1

2

3

Preface

I wish to express my gratitude to the instructor of my thesis, Seppo Ilkka, for his guidance and careful reviewing of the manuscript. To professor Juhani Pitkäranta I am grateful for kindly acting as the supervisor of my thesis. Professor Shosta Sulonen deserves special thanks for offering me the opportunity to work in this interesting research project and providing me with a pleasant working environment with state-of-the-art office accessories and equipment. Particular thanks are due to Yrjö Leino for mathematical discussions and for reviewing the manuscript. I also wish to thank Mikko Laine for inspiring discussions concerning this research. It has been a great pleasure to work within the SmartPush project team: Teppo, Sami, Antti, Andreas and Virpi.

This thesis is a part of SmartPush research project at TAI Research Centre of Helsinki University of Technology. I wish to acknowledge our sponsors, the National Technology Agency TEKES, Alma Media, WSOY, Sonera, ICL, Nokia Research Centre and TeamWARE Group.

Last but not least, I want to thank my spouse, Timo, for his love, patience and encouragement.

Espoo, 15th November, 1999

Eerika Savia

Contents

1 Introduction **1**

2 Background **2**
2.1 Information Filtering 2
2.2 Vector Space Model and Latent Semantic Indexing 3

3 Representation Model **4**
3.1 Metadata 4
3.2 Hierarchy 5
3.3 Learning and Dimensions of Metadata 7

4 Measuring Distance between Profile and Document **8**
4.1 Coarse Distance Measure 9
4.2 Fine Distance Measure 11
4.3 Combining Distance Measures of Various Levels of Hierarchy . 14
4.4 Negative Weights and Hierarchy 15
4.5 Combining Coarse and Fine Measure 19
4.5.1 Method for Combining Coarse and Fine Measure . . . 19
4.5.2 Alternative Method 20
4.6 Threshold Parameter ε for Ignoring Insignificant Weights in Profiles 21

5 Feedback and Learning of User Profiles **23**
5.1 Learning Method 23
5.1.1 Learning Parameter λ 24
5.1.2 Forgetting Parameter f 24
5.2 Learning and Decreasing of Distance 25
5.2.1 Learning Decreases Coarse Distance Measure 25
5.2.2 Problems with Fine Measure 30
5.2.3 When Does Combined Measure Decrease? 31

6 Matching and Learning in Other Dimensions of Metadata **33**
6.1 *Priority* of a Document 33
6.1.1 Learning of *Priority* 33
6.1.2 Matching of *Priority* 35
6.1.3 Negative Feedback and Negative Profile for *Priority* . . 36
6.1.4 Compatibility of Learning and Matching for *Priority* . 37

6.2 Other Flat Metadata Structures 41
6.2.1 Learning and Matching in these Dimensions 41

7 Social Information Filtering and Clustering of User Profiles 42
7.1 Social Filtering 42
7.1.1 Social Filtering with Content Based Profiles 42
7.2 Clustering Profiles 44
7.2.1 Maximin-Distance Clustering Method 45

8 Measuring Distance between Two User Profiles 46
8.1 Different Measures for Different Purposes 46
8.2 Similarity Measures for Social Filtering 46
8.2.1 Discussion about Objective of Similarity Measure . . . 46
8.2.2 Cosine Measure 48
8.2.3 Distance Measure for Social Filtering with Content Based Profiles 50
8.2.4 Reliability of Distance 50
8.3 Distance Measure Candidates for Clustering of User Profiles . 52
8.3.1 Requirements for Distance Measures for Clustering of User Profiles 52
8.3.2 Measures Based on the Angle between Vectors 56
8.3.3 Measures that Allow Negative Weights 63

9 Testing of Methods 64
9.1 Tests with Clustering User Profiles 64
9.2 Planned Further Testing 67
9.2.1 Available Data and Limitations 67
9.2.2 Designed Performance Scores 68
9.2.3 Combining Scores of Various Sessions and of Various Users 72
9.2.4 Learning Parameters and Amount of Training Data . . 72
9.2.5 Tests with Users 72

10 Open Issues and Future Work 73
10.1 Comparisons with Other Methodologies 73
10.2 Automatic Learning of Metadata from Data 73

11 Conclusions 75

A Appendix **76**
A.1 Compatibility of Learning Method and Distance Measures . . 76
A.1.1 Negative Feedback Increases Coarse Measure 76
A.1.2 Convergence of Learning Method 79
A.1.3 Compatibility of Learning Method and Symmetric Cosine Based Measure 80
A.2 Proofs of Triangle Inequality for Various Measures 82
A.3 Bayesian Learning in Information Filtering 84
A.4 Summary of Suggested Distance Measures 87

References **89**

8

1 Introduction

Traditional publishing and delivery of documents in printed form is very restricted in volume, so careful selection on what is published must be carried out by the editors. With electronic documents it is financially feasible to offer a much wider variety of information and a freedom of choice to the customers. When the editor no longer acts as a strict filter for the documents selecting of relevant documents is left to the user.

It is widely recognized that automated filtering is required to manage the overwhelming flow of new electronic information people face today. On the other hand, electronic delivery presents a great opportunity for publishers to offer personalized content to the customers.

In SmartPush project at Helsinki University of Technology a prototype of a personalized information filtering service is developed and implemented. This thesis is a study of the mathematical methods involved in this prototype.

2 Background

2.1 Information Filtering

The history of *information filtering* (IF) has its roots in *information retrieval* (IR). The IR techniques were relatively mature as early as in the 1960's [1], [2]. Although IF and IR have a lot in common there are some distinctive differences between them [3]. In a typical information retrieval situation there is a more or less specific information need, which is expressed explicitly to the system, e.g. as a query, and results are expected immediately. In traditional IR the document collection is stable and the queries change all the time whereas in IF the information need is rather stable and it is matched against a continuous flow of new documents.

To filter documents according to user interests we must have a common representation for both the users and the documents [4]. This knowledge representation model must be expressive enough to describe different media formats like video and audio as well as material in several languages.

A *metadata* representation of a document is a compact representation of the document content and can be used to make filtering decisions without access to the original document. This allows the system to operate only on the metadata instead of the whole content.

A *profile* is a representation of a user's interests in a knowledge representation model. The purpose of the user profile is to filter and rank documents in decreasing order of interest. The user profile should be adaptable to changes in the user's interests. Adaptation can be based on feedback the user gives for the documents. A user profile may also represent dislike for or avoidance of certain subjects. This is referred to as a *negative profile*.

To be able to make decisions about how well a document matches a profile, we must have a way to measure how similar or different they are. By a *distance measure* we mean a measure between a profile and a document that produces a nonnegative real number reflecting the amount by which they differ from each other. In general, it does not have to satisfy the requirements of a metric as defined in mathematics. These distance measures will be discussed in chapter 4.

Furthermore, a learning method must be designed to assemble the user profiles and to be responsible for the adaptation of the profiles to changes in the user interests. Our learning method will be discussed in chapter 5.

Situations where comparisons between different user profiles are required, such as *social filtering* and *clustering* of profiles, are also discussed (chapters 7 and 8).

2.2 Vector Space Model and Latent Semantic Indexing

The most popular information filtering model for textual material is the so called vector space model [1],[5],[6],[7],[8]. In this model, both documents and profiles are represented as vectors with components for different *terms*. Terms are words or word stems extracted from the text, with a set of most commonly appearing words removed. The weight for a given term k in document i is $w_{ik} = tf_{ik} \cdot idf_k$, where the term frequency tf_{ik} expresses the frequency of a given term k in a given document i and the inverse document frequency idf_k expresses the unusualness of a given term k in the whole document collection. Naturally, the vector space model can be applied to texts in only one language at a time.

