, and it still performed far worse than the RVS methods. In retrospect, this is not surprising because FOIL (and many other ILP [5] algorithms) do not perform

numeric aggregations without having them be defined explicitly. The RVS scores may provide useful constructed features for ILP programs.

We created an ensemble, multi-document, full-text classification method, using the stories from which the links were extracted. This method performed similarly to s_{wend} but was two orders of magnitude slower. Interestingly, when the sector-specific word models were examined, the names of major companies in the sector were given high scores. So the text-based method chose to use these "links" in its own vector-space model.

In the financial literature and industry, companies are clustered into industry groupings based on correlations in their financial time series (and singular-value decompositions) [11]. Our experiments so far with these methods have not yielded remarkable performance on our classification tasks.

Probabilistic and statistically oriented relational learning methods, such as PRMs [12], and relational versions of naïve Bayes [13], decision trees [14], etc., hold the most promise for competing with the RVS model. These methods do perform aggregations over the values of the attributes at linked nodes. In particular, properly utilized (weighted) COUNT or MODE operations would incorporate the fundamentals of the basic, direct RVS scores. However, even if they performed competitively, they far more complex learning procedures than the RVS scoring functions.

5. Discussion and Followup

So, what does our case study illustrate about the relational vector-space model? First, it shows that there are domains where the interlinkage between class members is strong enough for simple scoring methods based only on linkage to capture much of the "signal" needed for good classification. And for some tasks the scoring can lead to remarkable classification accuracy. For example, even though Transportation companies represent only 2% of the companies, the excellent Transportation scores (AUC>0.9) lead to a classification accuracy of 74%, when classifying by choosing the highest sector-score (of the 12).

Intuitively, we expect the direct RVS methods to excel when (as in Figure 1) entities are more likely to be linked to other entities with the same class membership. This intuitive notion is captured more formally by *relational autocorrelation* [15]: the correlation between values of the same attribute on linked entities "represents an extremely important type of knowledge about relational data, one that is just beginning to be explored and exploited for learning statistical models from relational data" (ibid). We can use this notion to understand the RVS model in more detail.

Adapting Jensen & Neville's [15] definition to our context, consider a set of entities E, an attribute *f*, and a set of paths P that connect objects in E.

Definition: Relational autocorrelation C' is the correlation between all pairs $(f(x_1),f(x_2))$ where $x_1, x_2 \in E, x_1 \neq x_2$ and such that $\exists p(x_1, x_2) \in P$.

Let us define *degree-k* relational autocorrelation as further restricting the length of $p(x_1, x_2)$ to be *k*. Intuitively, the direct RVS method should be appropriate when the *degree-1* relational autocorrelation in the entities' class values is high ("homophily"). We can use an existing measure of relational autocorrelation to verify this. Following Jensen & Neville we use Pearson's corrected contingency coefficient to measure class-value autocorrelation.

For our sector-classification problem, the degree-1 relational autocorrelation considering all classes is 0.84, reflecting our intuition from inspecting Figure 1. Figure 5 shows for each class the classification performance (accuracy) plotted against the class vs. not-class degree-1 autocorrelations. The rankings of performance and autocorrelation are very similar (Pearson's correlation coefficient is 0.76). This high value is due to a large part to Conglomerates, which has the lowest autocorrelation and the lowest accuracy. Nonetheless it suggests that the performance of the direct RVS method indeed is related to the degree-1 relational autocorrelation in the class values.

