Reduct-Based Result Set Fusion for Relevance Feedback in CBIR

Samar Zutshi Campbell Wilson Bala Srinivasan
Monash University, Australia
Samar.Zutshi@infotech.monash.edu.au

Abstract

Relevance feedback (RF) is a widely used technique to deal with the issues of user subjectivity and the semantic gap in Content-Based Image Retrieval (CBIR). We build on existing work that outlined a rough set based general framework called CAFé for RF and proposed a re-weighting strategy based on a rough set theoretic analysis of the user feedback. This paper presents a method that uses the approximation of the information need distilled from the user classification as the basis for multiple distinct retrievals. The final result set that is presented as the subsequent iteration to the user is obtained by fusing the result sets from the different retrievals. The method is demonstrated in the context of a simple test image collection for clarity. An analysis of the sample iterations of feedback is presented. The method presented remains independent of the retriever, relies on a conceptually appealing model of the user feedback and serves to establish the utility of the general framework.

1. RF as Classification

In Content-Based Information Retrieval (CBIR) two of the most challenging issues are user subjectivity (a given image may cause different perceptions in different users) and the so-called semantic gap (the problem of inferring high level concepts from low-level features) [10].

Making retrieval systems more user-centric and relying on user interaction can help mitigate the effects of these two issues. Relevance feedback (RF), is a widely used technique that allows users to interact with a retrieval system by indicating their opinion of a retrieved results. There is a vast body of work on RF in CBIR ranging from early techniques directly inspired by text retrieval through to sophisticated machine learning based approaches.

From a purely user-centric point of view, it is appealing to consider RF as a classification by the user of the viewed items from the collection. However, treating relevance feedback as a classification problem does pose significant challenges from a system-centric point of view as pointed out in [12]. Specifically, the *small sample* issue will be encountered, since the user is likely to only mark a few images per iteration, which can pose a problem for learning techniques. Secondly, there is an *asymmetry* in the sense that the retriever is to return a ranked list of the top-k matches; not a binary decision on every single item in the collection. However, it is conceptually appealing to have a general classificatory analysis based framework that could deal with various situations such as varying interpretations of the user need, the best number of classes for feedback, etc as special cases.

For an RF framework to be general, it should focus on RF and be independent of the retrieval mechanism. Since a retriever always returns the same results in response to a query specification and the purpose of RF is to handle user subjectivity, it can be argued that coupling the RF too closely to the retriever goes against the "spirit" of RF. Further, if the RF module is designed and implemented in general terms and loosely coupled with a retrieval engine, it can be used in conjunction with different retrievers and thus provide insight into comparative performance of the retrievers as well as into the nature of the relevance feedback problem itself.

We presented an initial attempt at formulating such a general framework based on Rough Set Theory (RST) [7] in [13]. RST was identified as a suitable basis because of its explicit emphasis on classificatory analysis and knowledge approximation which is well aligned with the motivation for the framework. Further, RST places no constraints on the domain of the attributes in data thus enabling a framework based on it to remain independent of the feature set of a particular collection. Finally, RST has been used in the improvement of other techniques based on hybrid approaches, thus lending itself well to an extensible framework that can be integrated with various retrieval paradigms in a natural and consistent manner [4].

Based on subsequent experimental work and a detailed analysis of the literature, further requirements for such a framework were identified in [14]. The framework has since evolved in order to better meet the requirements and

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

is briefly outlined here to provide adequate context for the subsequent discussion of the result combination method.

2. Overview of the Framework

In the proposed framework, which is referred to as CAFé (Classificatory Analysis based FEedback) the collection is modelled as rough information system as follows.

Consider a homogeneous multimedia collection (i.e a collection in which all items are of the same mode) which is our universe of interest labelled U. The "raw" multimedia data is the set of items, x_i that make up the collection so that $U = \{x_1 \dots x_i \dots x_N\}$ where N is the number of objects in the collection. In addition to the multimedia data, we are interested in meta-data, representing features extracted from the raw data, such as colour, texture, shape, etc.