A commonly used similarity measure for these vectors is the *cosine measure* introduced in section 8.2.2. For the cosine measure it is assumed that the basis, that consists of the terms, is the natural basis of $\mathbb{R}^n$, n being the number of terms. Thus, the terms should be independent and they should form an orthonormal basis for the document vector space. Since the terms are definitely not independent, the orthogonality assumption is troublesome. A method called Latent Semantic Indexing (LSI), which is based on the singular value decomposition (SVD), has been developed to solve the problem of orthogonality of the vector components [9], [10], [11].

LSI can be applied to any fixed document collection that is represented with term vectors. By decomposing the matrix that consists of the document vectors, one gets an orthonormal basis for that document space, and if the calculations are done according to that basis, the components are indeed orthogonal. One problem of this method is the intensive calculation needed to compute the SVD and, of course, finding a good enough collection of documents to represent all the future documents as well. Moreover, LSI does not solve the problem of expressing texts in different languages and formats other than text.

3 Representation Model

3.1 Metadata

We assume that all documents have a metadata representation regardless of whether it has been created manually or automatically. Constructing a profile is based solely on the metadata information of those documents the user has indicated some interest in. The methods introduced in this thesis are built on the assumption that both document content and user interests are represented in a metadata model.

There can be several aspects of metadata that are relevant in judging whether the user is interested in the document. These different aspects are called the *dimensions* of metadata. In our prototype there are 5 distinct dimensions. Two of them are hierarchical, *Subject* describing the content of the document and *Location* that refers to the geographical location. Two of the dimensions, *Industry* and *Company*, are Boolean vectors. They indicate a document relating to certain industries or companies. In addition, we have a dimension called *Priority*, which reflects the editorial view on the importance of the document. It is integer valued and will be discussed in more detail in section 6.1.

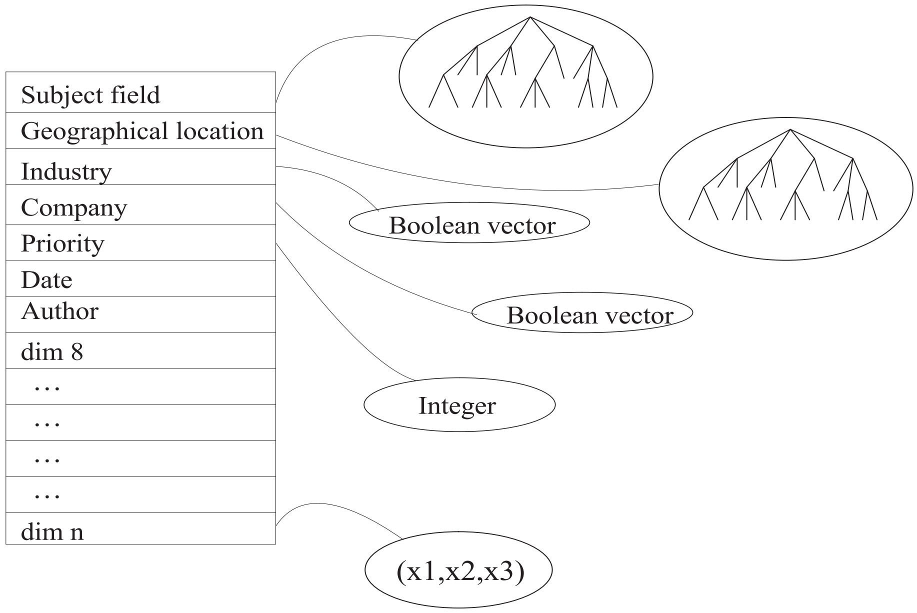

Figure 1: User profiles consist of several aspects (dimensions) of metadata. Some of them can be hierarchical, others for example Boolean values or vectors of real numbers.

3.2 Hierarchy

With the content description model (*Subject*) we would like to be able to represent information in various degrees of detail. More importantly, from detailed user information it should be possible to draw some general conclusions about the user's interests. It could also be useful to be able to extend the assumed interest to the neighboring concepts. Our suggestion for the content description model is a generalization hierarchy that groups related concepts together [1],[12],[13]. A hierarchy as such codifies information about the dependencies of different concepts. For a concrete example of a concept hierarchy, see table 3 on page 64.

In our model the document classification is a normalized discrete weight distribution over the leaf level concepts. The weight expresses the relative importance of the concept as classifying information for the document. In other words, the document representations are real valued vectors $\mathbf{d}$ with $\forall\ i \quad d_i \geq 0$ and the sum of the weights is $\sum d_i = 1$. The weights for the inner nodes of the tree are just summarized from the leaf level vector.

The concepts on one level can be seen as fuzzy sets, each of which has a membership function that expresses to which extent each document is a member of that set [14]. The advantage of this is that it has more expressive power than just crisp classification of documents into classes. Obviously, the same applies also for the vector space model.

The profiles are learned from the documents, hence they are also real valued vectors $\mathbf{x}$ on the leaf level. However, the weights of the profile can grow without a limit and they can also be negative signifying dislike of the subject. The profile is divided into positive and negative parts and they are normalized separately for the distance calculations.

The upper levels of the hierarchy simply gather related subjects together and the weight of an upper node is always the sum of the weights of its child nodes. The distance calculations can be performed separately on each level of the hierarchy and the results of different levels can be combined, as will be suggested in section 4.3.

It is desirable that the concepts at each level are approximately of the same degree of detail independent of the branch. In the distance calculations they are interpreted in that manner anyhow.

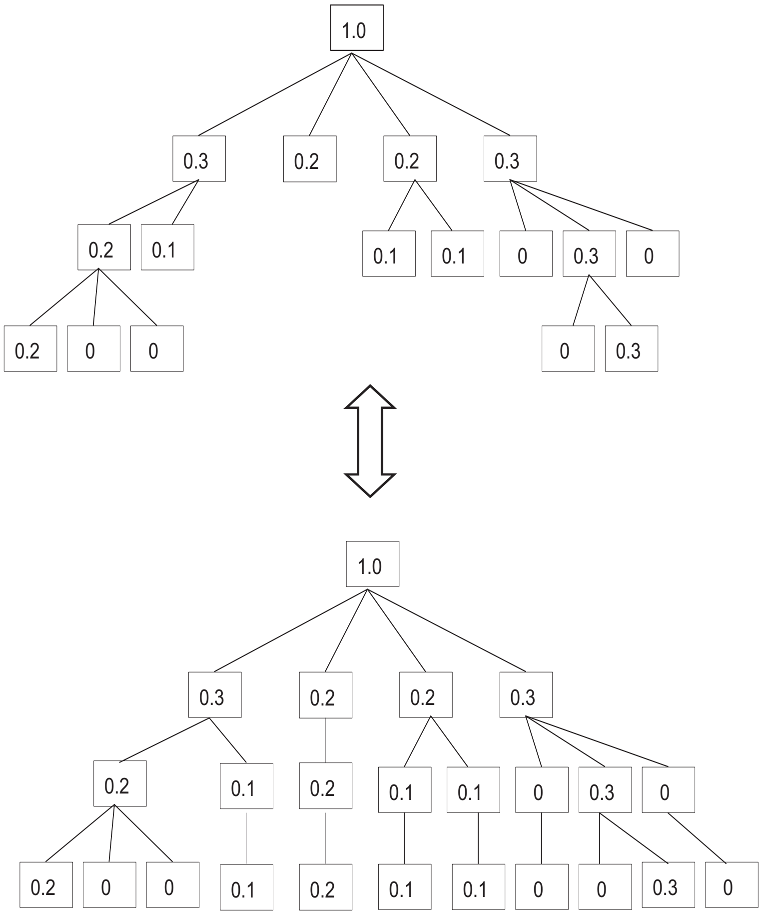

Figure 2: *Subject* is a hierarchical dimension of metadata, and the weights are summed upwards in the hierarchy. The branches that are shorter are in practice continued to the same level as the deepest branch.