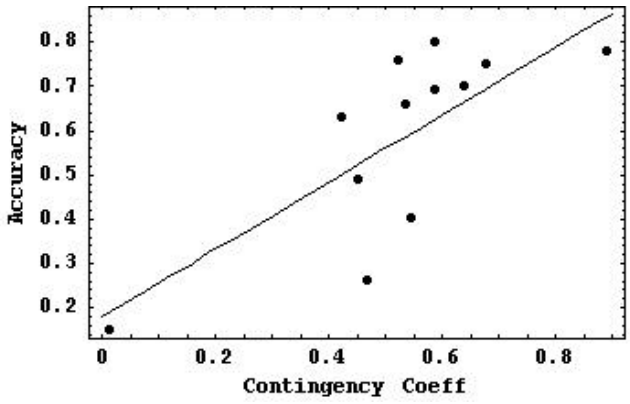

Figure 5: Accuracy versus degree-1 autocorrelation

More specifically, the direct RVS score itself is a measure of degree-2 relational autocorrelation where the path $p(x_1,x_2)$ passes through the entity to be classified. If the degree-1 relational autocorrelation is high, one would expect entities connected by paths of length 2 through an entity of class C, also to have class C (this is the condition for the direct RVS score to be effective for classification).

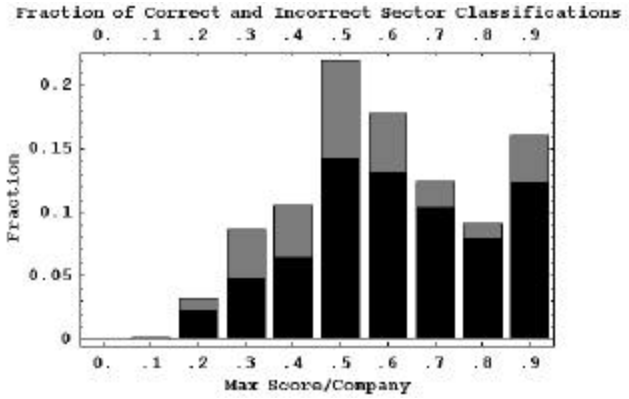

Figure 6: Fraction of correct and incorrect Sector Classifications (black are correct classifications, gray are incorrect classifications)

This suggests that the RVS scores can be used for assessments of the nature of the relational autocorrelation in a graph, that are finer-grained than given by the contingency coefficient. For example, for our sector-classification problem, Figure 6 is a histogram, plotting the distribution of companies over the maximum of s_{wend} for any of the 12 classes. The black (gray) shading shows the percentage of companies with the same (different) class as the class with the maximum score. Interestingly, the distribution shows that for this domain, most (>75%) of the entities have a (weighted) majority of the links to entities of a single class. More often than not, this class is correct.

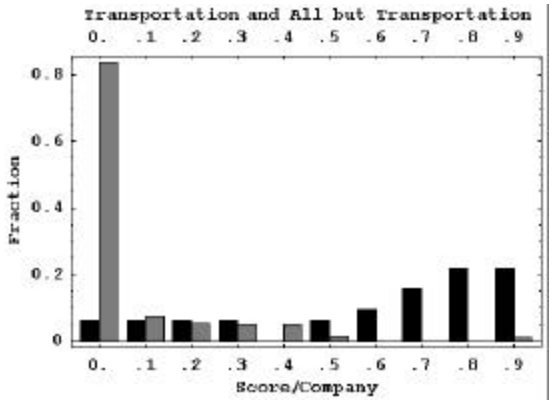

Figure 7: Sector specific s_{wend} scores for Transportation (gray is All but Transportation, black is Transportation)

Let us use s_{wend} to view two of the particular sector classification tasks, Transportation (high AUC & accuracy) and Capital Goods[5] (low AUC & accuracy). Figure 7 and Figure 8 show histograms of the sector-specific s_{wend} scores for the members of the class (black) and the non-members (gray). We can see clearly that Transportation companies are primarily linked to other Transportation companies, and other companies are not. Capital Goods companies, on the other hand, show very different connectivity—they are not primarily linked to other Capital Goods companies. In fact, their linkage to other Capital Goods companies is remarkably similar to that of the rest of the companies.

