Conference papers

School of Computer Science

2010-01-01

CBTV: Visualising Case Bases for Similarity Measure Design and Selection

Brian Mac Namee
Technological University Dublin, brian.macnamee@tudublin.ie

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

CBTV: Visualising Case Bases for Similarity Measure Design & Selection

Brian Mac Namee and Sarah Jane Delany

DIT AI Group, Dublin Institute of Technology, Dublin, Ireland
brian.macnamee@dit.ie,sarahjane.delany@dit.ie

Abstract. In CBR the design and selection of similarity measures is paramount. Selection can benefit from the use of exploratory visualisation-based techniques in parallel with techniques such as cross-validation accuracy comparison. In this paper we present the Case Base Topology Viewer (CBTV) which allows the application of different similarity measures to a case base to be visualised so that system designers can explore the case base and the associated decision boundary space. We show, using a range of datasets and similarity measure types, how the idiosyncrasies of particular similarity measures can be illustrated and compared in CBTV allowing CBR system designers to make more informed choices.

1 Introduction

Similarity measures are widely used in instance-based learning and more specifically in nearest neighbour classification. The importance of using the appropriate similarity measure in a k-NN classifier is well known and there has been significant work on proposing, learning and evaluating new similarity measures [28, 21, 16, 22, 25, 3].

In most cases, the selection of the best measure is based on a comparison of similarity measure performance in a cross-validation evaluation. Our proposal in this paper is to provide a tool, the Case Base Topology Viewer (CBTV), for case base visualisation. This will allow case-based system designers to explore different similarity measures through visualisations in order to assist them in choosing the most appropriate measure for the case base in question.

CBTV is based on the force-directed graph algorithm, similar to other work on case base visualisation [20, 19], and projects a case base onto a two dimensional plane such that cases that are similar to one-another appear closer together. Changing from one similarity measure to another, users can examine the class separability possible with different measures, which assists in their decision as to which similarity measure is most appropriate for their current task.

The contributions of this paper are the presentation of the CBTV system and the description of a number of illustrative examples that show how the differences between similarity measures can be explored through visualisations. The paper is organised as follows: Section 2 gives an overview of existing work, both in the context of designing or choosing the best similarity measure to use in a case-based system, and in the context of visualisation of case bases. Section 3 then

describes the CBTV system and how it can be used to visualise the impact of using different similarity measures. Section 4 then provides a series of evaluation examples of differing visualisations of various case bases - including numeric only datasets, textual datasets and heterogeneous datasets. Finally, in Section 5 we draw conclusions on some of the advantages and disadvantages of using a system such as CBTV and present a number of avenues for further research.

2 Related Work

A variety of similarity measures are proposed in the literature for use in case-based systems. Cunningham [5] provides a comprehensive taxonomy of strategies for similarity assessment in CBR, grouping measures into four different categories. The most well known of these is the *direct* category where cases are represented as feature vectors and similarity is assessed directly from these features. Less frequently used categories include *transformation-based* measures where similarity between cases is based on the effort required to transform one case into another case (e.g. Edit Distance and Earth Mover Distance); *information-theoretic* measures where the measure is based on the information content of the case, the most dramatic of these being Compression-Based Similarity; and, finally, *emergent measures* such as Random Forests which exploit the considerable computational power now available.

Most of the existing research into choosing similarity measures has focussed on (i) proposing or (ii) learning new *direct* measures. By and large the most commonly used *direct* similarity measure is the generalised weighted similarity measure where the similarity between two cases is a function (often linear) of individual weighted local feature similarities. The task is then to set the individual feature weights and Wettschereck *et al.* [27] introduced a five dimensional framework for categorising automated feature weight setting methods and included a comprehensive review of early work in this area.

Aside from setting feature weights, there has been work presenting a formal framework for the specification and construction of a wide range of heterogeneous similarity measures [22] and proposing new heterogenous and hybrid measures [21, 16]. More recently learning similarity measures has received more attention. A large proportion of research in this area uses incremental or evolutionary algorithms. These approaches can use system performance feedback [13, 9], introspective and reinforcement learning [2, 23] or qualitative information such as the ranking of cases [3, 4]. Xiong & Funk [29] introduce a novel learning approach with the advantage that the importance of a feature is built into a local similarity measure and there is no need to learn individual feature weights.

In most, if not all, of this previous work, measures such as classification accuracy in cross validation are used as a measure of the performance of the similarity measures, showing which similarity measure is most appropriate for the case base in question. Our approach to selecting the most appropriate similarity measure is to provide a visualisation of the case base for a specific similarity measure

which shows how the cases are distributed and where decision boundaries are located. It is not intended that this approach replace the use of performance measures, but rather that it complements these techniques and allows a system designer to further explore a case base and associated decision boundary space.

Previous work in case base maintenance has used the visualisation of the case base and its competence model to improve the efficiency of the authoring process and the quality of the resulting case base [20]. Case-base visualisation has also been used to provide dynamic visualisations of case base usage over the life-cycle of a case base [19]. Both approaches use the force-directed graph drawing algorithm known as the *spring model* [10] which allows the display of n-dimensional data on a 2-dimensional plane.