Furthermore, the topmost level can efficiently reduce the amount of required calculations, since matching is needed only for those profile–document pairs that have a lot in common. We need to consider only the relevant parts of the hierarchy instead of the whole vector with lots of zero entries. This approach scales well as the number of documents and profiles grow.

In addition, a hierarchical structure is helpful in the metadata creation process since for a human being a hierarchy is an intuitive way to navigate a multitude of concepts.

3.3 Learning and Dimensions of Metadata

We can keep track of what kind of documents a certain user usually wants to see and build the profile based on that information. If the user provides feedback on the documents the task is made even easier. But we must bear in mind that the only things the profile knows is what it has been told. It cannot anticipate any changes in the user's interests before they are somehow apparent in the feedback or user's behavior. Therefore, a profile is really helpful only when we talk about long term interests that do not change abruptly but rather evolve gradually. Of course, one could tune the system to be so sensitive that the first hint to some new interest would have a dramatic effect on the profile. But then, the profile would be totally different every time the user activated it, and it would hardly gain any trust from the user since it would seem to be offering different documents every time.

To be able to learn a profile we need a good representative set of features that describe the documents. The profile is learned in terms of these features. The simplest example could be a keyword system where the features are the presence or absence of a keyword in the document. If the document has some kind of a representative, it could include a lot of information about, e.g., what geographical areas the document involves, who is the author, whether it is a news article or a column etc. All this information can be used to determine the profile but the problem is that there is no way to discriminate the relevant features from the irrelevant ones. Which of the features determine whether a new document is of interest, can depend on the user. This can even be case dependent for a single user: one article is interesting to the user because the subject is important to him and another article is interesting because the user happens to like the author.

Learning in terms of various feature sets would require information on why a certain document was interesting instead of just a list of interesting documents. It is viable to provide the user with a feedback mechanism that allows him or her to specify the relevant features while giving feedback. What is not feasible is to expect the user to specify the reason of his feedback. It may even be questionable whether we can expect the user to give any feedback explicitly.

We shall first examine the distance measures and learning method for hierarchical dimensions of metadata in chapters 4 and 5, and then continue with the other dimensions in chapter 6.

4 Measuring Distance between Profile and Document

It is crucial to note that the roles of the profile and the document are not symmetric [15], [16]. A profile contains information of various interests the user has and is by no means the prototype of a perfect document. So, rather than trying to find a document that mimics the profile in slightest detail we are searching for good partial matches to the profile.

Example:

Let us examine documents *doc*1 and *doc*2 in table 1 shown in figure 3 as well. Both documents intersect with the profile only in the topic of Mathematics and neither one of them matches the other interests the user has. But there is a difference between these two cases: *doc*2 is related only to the topics of the profile while *doc*1 involves Economics, which is not present in the profile at all. Document *doc*2 does not involve topics outside the profile, so it can be seen as a perfect match to a part of the user profile. We can conclude that the white area in figure 3 is not as significant as the striped area when evaluating distance.

Concept	**User profile**	**doc1**	**doc2**
Mathematics	0.2	0.4	1.0
Railway timetables	0.3		
Apartment sales advertisements	0.5		
Economics		0.6	

Table 1: Example of partial matching.

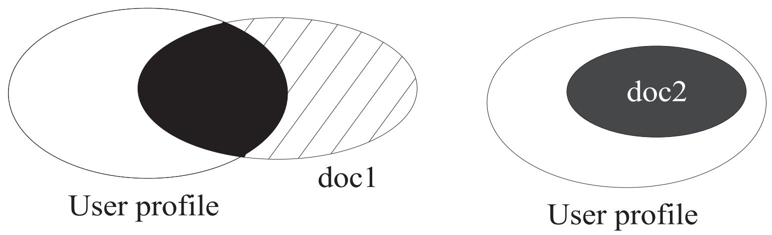

Figure 3: Both documents match with a part of the user profile. Document doc1 has the drawback that it involves subject areas outside the profile as well (the striped area). Document doc2 does not involve any subject areas outside the profile's interests.

We apply the distance measures separately at each level of hierarchy, so a user profile is a real valued vector $\mathbf{x}$. If only positive profiles are allowed, the components of $\mathbf{x}$ are nonnegative. For comparisons with documents we normalize the profile vector $\mathbf{x}$ so that the sum of positive weights is 1 and the sum of negative weights is -1. In other words, the positive part of the profile and the negative part of the profile are both normalized according to the 1-norm[1]. We denote the normalized user profile by $\mathbf{u}$. The document is also a real valued vector $\mathbf{d}$ at each level of hierarchy, but with the restriction to nonnegative weights and with $\sum_i d_i = 1$.

Combining the distances from various levels is discussed in section 4.3.

4.1 Coarse Distance Measure

We shall first introduce a measure that is designed to be concerned with those weights in the document that do not hit the positive profile (the striped area in figures 4 and 5 on the next page). A kind of coarse ordering of documents according to their potential relevance to the profile can be made by the *coarse measure* introduced in [15]

$$D_c(\mathbf{u}, \mathbf{d}) = \sum_{u_i=0} d_i \qquad \in [0, 1]. \tag{1}$$

This has to be somewhat modified if the user profile is allowed to contain both positive and negative weights:

$$D_c(\mathbf{u}, \mathbf{d}) = \sum_{u_i\leq 0} d_i \qquad \in [0, 1]. \tag{2}$$

Although (2) allows negative weights it ignores them by treating them as zeroes (see figure 5). In order to make a difference between a zero weight and a negative weight, a document matching a negative weight could be penalized in the distance measure, e.g., as follows

$$D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{2}\left(\sum_{u_i\leq 0} d_i + \sum_{u_i<0} d_i\right) = \frac{1}{2}\left(\sum_{u_i=0} d_i + 2\sum_{u_i<0} d_i\right) \qquad \in [0, 1]. \tag{3}$$

In the distance measure above those document weights that match the negative profile are penalized twice as much as those weights that just fail to match the positive profile. Naturally it can be weighted by any weighting scheme desired ($p, q \geq 0$ and $q \geq p$):

$$D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{q}\left(p\sum_{u_i=0} d_i + q\sum_{u_i<0} d_i\right) \qquad \in [0, 1]. \tag{4}$$

[1] $||d||_1 = \sum_i |d_i|$

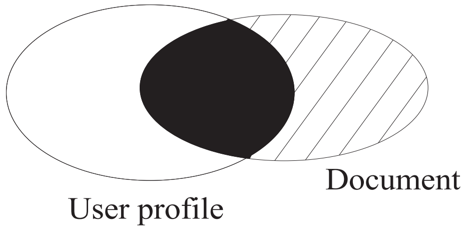

Figure 4: The roles of the profile and the document are not symmetric. It does not matter that the document does not match all interests in the profile (the white area) but it does matter that there are subject areas in the document not covered by the profile (the striped area). The coarse measure deals with the striped area while the fine measure concentrates on the black area in detail.