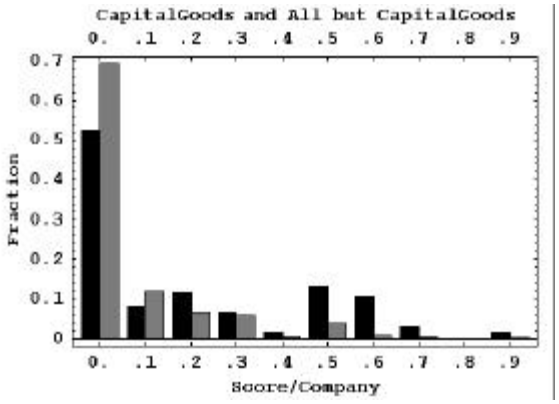

Figure 8: Sector specific s_{wend} scores for Captial Goods (gray is All but CapitalGoods, black is CapitalGoods)

Finally, consider the comprehensive view of class-interlinkage given in Figure 9 (on the last page), which shows the class interlinkage for all class pairs. Each individual graph shows the averages across the members of the class of the s_{wend} scores for each of the 12 classes. This figure gives a condensed visualization of the class-specific interlinkage in the graph.

We argue that this visualization could lead to insights about the classes. Pretend for the moment that we did not already have a basic understanding of the sectors. We see that Capital Goods has high linkage to most of the other classes. Transportation, on the other hand is linked primarily with itself.[6] And Services are linked almost uniformly to the rest of the sectors. Utilities are linked to Energy and Transportation (and in contrast to the rest of the sectors, not to Technology much at all). Each of these properties makes good sense for the corresponding class.

6. LIMITATIONS AND FUTURE WORK

For this study we limited ourselves to relatively simple RVS scoring functions. This was partially due to our desire to flesh out the basics of the model first before getting fancy, but more due to the remarkable performance of the basic methods in our case-study domain.

The RVS scoring functions are "learning" procedures only in the sense that nearest-neighbor classifiers are: they simply apply a scoring function to a database of instances-- no feature selection or parameter estimation takes

[5] Conglomerates is similar, but has only 13 member companies (as compared to 61 for Capital Goods).

[6] We have not normalized here by the size of the class here, in keeping with the rest of the paper (so Technology is weighted heavily across most of the classes). Doing so gives a different, and equally intriguing visualization.

place. Indeed, s_{wend} could be considered a "Relational Neighbor" classifier [16], that takes advantage of class homophily. Provost et al. argue that such a simple model should generally be used as a baseline for more complicated approaches, because it seems to perform remarkably well in many domains [16]. Jensen & Neville found high relational autocorrelation for almost all attributes they examined in linked movie data [15]. Furthermore, homophily has been observed in human groups with respect to a wide variety of descriptive variables, and is one of the basic premises of theories of social structure [17]. Chakrabarti et al. take advantage of autocorrelation in class values to classify hypertext documents [18]. Their procedure learns a probabilistic model based on the classes of related entities, and therefore can capture more complex relationships than simply homophily.

There are several ways in which the current model is limited. We only consider a single link type. This does not restrict the model's applicability, because (as we did in our case study) the type of links can simply be ignored. However, it may obscure information that is important for classification. The model as presented could be extended to handle multiple link types simply by creating multiple vectors (one per link type) and concatenating them. Alternatively, different models could be produced for different link types, and selected among or applied as an ensemble. Whether or not these would be effective techniques is a subject for future study.

We also only consider a single entity type. This is a more fundamental limitation of the model, and we have not considered carefully how to extend it. One obvious way to apply the model to data with multiple types of entities is to focus on one entity type, and consider paths between these entities (perhaps going through other entities) to be the links.

The direct RVS scores (as presented) abstract away most of the graph structure, only considering adjacency. This is the source of the model's elegant simplicity, but it also limits the types of problems on which it will be effective. It could be extended by defining links in the model to be paths of length greater than one. These could be treated similarly to multiple link types, as discussed above.

We have assumed that more data will (partially) resolve the issue with many zero scores (described in Section 4.1). We have little support for this assumption, but it seems reasonable. We have procured another data set to test with; however, we have not yet completed the data preprocessing necessary to make the two data sets comparable.