Let $\mathbf{F}$ be the set of features. For each feature F_i, multiple representations F_{i_j} are possible. A given representation may require more than one component $F_{i_{j_k}}$. The above collection meta-data is to be expressed as a rough set information system. Attributes are atomic, i.e. single valued per object but can take a textual, integral or fractional value. In order to express feature data in the information system the raw data items (images) $x \in U$ as objects and the feature components $F_{i_{j_k}}$ as attributes.

The collection becomes an information system $\mathcal{U} = (U, \mathbf{F})$. Each $F_{i_{j_k}}$ takes a value from the domain $V_{F_{i_{j_k}}}$ for every $x \in U$.

2.1. Capturing the User's Feedback

2.1.1 User's Feedback as Classification

The way relevance feedback is expressed is determined by the capabilities of the UI through which users interact with the system. Since one of the goals of the framework is to be as general as possible, it should be able to cater to existing user interfaces as well as be extensible so that advances in user interface technology can still benefit from it.

Users typically express their feedback in existing systems in one of the following ways:

1. indicating positive examples only (e.g. [3])
2. indicating positive items as well as explicit labelling of negative examples (e.g. [9])
3. classification based on a number of relevance categories (e.g. [11])
4. allocating "goodness" score to each item (also used in [3])
5. group-based labelling of the seen items [6]

In cases 1 – 3, it is trivial to formulate the users feedback as a classification, since all they differ in is the number of classes the seen items are categorised into.

For Case 4, the goodness scores can be converted into a number of classes by discretizing the goodness score interval, thus becoming simplified to an instance of case 3.

Case 5 can be interpreted as a multi-layered classification.

Cases 1 – 3 above can be seen as a special instance of case 5 when there is only one level in the group hierarchy. Thus in all cases, the user's feedback is expressible as a classification of seen items.

2.1.2 Constructing the Decision Systems

We can express a given iteration of feedback as a rough set decision system, or in the general case, as a family of decision systems.

The Simple Case – Single Decision Attribute When the user feedback takes the typical form of a single outcome of classification per item, as is typical (see cases 1 – 3 in Section 2.1.1), the decision system for the i^{th} iteration is constructed as follows.

Let $C = \{l_0 \ \dots l_{n_l}\}$ be the set of labels available to the user. Seen items that are left unmarked can be assigned the label l_0. Then, the definitive ground truth for the collection with respect to the query for this session is the classification U/C that the user would perform based on their own knowledge of their information need. We are then interested in acquiring the best possible approximation of the knowledge contained in X_i/C, where X_i denotes the seen items for iteration i. An attribute, d, is introduced with C as its domain.

Then the set of objects in the decision system $\mathcal{X}_i$ is $X_i \subset U$. The attribute set $\mathbf{F}$ of the rough information system forms the conditional attribute set of the decision system. The attribute d is added to correspond to the outcome of the classification and its domain V_d is the set of possible labels C. In the typical case, C is known in advance due to the inherent capabilities of the user interface hence we do not need to keep track of a changing V_{d_i} across iterations of feedback. So the decision system can be written as $\mathcal{X}_i = (X_i, \mathbf{F} \cup d)$.

The General Case – Groups and a Family of Classifications During the i^{th} iteration, a new attribute d_i^j is introduced for each level j in the hierarchy. The domain of the attribute is the list of the user labels for that level in the hierarchy. Let the hierarchy be n_h levels deep. Each group corresponds to a label, which is the name that user refers to the group by. The j-th level in the hierarchy of groups

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

corresponds to a set of labels $L_j = l_{j_1} \ldots l_{i_{n_{h_j}}}$, where n_{h_j} is the number of groups at level j.

Then, for a completely lossless adaptation of the hierarchy, we create an information system $\mathcal{I}_i$ with its attributes including the feature information as well as the classification information. So $\mathcal{I}_i = (X_i, \mathbf{F}'_i)$ where $\mathbf{F}'_i = \mathbf{F} \cup D_i$. For a full analysis of $\mathcal{I}_i$, a decision system can be constructed for each layer in the hierarchy by treating each d^j as the decision attribute in turn: $\mathcal{X}_i^{d^j} = (X_i, (\mathbf{F}'_i - d_i^j) \cup d_i^j), \forall d_i^j \in D_i$.