A second category of visualisation focusses more on showing the relationships between cases by visualising the connections between feature values. Falkman [11] visualises clinical experience as a three dimensional cube based on the idea of 3D parallel co-ordinate plots. A case feature can be viewed as a two-dimensional plot with the cases plotted in some order on the x-axis and the possible feature values on the y-axis. The Cube is a collection of planes, each plane representing a feature in the case representation and each case is represented as a line connecting individual values in the different feature planes. Massie *et al.* [18] adopts a similar approach to visualisation for explanation in pharmaceutical tablet formulation. They use a two-dimensional version with each feature represented as a vertical line rather than a plane, with feature values plotted along this line and each case represented by connecting the feature lines at positions reflecting the case feature value. Clusters of similar cases can be observed easily with this parallel co-ordinate plot approach and it facilitates the identification of similarities and differences between cases and groups of cases.

The limitation of these approaches is that they do not support the visualisation of large dimensional case bases. While some of the approaches mentioned can deal with high-dimensional data (e.g. [20][19]) Kontkanen *et al.* directly address this problem [14]. Visualisations are created in which, rather than basing case positions on their pairwise Euclidean distances, they use the output of a Bayesian network - the intuition being that cases that result in similar network output should appear close together in the visualisation. This, they argue, gives better visualisations than the use of Euclidean distance measures. However, they fail to acknowledge the fact that Euclidean distance is not the only similarity measure available. We examine whether it is possible to create more interesting visualisations using more sophisticated similarity measures.

3   The CBTV System

This section will describe the CBTV system and how it can be used to visually explore the suitability of using various similarity measures with particular datasets. Firstly, we will describe how the system generates visualisations using a force directed graph drawing algorithm, or the spring model. Then we will describe how these visualisations can be fine tuned using transformations of the

calculated similarity values. Finally, we will describe the technique we use to evaluate how well a particular visualisation matches an underlying dataset.

3.1 Representing Case Bases Using a Force-Directed Graph Drawing Algorithm

Similarly to Smyth *et al.* [24], we create visualisations of case bases using a force directed graph drawing algorithm [10]. We consider a case base as a maximally inter-connected graph in which each case is represented as a node that is linked to every other case (or node) in the case base. An example is shown in Fig. 1(a) in which cases a, b and c are shown as a maximally inter-connected graph. The layout of the graph should be such that cases most similar to each other appear close together in the graph.
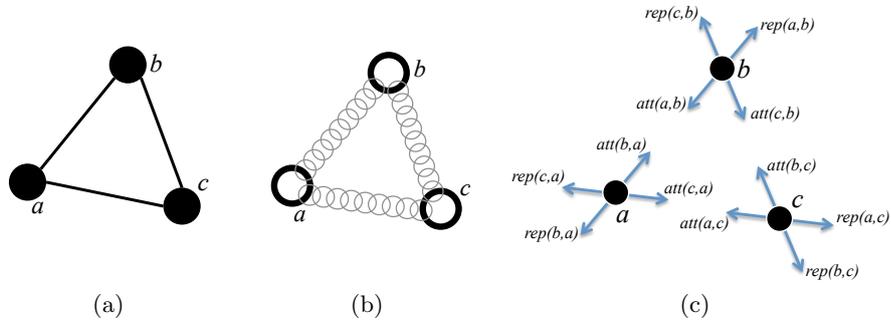
Fig. 1. Case base representations used by the force-directed graph drawing algorithm

To allow similarities between cases dictate the arrangement of the graph the metaphor of springs is used. Each case is imagined as a steel ring with springs connecting it to every other case (illustrated in Fig. 1(b)). The strength of the spring between any two cases is proportional to the similarity between the two cases, i.e. cases that are similar are linked by stronger springs than cases that are dissimilar. In order to create a visualisation, the cases in a case base are initially placed in random positions on a two-dimensional plane (a three dimensional visualisation could also be created in the same way, although this is not something that we explore in this work). The cases are then allowed to move according to the forces applied to them until equilibrium is reached. As the system is allowed find its own equilibrium the stronger springs will draw together those cases that are most similar to each other. So that all cases do not form a single small group, a repulsive force between each case is also introduced.

The *attractive* forces that pull cases together and *repulsive* forces that push cases apart are illustrated in Fig. 1(c), in which each case is shown as having

an attractive and repulsive force due to each other case. Thus the total force exerted on any case v, $f(v)$, can be calculated as follows:

$$f(v) = \sum_{u \in N(v)} att(u,v) + \sum_{u \in N(v)} rep(u,v) \tag{1}$$

where $att(u,v)$ is the attractive force exerted on case v by case u; $rep(u,v)$ is the repulsive force exerted on case v by case u; and $N(v)$ is the set of vertices emanating from case v. Attractive forces are dictated by the strength of the springs between cases and are calculated using Hooke's law - i.e. $att(u,v)$ is proportional to the distance between u and v and the *zero energy length* of the spring. The zero energy length of a spring is the length at which the spring will exert no attractive forces and is directly proportional to the similarities between the cases it connects. The repulsive forces between cases are modelled as Newtonian gravitational forces, and so follow an inverse square law.