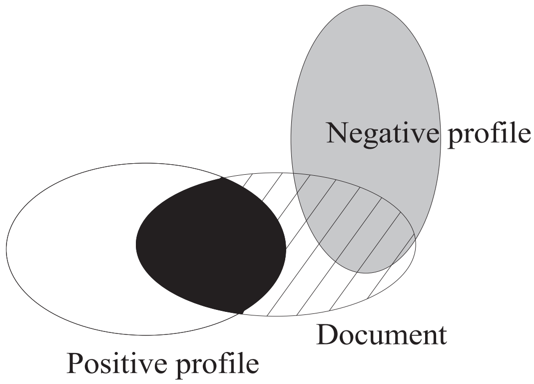

Figure 5: It does not matter that the document does not match all interests in the positive profile (the white area) but it does matter that there are subject areas in the document not covered by the profile (the striped area). Furthermore, the situation is worse if the document matches the negative profile (the grey area) than if it just fails to hit the positive profile.

4.2 Fine Distance Measure

So far we have just estimated whether the document matches the profile's areas of interest on the whole, without considering how important those areas of interest are for the profile. When two documents get equal distances using the coarse measure, we would still like to arrange them according to the extent they hit the weights of the profile using a *fine distance measure*. The next example justifies the need for a fine measure.

Example:

> Let us take an extreme example, where the user profile **u** is as in figure 6 and the documents we consider are **d1** in figure 7 and **d2** in figure 8. We find that the coarse measure treats these documents as equally well matching documents disregarding the fact that the first document **d1** has a hit in much larger weight than the second document **d2**.

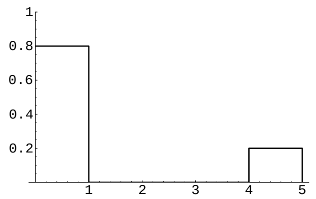

Figure 6: User profile **u**.

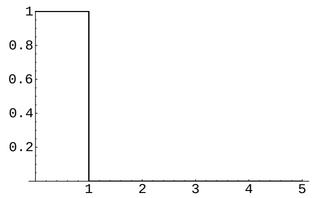

Figure 7: Document **d1**.

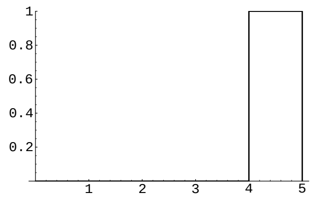

Figure 8: Document **d2**.

Our first candidate for such a fine measure, introduced in [15], is

$$D_{f1}(\mathbf{u}, \mathbf{d}) = 1 - \sum_{i \in I} u_i \cdot d_i \ , \tag{5}$$

where $\mathbf{u}$ is a normalized user profile vector and $I = \{i \mid u_i > 0 \wedge d_i > 0\}$. Note that if we only have positive profiles the index set includes all the indices of vector $\mathbf{u}$. The measure is symmetric but that is not a problem because it only deals with the intersecting part of the profile and the document (the black area in figures 4 and 5). We can safely assume that the profile $\mathbf{u}$ and the document $\mathbf{d}$ have at least one node in common; otherwise we can just set the fine measure to its maximum, $D_f(\mathbf{u}, \mathbf{d}) = 1$.

Another possibility is a cosine based measure, restricted to the nodes, at which the profile and the document match. We denote the index set of common nodes by $I = \{i \mid u_i > 0 \ \wedge \ d_i > 0\}$.

$$Dcos_f(\mathbf{u}, \mathbf{d}) = 1 - \frac{\sum_I u_i \cdot d_i}{\sqrt{\sum_I u_i^2} \cdot \sqrt{\sum_I d_i^2}} \ . \tag{6}$$

The following example points out some weaknesses of both the above mentioned measure candidates, (5) and (6).

Example:

Let us compare two simple user profiles **u1** and **u2** to one document **doc**, with $\mathbf{u1} = (0, 0.5, 0.5, 0, 0)$, $\mathbf{u2} = (0.9, 0.05, 0.05, 0, 0)$ and $\mathbf{doc} = (0, 0.4, 0.4, 0, 0.2)$ (See figures 9, 10 and 11). The coarse distance of both profiles is equal to 0.2. The fine measure candidate $D_{f1}(\mathbf{u}, \mathbf{d})$ in equation (5) gives

$$D_{f1}(\mathbf{u1}, \mathbf{doc}) = 0.6 \quad \text{and} \quad D_{f1}(\mathbf{u2}, \mathbf{doc}) = 0.96 \ ,$$

which is slightly unfair for the profile **u1**, since it already has optimal distribution of weights for the document *doc*, given that it has those two weights in common with the document. The restricted cosine measure $Dcos_f(\mathbf{u}, \mathbf{d})$ gives

$$Dcos_f(\mathbf{u1}, \mathbf{doc}) = 0 \quad \text{and} \quad Dcos_f(\mathbf{u2}, \mathbf{doc}) = 0 \ ,$$

which makes no sense in the case of profile **u2**.

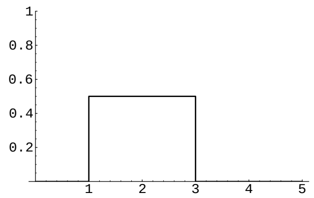

Figure 9: Profile **u1**.

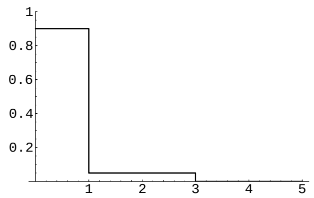

Figure 10: Profile **u2**.

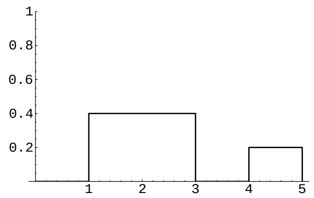

Figure 11: Document **doc**.

We still introduce another candidate for the fine measure

$$D_{f2}(\mathbf{u},\mathbf{d}) = 1 - \frac{\sum_{i\in I} u_i \cdot d_i}{\max_{i\in I} d_i}\ , \tag{7}$$

where $I = \{i \mid u_i > 0\ \wedge\ d_i > 0\}$. Note that

$$\sum_{i\in I} u_i \cdot d_i \le (\max_{i\in I} d_i) \sum_{i\in I} u_i \le \max_{i\in I} d_i\ .$$

If we compare this measure with the previous example (see page 13), we get $\max_{i\in I} \mathbf{doc}_i = 0.4$ and the distances

$$D_{f2}(\mathbf{u1},\mathbf{doc}) = 0 \quad \text{and} \quad D_{f2}(\mathbf{u2},\mathbf{doc}) = 0.9\ ,$$

which both seem to make sense.

4.3 Combining Distance Measures of Various Levels of Hierarchy

Once we have calculated the coarse and fine distance measures at each level of the representation hierarchy, it is time to combine the information into one coarse distance and one fine distance. Let our hierarchy have N levels (root node excluded). The calculations are done at each level separately and the results are then combined in some convenient manner. For positive profiles the weights can be accumulated upwards in the hierarchy and the profile vector is well defined at each level of hierarchy. Let us denote the profile vector at level m by $\mathbf{u}_m$ and the document vector at the same level by $\mathbf{d}_m$. The distances at various levels of hierarchy can be summarized into one distance in the following way

$$\mathcal{D}_{\#}(\mathbf{u},\mathbf{d}) = \sum_{m=1}^{N} w_m D_{\#}(\mathbf{u}_m,\mathbf{d}_m)\ , \tag{8}$$

where $\sum_{m=1}^{N} w_m = 1$ and $\forall\, m\ w_m \ge 0$ and the subscript $\#$ denotes either the coarse measure (c) or the fine measure (f).