2.2. Inferring Users' Information Need

For clarity let us consider the case when there is no group hierarchy and the user categorises each seen item into one of a pre-fixed number of categories C. To approximate U/C in terms of the feature information, a decision system can be constructed as $\mathcal{W} = (U, \mathbf{F} \cup d)$.

At the jth iteration of relevance feedback, the decision system $\mathcal{X}_j = (X_j, \mathbf{F} \cup d)$ is constructed. Since $X_j \subseteq U$, the knowledge contained in $\mathcal{X}_j$ is an approximation of the knowledge contained in $\mathcal{W}$. In CAFé, *reducts* are used as the basis for the approximation of the knowledge regarding the user's classification in in terms of the feature information. Each reduct represents a set of feature components that can classify the seen items as well as the entire feature set in terms of the user feedback. Hence, the set of reducts of $\mathcal{X}_j$ can be used as an approximation of the user need that can be revised when further information becomes available through the next iteration of feedback.

To better interpret the user need, previous iterations also need to be taken into account. This can help to overcome the small sample issue by gradually accumulating a "growing sample." This is done by constructing a cumulative decision system consisting of *all* the seen items, not simply items seen in the current iteration, $\mathcal{X}'_j = \bigcup_j \mathcal{X}_j$. When it is clear which iteration is being considered, the subscript j can be omitted.

We are interested in synthesising a family Θ of attribute sets θ from the $\mathcal{X}_i$s that each θ is likely to have a high degree of overlap with a reduct $\rho \in RED(\mathcal{W})$. We can call these θs "proto-reducts." Θ can be synthesised iteratively by keeping track of those reducts that remain the same or grow across multiple iterations of feedback (see Algorithm 1, where RED_h is used to mean a heuristically computed non-exhaustive set of reducts).

Algorithm 1 Iterative Synthesis of Θ with Frequency Ratio

```
Θ = ∅; i = 1; X' = ∅
while there are further iterations do
  Construct the decision system representing the current iteration X_i = (X_i, F ∪ d)
  Append X_i to X', thus constructing the cumulative decision system
  Compute RED_h(X'_i)
  if Θ = ∅ then
    Initialise Θ to RED_h(X'_i)
  else
    for all ρ ∈ RED_h(X') do
      if ∃θ ∈ Θ such that ρ ⊇ θ then
        Replace θ by ρ in Θ
        num_iterations_θ ++
        freq_ratio_θ = num_iterations_θ / i
      else
        Create a new θ = ρ
        Add θ to Θ
        num_iterations_θ = 1
        freq_ratio_θ = 1 / i
      end if
    end for
  end if
  i = i + 1
end while
```

3. Result Fusion

Each proto-reduct θ in set of proto-reducts Θ_j of the cumulative decision system representing the jth iteration of relevance feedback is either a reduct or a super-reduct of $\mathcal{X}'_j$, and therefore, by definition, a set of attributes that "explains" the user classification of the seen cases and therefore the attribute set represented by each reduct is potentially useful in identifying other matches. Hence, an intuitive way to use the proto-reduct set as the basis of subsequent retrieval simply involves doing a retrieval by each proto-reduct and then combining the results. The expectation is that retrieving by the different proto-reducts may yield different sets, possibly with distinct relevant matches. This is analogous to the rationale that is adopted by research into data fusion and evidence combination in the literature on text retrieval such as [1], [5].

Broadly speaking, the results can be combined either based on combining their similarity (or distance) scores, or based on their ranks. CombMNZ and CombSUM are simple but effective similarity score combination techniques [1] CombSUM is an addition of the similarity scores for a given item from multiple retrievals CombMNZ computes overall similarity as CombSUM multiplied by the number of nonzero similarity scores.