Thus, Equation 1 can be expanded as follows (only the x component of the forces is shown - the y component is calculated similarly):

$$f_x(v) = \sum_{u \in N(v)} k_{att} * \frac{(dist(u,v) - zero(u,v)) * (v_x - u_x)}{dist(u,v)}$$
$$+ \sum_{u \in N(v)} k_{rep} * \frac{(u_x - v_x)}{dist(u,v)^3} \tag{2}$$

where $dist(u,v)$ is the distance on the graph between nodes u and v; $zero(u,v)$ is the zero energy length of the spring between u and v; u_x and v_x are the x components of the positions of cases u and v respectively; k_{rep} is the repulsive force coefficient; and k_{att} is the attractive force coefficient. k_{att} and k_{rep} are used to balance the effects of the attractive and repulsive forces. Based on recommendations from [12], and some preliminary experiments, we set $k_{att} = \frac{1}{3}$ and $k_{rep} = 3$.

The zero energy length of each spring should be proportional to the similarity between the cases it connects, and is calculated as follows:

$$zero(u,v) = (1 - sim(u,v)) * maxZeroEnergyLength \tag{3}$$

where $sim(u,v)$ is the similarity between cases u and v, and $maxZero-EnergyLength$ is the distance apart that two cases should appear in the visualisation graph if they are totally dissimilar to each other (i.e. the zero energy length of a very weak spring). $maxZeroEnergyLength$ is a constant that, again following recommendations from [12], we set to $4,900$. In effect, $maxZeroEnergyLength$ determines the size of the visualisation created.

Equation 2 calculates the forces applied to each case in a visualisation at a specific moment in time. To animate the way in which a visualisation finds equilibrium, the forces are repeatedly calculated and small movements are applied to the cases until equilibrium is reached. In order to avoid large jumps across the graph space these small movements are limited as follows:

$$f_x(v) = \begin{cases} -5 & \text{for} & f_x(v) \leq -5 \\ f_x(v) & \text{for} & -5 < f_x(v) < 5 \\ 5 & \text{for} & f_x(v) \geq 5 \end{cases} \qquad (4)$$

The system at present is capable of handling case bases including thousands of cases. However, further work will be required in order to make it scalable to case bases beyond this size.

3.2 Using Similarity Transformations to Aid Visualisations

All of the similarity measures described in this paper are designed to produce similarity scores in the range $[0, 1]$. However, with certain case bases some similarity measures can return collections of similarity values that have very skewed distributions. This can result in visualisations where cases tend to be maximally separated (when the distributions are skewed towards zero) or bunched very tightly together which does not allow patterns to be seen (when similarity distributions are skewed towards one). In these scenarios it is useful to perform a transformation on the similarity values in order to create more useful visualisations. The CBTV system uses power law transformations to allow fine tuning by a user. Power laws transformations were chosen as they are straightforward to implement and were shown empirically to be effective. Each similarity value in the pair-wise similarity matrix used to create visualisations is transformed according to $s' = s^\gamma$ where s is the original similarity value, s' is its transformed equivalent, and γ is a value chosen by the user.

Fig. 2 shows an example for a small classification dataset. It is difficult to interpret the visualisation shown in Fig. 2(a) as all of the cases are overly clustered together due to the distribution of the similarities being skewed, evident from the similarity histogram shown in Fig. 2(a). However, by applying a power law transformation with $\gamma = 4$ the skew can be corrected, leading to a much more useful visualisation, as shown in Fig. 2(b).

However, these sorts of transformations must be used with care, particularly when comparing different similarity measures. One may draw the conclusion that one similarity measure creates a better separation of classes than another, when in fact it is the case that a more appropriate transformation has been applied to the first visualisation than to the second. For this reason in the scenarios presented in Section 4 the most appropriate transformations are applied in each individual case. The appropriateness of a transformation is measured by how well the similarities in the original feature space and the distances on the resulting visualisation correlate. This is discussed in the Section 3.3.

3.3 Measuring the Quality of Visualisations

Creating a force directed graph model of a case base involves projecting a series of multi-dimensional cases to a two dimensional spatial representation. There is a danger in this projection that the lower dimensional version will not be capable of
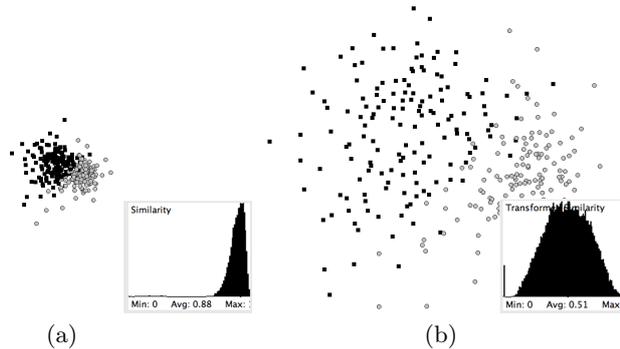
Fig. 2. The effect on a visualisation of a similarity transformation

suitably representing the higher dimensional information. In fact, we should *not* expect the lower dimensional representation to contain all of the information in the higher dimensional version. However, unless the lower dimensional version is a reasonable approximation of the higher dimensional version the visualisations created will not be useful.

In order to evaluate this projection correlation tests following Smyth *et al.* [24] are performed. The correlation tests evaluate the Pearson correlation between the pair-wise similarity matrix generated for the case base and the pair-wise distance matrix calculated from the visualisation. All correlations are negative because distances and similarities are being correlated. As long as a reasonable correlation exists between the two matrices, it is acceptable to interpret the potential of a similarity measure from the visualisation which uses it. However, if a reasonable correlation does not exist, then the visualisation is not useful. Preliminary experiments have suggested that correlations above $|0.6|$ indicate that a visualisation is a useful representation of the case base.