In our tests we have used mainly the following weighting schemes

$$w_m = 2^{m-N-1} \quad m = 2,\ldots,N, \qquad w_1 = 2^{1-N} \tag{9}$$

in order to emphasize the leaf level and

$$w_m = 2^{-m} \quad m = 1,\ldots,N-1, \qquad w_N = 2^{1-N} \tag{10}$$

when emphasizing the uppermost levels.

Alternatively, one could just take the maximum value among levels

$$\mathcal{D}_{\#}(\mathbf{u}, \mathbf{d}) = \max_{m} D_{\#}(\mathbf{u}_m, \mathbf{d}_m) = D_{\#}(\mathbf{u}_N, \mathbf{d}_N) \ ,$$

where N denotes the leaf level of the hierarchy, but this would, of course, make the hierarchical structure absolutely useless.

When one has a more general comparison in mind the uppermost levels should be taken as the most significant while ignoring little differences at leaf level. This could be the case if similar profiles are searched. When ranking the documents in decreasing order of interest we want the leaf level differences to count too, and the leaf level end should be emphasized. In order to reduce workload it is of course possible to pick just a few separate levels into calculations. From the point of view of distance calculations this would be equivalent to removal of those nodes that are not used to calculate the distance. There might still be some ground to keep those nodes in the model to ease the navigation in the hierarchy during the metadata creation process.

Different measures for positive profiles are compared and discussed in detail in [15].

4.4 Negative Weights and Hierarchy

In the case of positive profiles the weights could be summed upwards in the hierarchy to get a representation in terms of less detailed concepts. When we allow negative weights in the leaf level nodes we must redefine the meaning of the upper nodes. If we simply add the positive and negative weights in a straightforward manner, we just diminish the information gathered.

The following example introduces an extreme case where the profile would just disappear if the positive and negative weights were added. The result would be a totally unknown profile, which contradicts the actual leaf level profile.

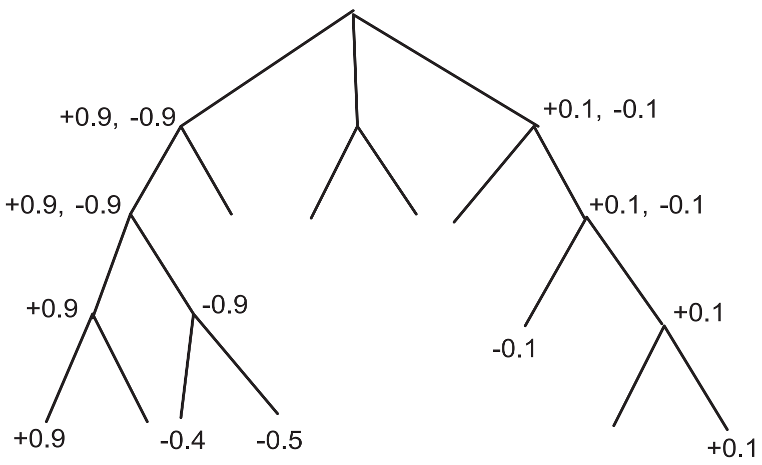

Figure 12: Example profile y with both positive and negative weights.

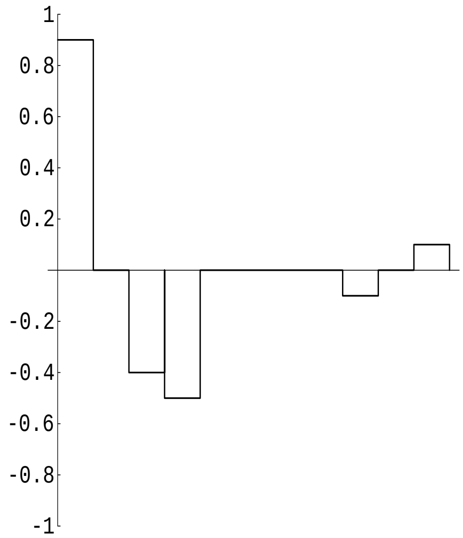

Figure 13: Profile y at leaf level.

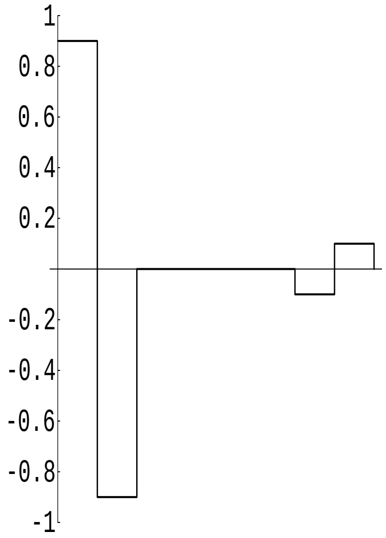

Figure 14: Profile y at the level next to the leaf level.

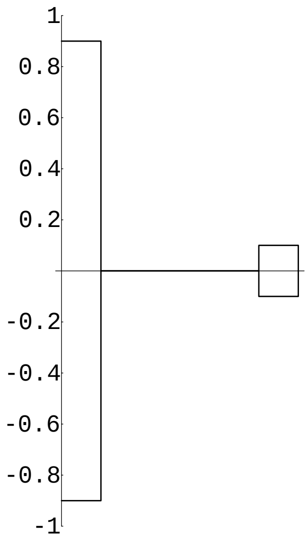

Figure 15: Profile y at the second level from above.

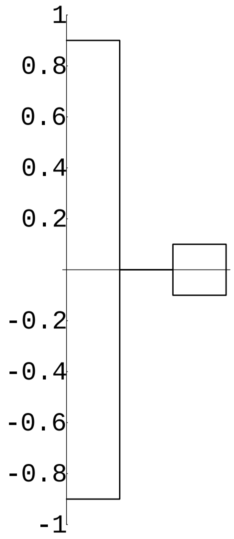

Figure 16: Profile y at the uppermost level.

A rational conclusion is that the weights are no longer well defined in those nodes where positive and negative weights overlap and we should consider those levels where positive and negative profiles do not yet overlap. Let there be N levels, at k of which the positive and negative profiles are separate. We should still have the same weighting of levels as we do for all other situations, e.g.,

$$w_m = 2^{m-N-1} \quad m = 2, \ldots, N, \qquad w_1 = 2^{1-N}. \tag{11}$$

For levels $1, \ldots, N-k$ the distance is undefined and for levels $N-k+1, \ldots, N$ the weights should be exactly as in (11). The best approximation we have for distance at upper levels is the one at level $N-k+1$, so let us use it. In practice this means that the weight at level $N-k+1$ is the sum of all the upper weights. So, the weights for the levels will be

$$w_m = 2^{m-N-1} \quad m = N-k+2, \ldots, N, \qquad w_{N-k+1} = 2^{1-k}, \tag{12}$$

and only the well defined distances are taken into summation over the levels

$$\mathcal{D}_{\#}(\mathbf{u}, \mathbf{d}) = \sum_{m=N-k+1}^{N} w_m D_{\#}(\mathbf{u}_m, \mathbf{d}_m). \tag{13}$$

4.5 Combining Coarse and Fine Measure

In principle, we use the coarse measure discussed in section 4.2 to determine to which extent the document fails to match the profile. The fine measure discussed in section 4.2 is then used to make a distinction between matching of small weights and matching of larger weights. A combination of these two should give an overall measure.