To incorporate rank-based similarity, a similarity measure can be defined in terms of rank and then the standard similarity score techniques can be applied as in [5], which advances the hypothesis that using rank provides better re-

Figure 1. Top 20 Retrieval Results by Euclidean Distance for Query 1

Label	Reduct
ρ_{0_0}	$\{ASM_0, Contr_0\}$
ρ_{0_1}	$\{Contr_0, Corr_0\}$
ρ_{0_2}	$\{ASM_0, Contr_{45}, Corr_{45}\}$
ρ_{0_3}	$\{B, Corr_0, Contr_{45}, Corr_{45}\}$
ρ_{0_4}	$\{Corr_0, ASM_{45}, Contr_{45}, Corr_{45}\}$

Table 1. Reducts computed from Top 20 Matches to Query 1

trieval results when combining two runs that generate very different rank-similarity curves.

3.1. Initial Retrieval

As an illustration, we consider an image collection depicting letters. It consists of 156 images, the images representing the alphabet 6 times in a different colour combinations. The colour index is a three-color histogram in RGB space. For textural features we use the Angular Second Moment (ASM), Measure of Correlation (Corr), Contrast (Contr) and Variance (Var) at $\theta = 0°$ and $\theta = 45°$ with $d = 1$ as in [2].

Assume that the user's criterion for similarity is based on letter rather than by family and that they initiate a session with query image 1. The initial retrieval based on Euclidean distance reveals that that the twenty "closest" images are the ones shown in Figure 1.

The retrieval results in Figure 1 are similar to each other by colour. However, from the user's perspective, only a single image in this result set is relevant – the query image itself. Hence this is the only result considered as having been marked relevant. The other images in the top twenty matches were seen, but not marked relevant. They are considered to belong to another class, which can be called "ignored." This is a degenerate case in terms of the early relevance feedback techniques (e.g. [8, 3]), since there is only one relevant sample and hence using the variance of the feature components across the relevant samples and computing a weighted average query point is not possible.

As in 2.1.1 we can construct the decision system $\mathcal{X}_0 = (X_0, F \cup d)$ corresponding to the users' feedback where the subscript zero implies the "zeroth" iteration of relevance feedback, or the initial retrieval; X_0 is the set of seen images, d is the decision attribute that indicates the label assigned and the set of feature components is $F = \{R, G, B, ASM_0, Contr_0, Var_0, Corr_0, ASM_{45}, Contr_{45}, Var_{45}, Corr_{45}\}$

3.2. Reduct-Based Multiple Retrieval Sessions

The exhaustive reduct set $RED(\mathcal{X}_i)$ of the decision system $\mathcal{X}_0$ computed [1] and shown in Table 1, with the symbol ρ_{i_j} being used to refer to the jth reduct of the ith iteration. Since this is only the beginning of the relevance feedback, we have Θ initialised to $RED(\mathcal{X}_0)$.

It is interesting to note that there is only one reduct that contains a colour feature component (ρ_{0_3} contains the attribute B). This is what we would expect since the users feedback contains the semantic interpretation "even though these images are very similar by colour, only one of them is relevant to the query and different from the others."

Each reduct ρ_{0_j} is fed to the Euclidean distance based retrieval engine separately. In this example, since the collection is small, we can assume that the entire collection is returned as the result set for each reduct, but with a potentially different ordering.

3.3. Result Set Fusion

As mentioned earlier the two most common bases for fusing the results from separate retrievals are the similarity score (in our case distance) and the rank.

Since we retrieve the entire result set in response to a query ordered by Euclidean distance, every single image possesses a degree of similarity to the query i.e. there are no images in any result set corresponding to a reduct that have a zero similarity. CombSUM and CombMNZ as applied to a similarity measure that reflects the distance score are therefore equivalent in this case. The top twenty results by CombSUM fusion of the individual result sets are presented in Figure 2.