4 Demonstration and Discussion

In this section we will demonstrate how CBTV can be used to explore the suitability of different similarity measures for different case bases. Before presenting the actual visualisations, the datasets used and the similarity measures available for comparison when visualising each of these datasets are described.

4.1 Datasets and Similarity Measures

The datasets used in this work are listed in Table 1. Also shown are the size of each dataset and the similarity measures that have been used for each dataset type. Three different types of dataset have been used: textual datasets, numeric datasets, and heterogeneous datasets.

Table 1. The datasets used in the demonstration of the CBTV system with their sizes and the range of similarity measures available for each type.

Type	Similarity Measures	Name	#Instances	#Features
Textual	Euclidean, Cosine, Jacard, Normalized Compression Distance	Spam-1 Spam-2 Reuters	500 500 500	$4,449$ $3,219$ 828
Numeric	Euclidean, Manhattan, Chebyshev, Mahalanobis	Breast Cancer	569	30
Heterogeneous	Basic, Designed	Camera	210	19

Three textual classification datasets are included: two spam detection datasets [7], and a binary classification dataset generated from the Reuters collection[1]. For the textual datasets we have experimented with three feature-based similarity measures: normalised Euclidean distance [5], cosine similarity [1], and the Jacard index [5]; and one feature-free similarity measure: Normalized Compression Distance (NCD) [15]. For the three feature-based measures, texts are tokenised at the word level and feature values are recorded as unit length normalised word frequencies. Stop-word removal and document frequency reduction (removing all words that occur in less than 3 documents in the dataset) was performed on each dataset. Normalized Compression Distance was implemented using the gzip compressor as described in [6].

The numeric dataset is the Breast Cancer classification dataset [26] from the UCI Machine Learning Repository[2]. Four similarity measures are considered: normalised Euclidean distance, normalised Manhattan distance, Chebyshev distance [5] and Mahalanobis distance [17].

Although all of the classification datasets used represent binary classification problems, the system works equally well for classification problems with any number of classes. Table 2 shows 5*10 fold cross validation average class accuracies using a 3 nearest neighbour classifier for each dataset using each similarity measure (the best result for each dataset is shown in bold).

Table 2. Cross validation accuracies for each dataset.

Dataset	Cosine	Euclidean	Jaccard	NCD
Spam-1	96.96	97.56	94.6	**98.4**
Spam-2	96.12	96.84	95.4	**97.92**
Reuters	95.36	91.64	94.88	**95.8**

Dataset	Chebyshev	Euclidean	Manhattan	Mahalanobis
BreastCancer	94.8	96.66	**96.9**	81.98

(a) Cosine Similarity (corr: −0.846)   (b) Euclidean Distance (corr: −0.854)

(c) Jacard Index (corr: −0.748)   (d) NCD (corr: −0.781)
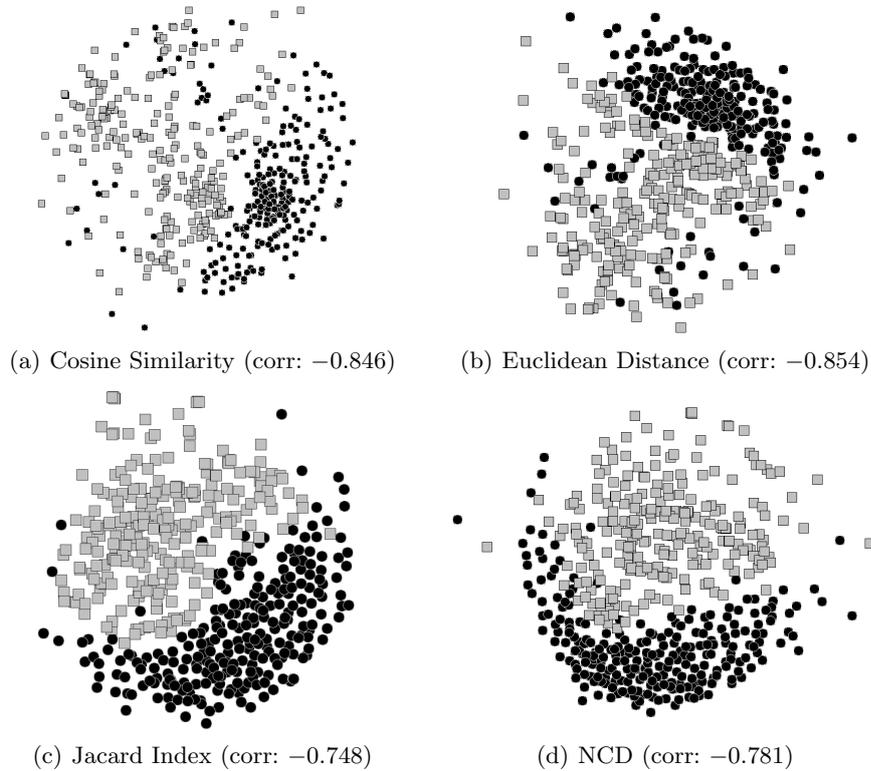
Fig. 3. Visualisations of the Spam-1 dataset using different similarity measures

The final dataset considered in this paper, the Camera dataset[3], is a heterogeneous case base which contains both numeric and discrete features and examines choices made by customers in selecting a camera to purchase. Two types of similarity measure are considered for this dataset. The first is a straightforward direct type distance metric and the second type is a *designed* direct measure which takes into account domain knowledge to create feature-level utility-type similarity measures such as those in [8].