4.5.1 Method for Combining Coarse and Fine Measure

The method for combining the coarse measure and the fine measure introduced in [15] and [16] ensures that the coarse measure roughly determines the ordering of documents and the fine measure only separates those documents that are equally good matches according to the coarse measure. The idea is to partition the range of coarse measure and divide the documents into n equivalence classes accordingly. Then the fine measure is used only to determine the ordering of documents inside these equivalence classes.

$$D(\mathbf{u},\mathbf{d}) = \begin{cases} \frac{1}{n}\lfloor n\cdot\mathcal{D}_c(\mathbf{u},\mathbf{d})\rfloor + \frac{1}{n}\cdot\mathcal{D}_f(\mathbf{u},\mathbf{d}) \quad , & \text{if } \mathcal{D}_c(\mathbf{u},\mathbf{d}) < 1 \\ 1 \quad , & \text{if } \mathcal{D}_c(\mathbf{u},\mathbf{d}) = 1. \end{cases} \tag{14}$$

Here $\lfloor\cdot\rfloor$ denotes the floor function.

When the partition is refined the effect of the fine measure gradually vanishes

$$\lim_{n\to\infty}\left(\frac{1}{n}\lfloor n\cdot\mathcal{D}_c\rfloor + \frac{1}{n}\cdot\mathcal{D}_f\right) = \lim_{n\to\infty}\frac{1}{n}\lfloor n\cdot\mathcal{D}_c\rfloor .$$

Denote the decimal part of a real number by $[\cdot]$. Then $n\cdot\mathcal{D}_c = \lfloor n\cdot\mathcal{D}_c\rfloor + [n\cdot\mathcal{D}_c]$ and the limit above is

$$\lim_{n\to\infty}\frac{\lfloor n\cdot\mathcal{D}_c\rfloor}{n} = \lim_{n\to\infty}\frac{n\cdot\mathcal{D}_c - [n\cdot\mathcal{D}_c]}{n} = \lim_{n\to\infty}\left(\mathcal{D}_c - \frac{[n\cdot\mathcal{D}_c]}{n}\right) = \mathcal{D}_c\ . \tag{15}$$

Thus,

$$\lim_{n\to\infty} D(\mathbf{u},\mathbf{d}) = \mathcal{D}_c(\mathbf{u},\mathbf{d}). \tag{16}$$

At the other end of the scale, when $n = 1$ we have

$$D(\mathbf{u},\mathbf{d}) = \mathcal{D}_f(\mathbf{u},\mathbf{d}).$$

So, we can choose the desired ratio of the effects of the coarse and the fine measure by adjusting the parameter n.

4.5.2 Alternative Method

Let us introduce an alternative, much more appealing method for combining the coarse and the fine measure

$$D(\mathbf{u},\mathbf{d}) = p \cdot \mathcal{D}_c(\mathbf{u},\mathbf{d}) + (1-p) \cdot \mathcal{D}_f(\mathbf{u},\mathbf{d}), \tag{17}$$

where $p \in [0,1]$.

Its disadvantage is that it can only be used provided that inequality (18) holds

$$\mathcal{D}_f(\mathbf{u},\mathbf{d}) \geq \mathcal{D}_c(\mathbf{u},\mathbf{d}). \tag{18}$$

When (18) holds we can write $\mathcal{D}_f(\mathbf{u},\mathbf{d}) = \mathcal{D}_c(\mathbf{u},\mathbf{d}) + g(\mathbf{u},\mathbf{d})$, where $g(\mathbf{u},\mathbf{d}) \geq 0$.

$$\begin{aligned} D(\mathbf{u},\mathbf{d}) &= p \cdot \mathcal{D}_c(\mathbf{u},\mathbf{d}) + (1-p) \cdot (\mathcal{D}_c(\mathbf{u},\mathbf{d}) + g(\mathbf{u},\mathbf{d})) \\ &= \mathcal{D}_c(\mathbf{u},\mathbf{d}) + (1-p) \cdot g(\mathbf{u},\mathbf{d}) \\ &\geq \mathcal{D}_c(\mathbf{u},\mathbf{d}). \end{aligned}$$

For the condition $\mathcal{D}_f(\mathbf{u},\mathbf{d}) \geq \mathcal{D}_c(\mathbf{u},\mathbf{d})$ to hold for the measure combined from various levels of hierarchy, $\mathcal{D}_f$, it must hold for the measure at each level, D_f.

The condition $D_f(\mathbf{u},\mathbf{d}) \geq D_c(\mathbf{u},\mathbf{d})$ only holds for one of the discussed fine measures combined with the coarse measure (4), namely

$$D_{f1}(\mathbf{u},\mathbf{d}) = 1 - \sum_{u_i>0} u_i \cdot d_i \ .$$

The proof is a straightforward calculation

$$\begin{aligned} D_{f1}(\mathbf{u},\mathbf{d}) &= 1 - \sum_{u_i>0} u_i \cdot d_i \\ &\geq 1 - \max_{d_i>0} u_i \sum_{u_i>0} d_i \geq 1 - \sum_{u_i>0} d_i \\ &= \sum_{u_i=0} d_i + \sum_{u_i<0} d_i = D_c(\mathbf{u},\mathbf{d}) + \left(1 - \frac{p}{q}\right) \sum_{u_i=0} d_i \\ &\geq D_c(\mathbf{u},\mathbf{d}) \ . \end{aligned}$$

4.6 Threshold Parameter ε for Ignoring Insignificant Weights in Profiles

As time goes on the profile might get small weights in almost all subjects of interest , which may mislead the distance measure. It is intended that a nonzero weight in a profile would imply at least some degree of interest of the user to that subject. Most prominently it affects the coarse measure (equations (2),(3),(4)) where all positive weights are considered meaningful. There, a threshold could be used to ignore too small weights in the profile

$$D_c(\mathbf{u},\mathbf{d}) = \sum_{u_i<\varepsilon} d_i \qquad \in [0,1] \tag{19}$$

or, with the negative profile taken into account ($q \geq p$ and $p, q > 0$)

$$D_c(\mathbf{u},\mathbf{d}) = \frac{1}{q}\left(p \sum_{u_i \in]-\varepsilon,\varepsilon[} d_i + q \sum_{u_i \leq -\varepsilon} d_i\right) \qquad \in [0,1] \tag{20}$$

The appropriate level of threshold ε varies surely between different metadata structures. One important parameter to take into account when choosing ε is the number of nodes on that level of hierarchy. Even though insignificant weights are ignored in distance calculation they should not simply be drawn to zero in the profile but preserved as they are. Otherwise the growth of those unfortunate weights would be prevented too efficiently. Questions to be discussed are whether the threshold should be set for the normalized or unnormalized profile and, whether the threshold would apply to the leaf level and let the effect accumulate to upper levels of hierarchy or to every level of hierarchy separately. It becomes very complicated to apply thresholds level by level. The following example compares applying the threshold at the leaf level only to applying it level by level.

<u>Example:</u>

> Let the profile be as in figure 17. In figure 18 the ε-threshold is applied at the leaf level only and in figure 19 the threshold is applied at each level of the hierarchy separately. As we can see, applying the threshold level by level is quite complicated and the weights do not sum upwards in the hierarchy as they should.

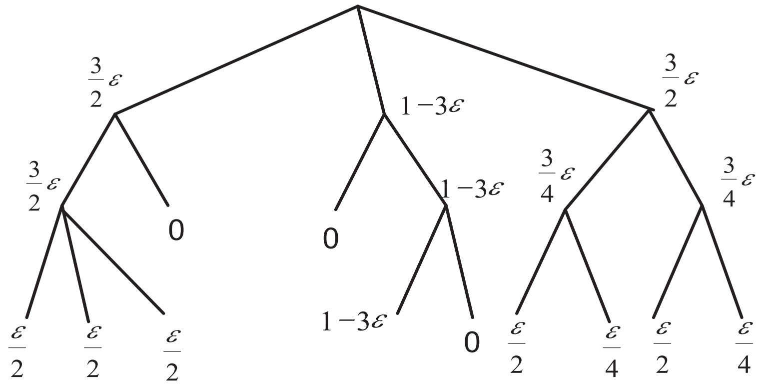

Figure 17: Example profile z.