To fuse the result sets by rank in an analogous fashion, we use a similarity measure that is defined so that it decreases linearly as rank increases (after [5]):

$$Rank\_Sim(rank) = 1 - \frac{rank - 1}{num\_items\_retrieved}$$

[1] After Entropy MDL discretization of the data

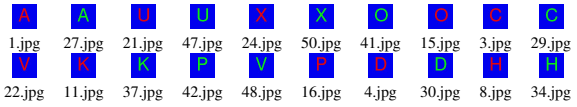

Figure 2. Top 20 Matches After Feedback: CombSUM Distance

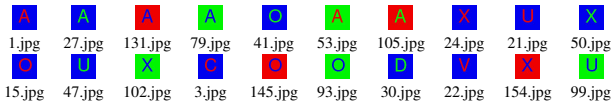

Figure 3. Top 20 Matches After Feedback: CombSUM Rank_Sim

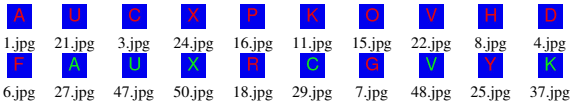

Figure 4. Query 1 top 20 matches after CombSUM distance based on family

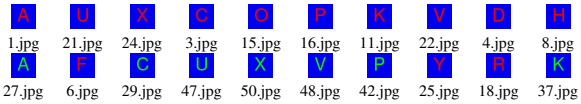

Figure 5. Query 1 top 20 matches by CombSum $rank\_sim$ based on family

Again, for this example, the number of items retrieved is always 156, the size of the collection. The results obtained by using CombSUM on Rank_Sim are presented in Figure 3.

3.4. Analysis

In the example presented there is quite a dramatic difference in the results obtained based on rank and similarity. Specifically, the rank based combination results in all six relevant items being presented within the top seven matches, while the distance-based combination only results in two relevant items being presented in the top twenty of the final result set.

The difference can be explained by a comparative analysis of the the distance-rank curve for each reduct-based retrieval (Figure 6). Each curve is a plot of the distance of each match from the query versus its rank. The curve corresponding to ρ_{0_3} is significantly different from the rest by virtue of the particular step at rank 52. Recall that this is the reduct containing the attribute B and the step can be explained by realising that there are two "families" of 26 images with blue backgrounds – the colour feature components are so dominant in this particular case that the inclusion of one colour component completely biases the distance measure.

While not statistically meaningful in itself as one particular observation, it is in accordance with the hypothesis in [5] that when there is a significant difference in the rank-similarity curves of multiple retrievals, rank may be a better candidate for result combination.

For a contrasting example, let us consider the other valid ground truth in the collection. Assume the user considers all red-on-blue images as representing one family. To simulate this let us use Figure 3 as the initial result set since it contains a number of non-relevant items. Computing the reducts, using them as Θ and combining the result sets by distance and rank_sim yield results as shown in Figures 4 and 5 respectively.

The distance-rank curves are shown in Figure 7. Here the curves are very similar to each other and hence, as might be expected, combining by rank and similarity yield very similar results.

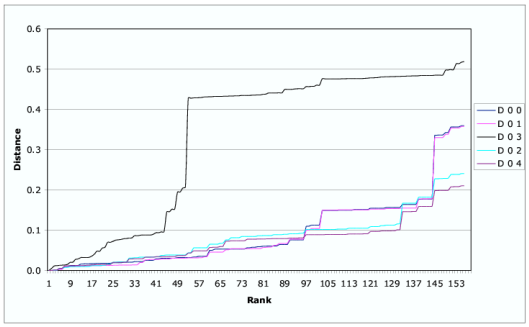

Figure 6. Query 1 Distance-Rank Data for retrievals by each reduct

4. Conclusion

We have presented a result set fusion technique that can be used within used within the overall framework CAFé.