4.2 Exploring Datasets Using CBTV

Fig. 3 shows visualisations (and correlation scores) of the Spam-1 dataset using the four specified similarity measures (with γ equal to 1, 0.3, 0.3 and 0.3 for the Cosine, Euclidean, Jacard and NCD measures respectively) and tells an interesting story. Figs. 3(a), 3(b) and 3(c) show that when the Cosine, Euclidean and Jacard measures are used, the two classes do not appear to separate well

---

[3] Available at: http://cbrwiki.fdi.ucm.es/wiki/index.php/Case_Bases

and there is an amount of confusion at the class boundary. However, in Fig. 3(d) much better class separation is apparent. This would suggest to us that either the NCD similarity measure is the most appropriate for this problem - which is backed up by the accuracies in Table 2.

(a) Cosine Similarity (corr: -0.919)   (b) Euclidean Distance (corr: -0.831)

(c) Jacard Index (corr: -0.701)   (d) NCD (corr: -0.742)
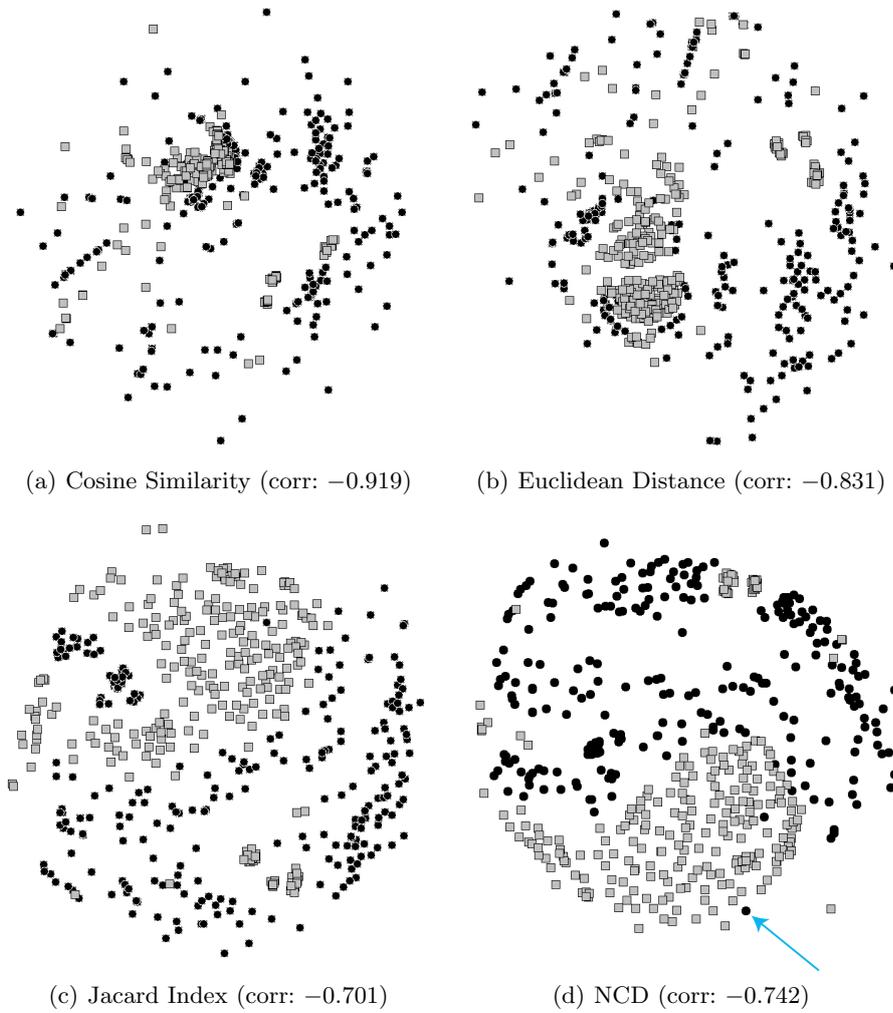
Fig. 4. Visualisations of the Spam-2 dataset using different similarity measures

Fig. 4 shows a set of visualisations for the Spam-2 dataset (no transformations are used) and tells a similar, if less pronounced, story. The visualisation showing the use of the NCD similarity measures, Fig. 4(d), displays the cleanest class

separation which again suggests that this is likely to be the most successful similarity measure for this problem. This is in agreement with the results in Table 2 and confirms previous work on feature-free similarity measures in [6].