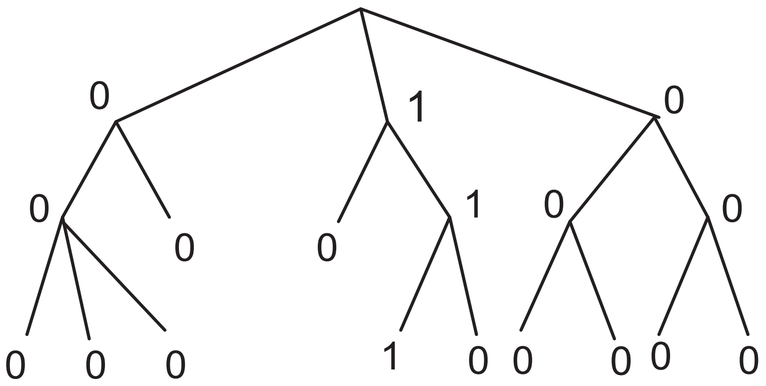

Figure 18: If ε-threshold is applied at the leaf level only and the effect is accumulated upwards in the hierarchy, profile z (in figure 17) will be as above.

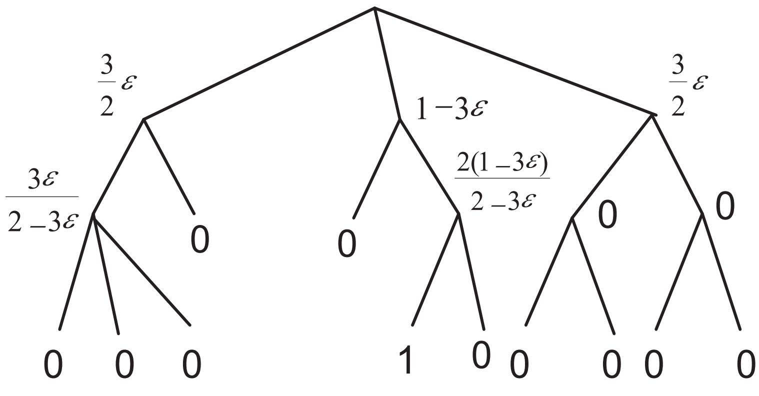

Figure 19: If ε-threshold is applied at each level separately and the normalization is done at each level independently, profile z (figure 17) results in the above. Note that the weights no longer sum upwards in the hierarchy.

5 Feedback and Learning of User Profiles

In this chapter we discuss the method used in learning the user profiles from interaction with the users.

The objective of the learning method is to learn each user's interests and represent them with a user profile, and besides that, to gradually forget the obsolete parts of the profile [8]. Users are provided with mechanisms for giving both positive and negative feedback on the documents and their feedback is used as an input of the learning method. This kind of active judging of documents by users is called *explicit feedback*.

The profile can also be learned from *implicit feedback*, which can be, e.g., knowledge of the documents the user opens, or some other information that can be observed from a user's behavior. A significant disadvantage of this approach is that it confuses positive and negative feedback since we do not know whether the user liked the documents he opened.

5.1 Learning Method

As earlier, we concentrate on one level of hierarchy at a time. The profile vector without normalization is denoted by $\mathbf{x}$, and the document vector is denoted by $\mathbf{d}$. Our learning method, introduced in [17], consists of two components: the learning of new interests and the forgetting of the past interest. If we denote the profile after feedback with the superscript * the two components are

$$\text{learning:} \qquad \mathbf{x}^* = \mathbf{x} + \lambda\mathbf{d} \tag{21}$$

$$\text{forgetting:} \qquad \mathbf{x}^* = \frac{\mathbf{x}}{f}\,. \tag{22}$$

We have introduced two ways to integrate the forgetting of the old profile to the learning, when new (explicit or implicit) user feedback comes along. The profile can be forgotten either each time feedback is given, or automatically with certain time intervals [17]. If the former profile is forgotten each time a user gives feedback with amplifier λ to some document $\mathbf{d}$, we use the following equation for learning

$$\mathbf{x}^* = \frac{\mathbf{x}}{f} + \lambda\mathbf{d}. \tag{23}$$

If, on the other hand, forgetting is supposed to happen periodically with time, we can use equation (24) for learning.

$$\mathbf{x}^{k+1} = \frac{\mathbf{x}^k}{f^{T_{k+1}-T_k}} + \lambda_{k+1}\mathbf{d}^{k+1} \qquad \text{where} \quad T_k = \left\lfloor \frac{t_k}{B} \right\rfloor \tag{24}$$

For this method a time interval B has to be chosen. After every time interval B the forgetting is applied to the profile. Since pure forgetting (22) does not affect the normalized profile $\mathbf{u}$ used in the distance calculations we do not need the result of forgetting before the next learning. We need to update forgetting only when applying learning, since the normalized profile used in distance calculations does not change in pure forgetting.

5.1.1 Learning Parameter λ

The learning parameter λ indicates the magnitude and direction (positive or negative) of the given feedback. For positive feedback $\lambda > 0$ and for negative feedback $\lambda < 0$. Note that if λ were zero no learning would occur. For simplicity, we use an integer valued λ ($\lambda \in \mathbb{Z}\backslash\{0\}$) that has some fixed range $\{-M, \ldots, -1\} \cup \{1, \ldots M\}$. We interpret implicit feedback as learning with a smaller learning coefficient, so the smallest feedback coefficients in absolute value, $\lambda = 1$ and $\lambda = -1$, are reserved for implicit feedback. Values $\lambda \geq 2$ then correspond to explicit positive feedback and values $\lambda \leq -2$ to explicit negative feedback.

Since we infer implicit feedback directly from the documents the user opens, we can, in fact, erroneously consider both positive and negative implicit feedback as positive. So, actually, the coefficient for negative implicit feedback $\lambda = -1$ is never needed, unless we find some way to implicitly infer negative feedback.

5.1.2 Forgetting Parameter f

Apparently the appropriate magnitude of the forgetting parameter f can vary a lot among different users. It is not feasible to assume that a user could set the parameter value for himself either. One way to facilitate adaptation to changing user needs is to carry along two to three profiles that forget in different paces.

Let us study a case where there were three different forgetting paces f_{slow}, f_{medium} and f_{fast}. The slowest evolving profile would hardly change in a few months time while the fastest evolving profile could follow the trends of the week. If, for example, the medium profile and the fast profile agreed with each other and differed on the other hand substantially from the slow profile, the change in interests could be interpreted rather permanent and the slow profile could be updated to be equal with the medium one.

5.2 Learning and Decreasing of Distance

It is important that the methods for learning user profiles and matching between documents and profiles are compatible. In practice this means that positive feedback for a document should decrease the distance between the document being judged and the profile considered. Negative feedback should correspondingly increase the distance.

5.2.1 Learning Decreases Coarse Distance Measure

In this section we conclude that the learning method (23) decreases the coarse distance between the profile and the document used in learning. There are two cases to be considered separately: the simple case, where any nonzero weight is regarded as a hit weight when calculating the coarse measure (4) and the version with an ε-threshold (20).