Finally, we have looked at different sector and industry classifications (SIC codes and Hoover's classification) with qualitatively similar results, but have not studied them comprehensively. We would like to show that the RVS model with newswire-extracted links can model various, different classifications that have little similarity to each other (the aforementioned surprisingly do not) but are nevertheless meaningful.

7. CONCLUSIONS

The relational vector-space model is a useful abstract representation for studying relational classification. With simple choices for its components (entity vector, class vector, normalization function) it represents intuitive notions of classification by relational autocorrelation. With more complicated choices, it can represent more complex classification models on linked data (still abstracting away much of the graph structure).

In our case study of company affiliation classification, relatively simple scoring functions performed remarkably well, illustrating the potential utility of the RVS model. However, the RVS scores may be most useful as feature constructors in other, more complicated systems. Relational learners can include these scores as (additional) aggregation functions. Standard feature-vector learners can use the RVS scores to take into account an important part of relational structure.

The case study also illustrated the advantage of the structure that the RVS model places on the space of scoring functions, allowing them to be explored systematically. Although the improvement for this domain was not dramatic, the results of combining the different scores do suggest that combined RVS scoring models may be advantageous in certain domains.

8. ACKNOWLEDGMENTS

We thank Claudia Perlich, Shawndra Hill, David Jensen, Tom Fawcett, and Roger Stein for discussions during various stages of this project. This work is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement F30602-01-2-585.

9. REFERENCES

[1] S. Dzeroski and N. Lavrac, *Relational data mining*. Berlin; New York: Springer, 2001.

[2] MRDM Workshop on Multi-Relational Data Mining at KDD, Edmonton, Alberta, Canada, July 23, 2002

[3] L. Getoor and D. Jensen, "Learning Statistical Models from Relational Data - AAAI 2000 Workshop," American Association for Artificial Intelligence (AAAI), Menlo Park, California, Technical Report WS-00-006, 2000.

[4] G. Salton and M. J. McGill, *Introduction to modern information retrieval.* New York: McGraw-Hill, 1983.

[5] S. Muggleton, *Inductive logic programming*. London: Academic Press in association with Turing Institute Press, 1992.

[6] K. M. Kahle and R. A. Walking, "The Impact of Industry Classification on Financial Research," *The Journal of Financial and Quantitative Analysis*, vol. 31, pp. 309-335, 1996.

[7] J. Swets, "Measuring the Accuracy of Diagnostic Systems," *Science*, vol. 240, pp. 1285-1293, 1988.

[8] F. Provost and T. Fawcett, "Robust Classification for Imprecise Environments," *Machine Learning*, vol. 42, pp. 203-231, 2001.

[9] D. J. Hand, *Construction and Assessment of Classification Rules*. Chichester, UK: John Wiley and Sons, 1997.

[10] J. R. Quinlan and R. M. Cameron-Jones, "FOIL: A midterm report," 6th European Conference on Machine Learning, 1993.

[11] P. Gopikrishnan, B. Rosenow, V. Plerou, and H.E. Stanley, "Identifying business sectors from stock price fluctuations," Phys. Rev. E 64, 035106R, 2001.

[12] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning Probabilistic Relational Models," 16th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 1999.

[13] J. Neville, D. Jensen, B. Gallagher, and R. Fairgrieve, "Simple estimators for relational Bayesian classifiers," Department of Computer Science. University of Massachusetts Amherst., Amherst, MA, USA, Technical Report 03-04, 2003.

[14] D. Jensen and J. Neville, "Schemas and models," Multi-Relational Data Mining Workshop, 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining., 2002.

[15] D. Jensen and J. Neville, "Linkage and autocorrelation cause feature selection bias in relational learning," Nineteenth International Conference on Machine Learning (ICML2002), 2002.

[16] F. Provost, C. Perlich, and S. Macskassy, "Relational learning problems and simple models." IJCAI-2003 Workshop on Learning Statistical Models from Relational Data, 2003.

[17] P. Blau. *Inequality and Heterogeneity: A Primitive Theory of Social Structure*. New York: The Free Press, 1977.