(a) Cosine Similarity (corr: −0.874)   (b) Euclidean Distance (corr: −0.854)

(c) Jacardian Similarity (corr: −0.778)   (d) NCD (corr: −0.828)
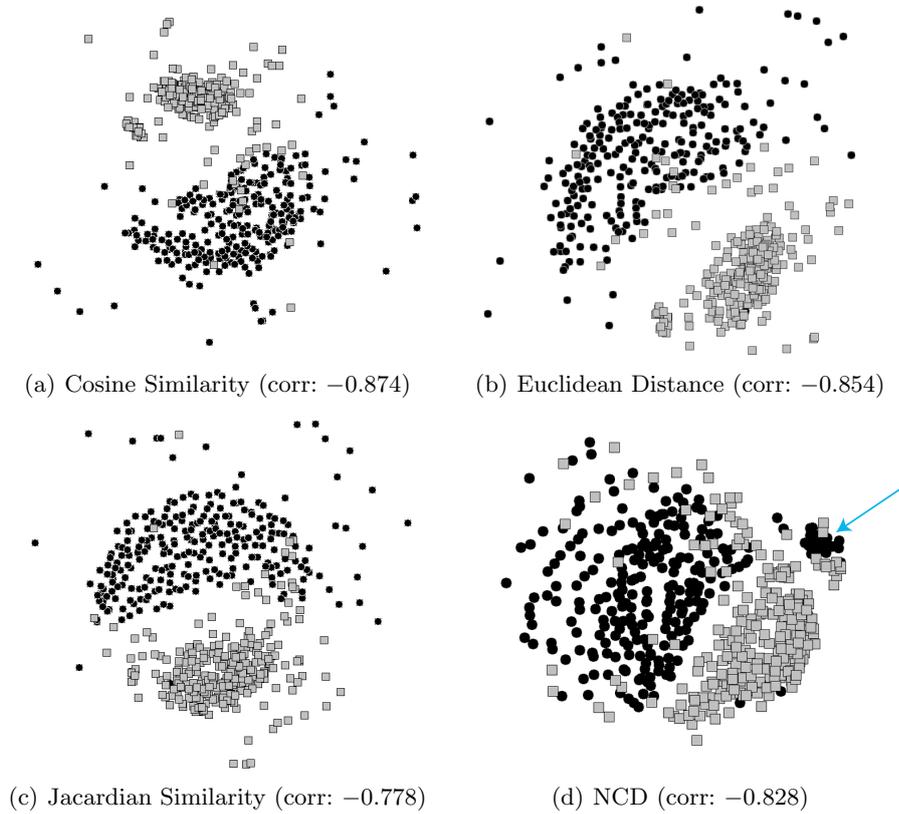
Fig. 5. Visualisations of the Reuters dataset using different similarity measures

Fig. 4(d) also illustrates how CBTV can be used for noise detection. A lone spam example, denoted with the arrow, is seen amongst a number of non-spam examples. CBTV allows users to select an example and view the underlying case (in this case the original spam email). Investigation showed that this email is a one of a number of new journal issue notifications that the user originally classified as non-spam but over time began to classify as spam.

Fig. 5 shows the set of visualisations for the Reuters text dataset. In all of these visualisations a power law transformation with $\gamma = 0.3$ is used. This time it is particularly interesting to note the clustering in the region marked with the arrow in Fig. 5(d). This small cluster, which includes cases from each class, represents documents that have been truncated and appended with the text "*blah*

11

blah blah". The NCD similarity measure responds particularly strongly to this repeated text, a connection that is not evident in the other visualisations. This is a clear example of how visualising different similarity measures can illustrate how each responds to particular characteristics of the underlying dataset. The visualisations of the Reuters dataset also suggest the existence of possible *outliers*, or unusual cases, particularly in Figs. 5(a), 5(b) and 5(c). This is another useful feature of creating visualisations.

(a) Chebyshev Distance (corr: −0.912)　　(b) Euclidean Distance (corr: −0.939)

(c) Manhattan Distance (corr: −0.948)　　(d) Mahanalobis Distance (corr: −0.741)
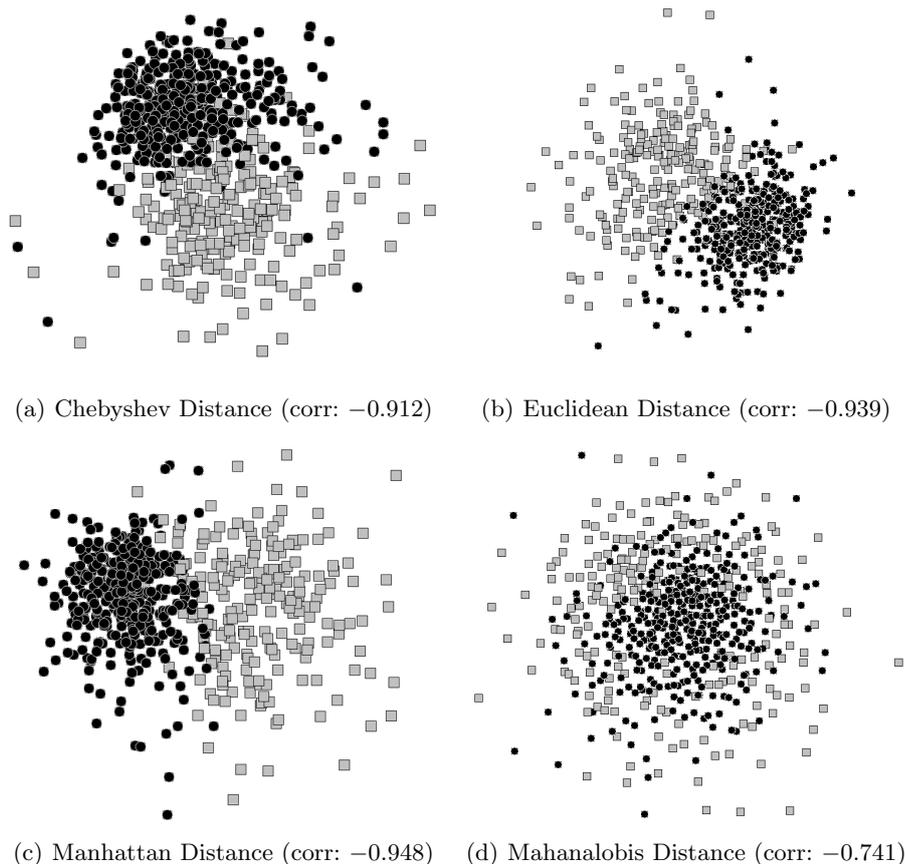
Fig. 6. Visualisations of the Breast Cancer dataset using different similarity measures