The utility of being able to do so lies in the fact that it is an intuitively appealing and simple mechanism that allows an approximation of the user's information need (the proto-reducts Θ) distilled from feedback to be communicated to a

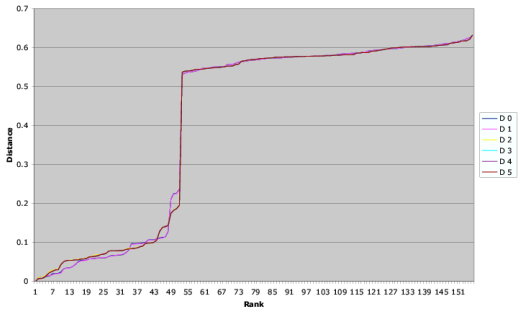

Figure 7. Query 1 Distance-Rank Data for retrievals by each θ, family

retrieval engine while treating the retriever as a black box. The example shown was based on a simple Euclidean distance based retriever, but theoretically it can be used with any retriever that can return a list of matches ranked in order of computed relevance based on some distance or similarity measure and that allows retrieval by a subset of the entire feature set. Hence the method remains independent of the retrieval engine.

The theory and the example results presented show that there is potential to be able to deal with very small samples while treating the feedback as a classification. We have attempted to do so while focussing the analysis on the classification using RST methods.

Hence the result fusion method (and CAFé) remains independent of the retriever and also to be able to deal with a small sample, as was deemed desirable in section 1.

However, attempting to achieve these objectives does come at a significant computational cost. Computing reducts exhaustively is an NP-hard problem as is attempting to find an optimal family of cuts for discretization. Effective heuristics for computing semi-optimal cuts and genetic algorithms that can produce "sufficiently many" reducts have been proposed and found to be acceptable [4].

Future work is to be done related to result fusion to allow a semi-automatic select of fusion technique (e.g. based on analysing the distance-rank curves) and possibly regarding strengthening the confidence in the specific proto-reducts that result in better retrieval based on feedback in successive iterations. Further evaluation is also required to establish the extent to which the proposed result set fusion technique can improve retrieval performance.

References

[1] E. Fox and J. Shaw. Combination of multiple searches. In *Proceeding of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technoology Special Publication*, pages 243–252, 1994.

[2] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(6):610–621, 1973.

[3] Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Querying databases through multiple examples. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 218–227, 1998.

[4] J. Komorowski, L. Polkowski, and A. Skowron. Rough sets: a tutorial. In S. Pal and A. Skowron, editors, *Rough-Fuzzy Hybridization: A New Method for Decision Making*, pages 3–98. Springer-Verlag, Singapore, 1998.

[5] J. H. Lee. Analyses of multiple evidence combination. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–276, New York, NY, USA, 1997. ACM Press.

[6] M. Nakazato, L. Manola, and T. Huang. Imagegrouper: a group-oriented user interface for content-based image retrieval and digital image arrangement. *Journal of Visual Languages & Computing*, 14(4):363–386, August 2003.

[7] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11:341–356, 1982.

[8] Y. Rui, T. Huang, and S. Mehrotra. Content-Based image retrieval with relevance feedback in MARS. In *Proceedings of the IEEE International Conference on Image Processing*, pages 815–818, 1997.

[9] K. Tieu and P. Viola. Boosting image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[10] C. Wilson, B. Srinivasan, and M. Indrawan. A General Inference Network Based Architecture for Multimedia Retrieval. In *Proceedings of IEEE International Conference on Multimedia (ICME2000)*, pages 347–350, New York City, USA, 2000.

[11] H. Wu, H. Lu, and S. Ma. Willhunter: Interactive image retrieval with multilevel relevance measurement. In *17th International Conference on Pattern Recognition (ICPR'04)*, volume 2, pages 1009–1012, 2004.

[12] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. In *ACM Multimedia Systems Journal Special Issue on CBIR*, volume 8, pages 536–544, 2003.

[13] S. Zutshi, C. Wilson, S. Krishnaswamy, and B. Srinivasan. Modelling Relevance Feedback Using Rough Sets. In *Proceedings of the 5th International Conference on Advances in Pattern Recognition (ICAPR)*, pages 495–500, Kolkata, India, 2003.

[14] S. Zutshi, C. Wilson, S. Krishnaswamy, and B. Srinivasan. The role of relevance feedback in managing multimedia semantics: A survey. In *Managing Multimedia Semantics*, pages 288–304. Idea Group Inc., 2005.

0-7695-2841-4/07 $25.00