[18] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks. ACM SIGMOD 1998.

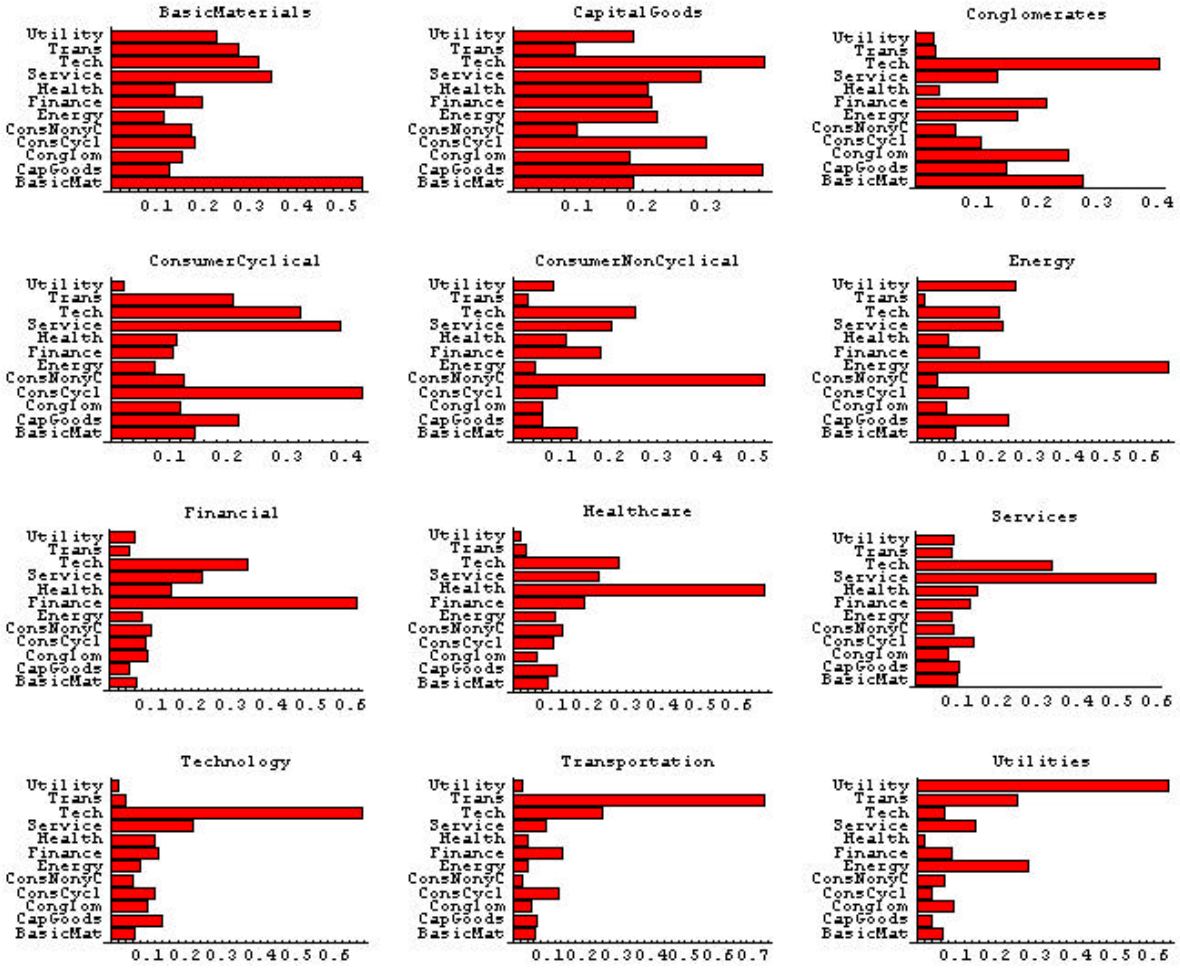

Figure 9. Average class-specific s_{wend} scores by class, as visualization of class interlinkage in graph