Fig. 6 shows the visualisations created for the numeric BreastCancer dataset (power-law transformations with γ equal to 5, 5, 3 and 2 are used for the Euclidean, Manhattan, Chebyshev and Mahalanobis measures respectively). These visualisations tell a slightly different story than those of the text-based datasets. The best class separation appears using the Manhattan and Euclidean measures,

which is not surprising as these are the predominant measures used for this kind of dataset. The visualisation based on the Chebyshev measure (Fig. 6(a)) shows reasonable separation, while for the Mahalanobis measure (Fig. 6(d)) class separation is poor, although an interesting and very distinctive pattern is present. This visualisations match the cross validation results shown in Table 2.

(a) Basic Similarity (corr: −0.66)   (b) Designed Similarity (corr: −0.65)
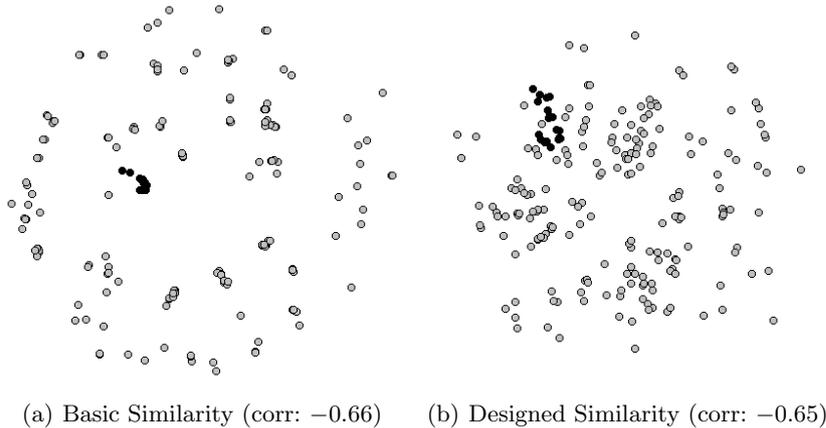
Fig. 7. Visualisations of the Camera dataset using different similarity measures

Finally, Fig. 7 shows the visualisations for the heterogeneous Camera dataset - the only non-classification dataset used in this paper. We created visualisations for a number of heterogeneous datasets using basic and the designed measures and in general found that differences were minimal. This might suggest that the extra effort of designing a similarity measure to incorporate domain knowledge is not worthwhile. However, the Camera dataset proved quite interesting and illustrates the benefits of domain-focussed similarity measures. Fig. 7(a) shows the visualisation of the basic direct measure where there is a small number of very tightly packed clusters of cases. In Fig. 7(b) cases are much better spread out across the visualisation space as a result of the domain knowledge introduced by the utility-type similarity measure used. We illustrate this by highlighting in black one of the clusters in Fig. 7(a) and highlighting in black the same cases in Fig. 7(b). It is clear that in Fig. 7(b) the cases are better spread out.

5   Conclusions and Future Work

In CBR the design and selection of similarity measures is paramount. While techniques such as the comparison of cross-validation accuracies can be used for this, it is also useful to have more exploratory techniques available and the CBTV system has been built for this purpose. Using a force-directed graph drawing

algorithm visualisations of cases bases using different similarity measures can be created and compared. Furthermore, by allowing the user to query the underlying cases in areas of particular interest, possible noisy cases and probable outliers can be investigated. In this paper we have presented the CBTV system and illustrated a range of visualisations that can be created using it for a variety of case base and similarity measure types. We believe that these worked examples illustrate the usefulness of the system.

In the future we intend to extend this work in a number of potentially interesting directions. The first of these is to look at how the visualisations can be given broader coverage and made more interactive. For example, we will allow interactive selection of particular features to use within a similarity measure and interactive adjustment of feature weights. Focusing on text, we will allow for pre-processing tasks such as stop-word removal and stemming to be activated or deactivated at run-time and, finally, allow for visualisation of interactive machine learning tasks such as active learning. Finally, there is a random element present in the creation of visualisations (cases are initially placed randomly in the visualisation space) and there can be small differences in repeated visualisations of a dataset. We will investigate techniques such as repeatedly running the visualisation to an equilibirum state and using an aggregation of these results in the final visualisation in order to address this issue.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
2. Bonzano, A., Cunningham, P., Smyth, B.: Using introspective learning to improve retrieval in CBR. In: ICCBR 1997. pp. 291–302 (1997)
3. Cheng, W., Hüllermeier, E.: Learning similarity functions from qualitative feedback. In: Advances in Case-Based Reasoning, Procs. ECCBR 2008. LNCS, vol. 5239, pp. 120–134. Springer (2008)
4. Coyle, L., Cunningham, P.: Improving recommendation ranking by learning personal feature weights. In: ECCBR'04. LNCS, vol. 3155, pp. 560–572 (2004)
5. Cunningham, P.: A taxonomy of similarity mechanisms for case-based reasoning. IEEE Transactions on Knowledge and Data Engineering 21, 1532–1543 (2008)
6. Delany, S.J., Bridge, D.: Textual case-based reasoning for spam filtering: A comparison of feature-based and feature-free approaches. Artificial Intelligence Review 26(1-2), 75–87 (2006)
7. Delany, S.J., Cunningham, P., Coyle, L.: An assessment of case-based reasoning for spam filtering. Artificial Intelligence Review Journal 24(3-4), 359–378 (2005)
8. Doyle, D., Cunningham, P., Bridge, D., Rahman, Y.: Explanation oriented retrieval. In: Advances in Case-Based Reasoning, Procs. ECCBR '04 (2004)
9. Dubitzky, W., Azuaje, F.J.: A genetic algorithm and growing cell structure approach to learning case retrievel structures. Soft computing in case based reasoning pp. 115–146 (2001)

10. Eades, P.: A heuristic for graph drawing. Congressus Nutnerantiunt 42, 149–160 (1984)
11. Falkman, G.: The use of a uniform declarative model in 3D visualisation for case-based reasoning. In: Craw, S., Preece, A.D. (eds.) ECCBR. LNCS, vol. 2416, pp. 103–117. Springer (2002)
12. Huang, M.L., Eades, P., Lai, W.: On-line visualization and navigation of the global web structure. Journal of Software Eng and Knowledge Eng 13(1), 27–52 (2003)
13. Jarmulak, J., Craw, S., Crowe, R.: Genetic algorithms to optimise cbr retrieval. In: Proceedings of EWCBR '00. pp. 136–147. Springer-Verlag, London, UK (2000)
14. Kontkanen, P., Lahtinen, J., Myllymäki, P., Silander, T., Tirri, H.: Supervised model-based visualization of high-dimensional data. Intelligent Data Analysis 4(3-4), 213–227 (2000)
15. Li, M., Chen, X., Li, X., Ma, B., Vitanyi, P.: The similarity metric. In: Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 863–872 (2003)
16. Liao, T.W., Zhang, Z., Mount, C.: Similarity measures for retrieval in case-based reasoning systems. Applied Artificial Intelligence 12(4), 267– 288 (1998)
17. Maesschalck, R.D., Jouan-Rimbaud, D., Massart, D.L.: The mahalanobis distance. Chemometrics and Intelligent Laboratory Systems 50(1), 1 – 18 (2000), http://www.sciencedirect.com/science/article/B6TFP-3Y8VGYK-1/2/92ac3e8ac922320df9a62b096aca9bee
18. Massie, S., Craw, S., Wiratunga, N.: Visualisation of case-base reasoning for explanation. In: Procs. ICCBR'04 Workshop on Long-Lived CBR Systems. pp. 135–144 (2004)
19. McArdle, G., Wilson, D.C.: Visualising case-base usage. In: Procs. of ICCBR'03 Workshop on Long-Lived CBR Systems (2003)
20. McKenna, E., Smyth, B.: An interactive visualisation tool for case-based reasoners. Applied Intelligence 14(1), 95–114 (2001)
21. Núñez, H., Sànchez-Marrè, M., Cortés, U., Comas, J., Martínez, M., Rodríguez-Roda, I., Poch, M.: A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations. Environmental Modelling and Software 19(9), 809–819 (2004)
22. Osborne, H., Bridge, D.: Similarity metrics: A formal unification of cardinal and non-cardinal similarity measures. In: Leake, D.B., Plaza, E. (eds.) Case-Based Reasoning Research and Development (Procs. of the 2nd International Conference on Case-Based Reasoning). pp. 235–244. LNAI 1266, Springer (1997)
23. Ricci, F., Avesani, P.: Learning a local similarity metric for case-based reasoning. In: ICCBR-95. pp. 301–312. Springer-Verlag (1995)
24. Smyth, B., Mullins, M., McKenna, E.: Picture perfect: Visualisation techniques for case-based reasoning. In: Procs of ECAI 2000. pp. 65–72 (2000)
25. Stahl, A., Gabel, T.: Using evolution programs to learn local similarity measures. In: Procs. ICCBR 2003. LNCS, vol. 2689, pp. 537–551. Springer (2003)
26. Street, W., Wolberg, W., Mangasarian, O.: Nuclear feature extraction for breast tumor diagnosis. In: IS&T/SPIE 1993. pp. 861–870 (1993)
27. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif. Intell. Rev. 11(1-5), 273–314 (1997)
28. Wilson, D., Martinez, T.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Review 6, 1–34 (1997)
29. Xiong, N., Funk, P.: Building similarity metrics reflecting utility in case-based reasoning. Journal of Intelligent and Fuzzy Systems 17(4), 407–416 (2006)