Proof for the simple case:

After feedback the resulting profile (unnormalized) is written component-wise [17]:

$$x_i^* = \frac{x_i}{f} + \lambda d_i \ ,$$

where the forgetting parameter $f > 1$ and the parameter $\lambda > 0$ for positive feedback and $\lambda < 0$ for negative feedback (generally $\lambda \in \mathbb{Z}\backslash\{0\}$). We now restrict our focus on positive feedback only, but the increase of the coarse distance for negative feedback can be concluded with similar arguments. The proof for the negative case can be found in appendix A.1.1.

Let us divide the indices into three distinct sets

$$I_p = \{i \mid x_i > 0\} = \{i \mid u_i > 0\}$$

$$I_0 = \{i \mid x_i = 0\} = \{i \mid u_i > 0\}$$

$$I_q = \{i \mid x_i < 0\} = \{i \mid u_i > 0\}.$$

Similarly, we denote the corresponding sets for the resulting profile $\mathbf{x}^*$ by I_p^*, I_0^* and I_q^*. As earlier, we denote the normalized profile by $\mathbf{u}$.

It is easy to see that in the case of positive feedback the following is true:

$$
\begin{aligned}
i \in I_p \quad &\Rightarrow \quad x_i^* = \frac{x_i}{f} + \lambda d_i > 0 && \Rightarrow \quad i \in I_p^* \\
i \in I_0 \quad &\Rightarrow \quad x_i^* = \lambda d_i \geq 0 && \Rightarrow \quad i \in \left(I_p^* \cup I_0^*\right) \\
i \in I_q \quad &\Rightarrow \quad x_i^* < \lambda d_i \leq \lambda && \Rightarrow \quad i \in \left(I_p^* \cup I_0^* \cup I_q^*\right) \ .
\end{aligned}
$$

Let us use the most general coarse measure

$$
D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{q}\left(p \sum_{I_0} d_i + q \sum_{I_q} d_i\right) \tag{25}
$$

in the proof. The other cases: coarse measure without negative weights (eq. (2)) corresponding to $p = q = 1$ and coarse measure with negative weights (eq. (3)) where $p = 1$ and $q = 2$, will be proved at the same time.

Before feedback the coarse distance measure between a profile $\mathbf{u}$ and a document $\mathbf{d}$ is $D_c(\mathbf{u}, \mathbf{d})$ as defined in equation (25). After the feedback the coarse distance measure between the new profile $\mathbf{u}^*$ and document $\mathbf{d}$ is

$$
\begin{aligned}
D_c(\mathbf{u}^*, \mathbf{d}) &= \frac{1}{q}\left(p \sum_{I_0^*} d_i + q \sum_{I_q^*} d_i\right) \\
&= \frac{1}{q}\left(p \sum_{I_0^* \cap I_0} d_i + p \sum_{I_0^* \cap I_q} d_i + q \sum_{I_q^*} d_i\right) \\
&= \frac{p}{q} \sum_{I_0^* \cap I_q} d_i + \sum_{I_q^*} d_i \ ,
\end{aligned} \tag{26}
$$

where the fact $\quad i \in I_0 \cap I_0^* \quad \Rightarrow \quad d_i = 0 \quad$ has been used.

By dividing the indices of I_q into three distinct sets in the following way

$$
\sum_{I_q} d_i = \sum_{I_q^*} d_i + \sum_{I_q \cap I_0^*} d_i + \sum_{I_q \cap I_p^*} d_i
$$

we get

$$
\sum_{I_q \cap I_0^*} d_i = \sum_{I_q} d_i - \sum_{I_q^*} d_i - \sum_{I_q \cap I_p^*} d_i. \tag{27}
$$

Next, substitute equation (27) into equation (26) (and notice that $\sum_{I_q^*} d_i \leq \sum_{I_q} d_i$) to get

$$
\begin{aligned}
D_c(\mathbf{u}^*, \mathbf{d}) &= \frac{p}{q}\left(\sum_{I_q} d_i - \sum_{I_q^*} d_i - \sum_{I_q \cap I_p^*} d_i\right) + \sum_{I_q^*} d_i \\
&\leq \frac{p}{q}\left(\sum_{I_q} d_i - \sum_{I_q*} d_i\right) + \sum_{I_q^*} d_i = \frac{p}{q}\sum_{I_q} d_i + \left(1 - \frac{p}{q}\right)\sum_{I_q*} d_i \\
&\leq \frac{p}{q}\sum_{I_q} d_i + \left(1 - \frac{p}{q}\right)\sum_{I_q} d_i \\
&= \sum_{I_q} d_i. \qquad (28)
\end{aligned}
$$

On the other hand,

$$
D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{q}\left(q\sum_{I_q} d_i + p\sum_{I_0} d_i\right) \geq \sum_{I_q} d_i. \qquad (29)
$$

It follows from equations (28) and (29) immediately that

$$
D_c(\mathbf{u}^*, \mathbf{d}) \leq D_c(\mathbf{u}, \mathbf{d}). \quad \square
$$

Proof for the case with an ε-threshold:

In this case only those nonzero weights whose absolute value exceeds certain ε are regarded as hit weights when calculating the distance between the user profile and a document. We can also assume guaranteed that the weights in the documents are always either greater than ε or exactly zero.

Again, we can restrict our focus on positive feedback only and leave the proof for negative feedback to the appendix A.1.1. The indices are divided into three distinct sets

$$\begin{aligned}
I_p =& \{i \mid x_i \geq \varepsilon\} \\
I_0 =& \{i \mid -\varepsilon < x_i < \varepsilon\} \\
I_q =& \{i \mid x_i \leq -\varepsilon\}.
\end{aligned}$$

Similarly, denote the corresponding sets for the resulting profile $\mathbf{x}^*$ by I_p^*, I_0^* and I_q^*. In the case of positive feedback ($\lambda \geq 1$) the following holds:

$$\begin{aligned}
(i \in I_p) \wedge (d_i \geq \varepsilon) \quad & \Rightarrow \quad x_i^* = \frac{x_i}{f} + \lambda d_i \geq \frac{\varepsilon}{f} + \lambda\varepsilon \geq \left(1 + \frac{1}{f}\right)\varepsilon \geq \varepsilon \\
& \Rightarrow \quad i \in I_p^* \\
(i \in I_p) \wedge (d_i = 0) \quad & \Rightarrow \quad x_i^* = \frac{x_i}{f} \geq \frac{\varepsilon}{f} > 0 > -\varepsilon \\
& \Rightarrow \quad i \in \left(I_p^* \cup I_0^*\right) \\
i \in I_0 \quad & \Rightarrow \quad x_i^* > -\frac{\varepsilon}{f} + \lambda d_i \geq -\frac{\varepsilon}{f} \geq -\varepsilon \\
& \Rightarrow \quad i \in \left(I_p^* \cup I_0^*\right) \\
i \in I_q \quad & \Rightarrow \quad x_i^* \leq -\frac{\varepsilon}{f} + \lambda d_i \leq \lambda - \frac{\varepsilon}{f} \\
& \Rightarrow \quad i \in \left(I_p^* \cup I_0^* \cup I_q^*\right) \ .
\end{aligned}$$

Before feedback the coarse distance measure between the normalized profile $\mathbf{u}$ and document $\mathbf{d}$ is

$$D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{q}\left(p \sum_{I_0} d_i + q \sum_{I_q} d_i\right) . \tag{30}$$

By dividing the indices further into smaller distinct sets, we get the equations (31) and (32) :

$$\sum_{I_q} d_i = \sum_{I_q^*} d_i + \sum_{I_0^* \cap I_q} d_i + \sum_{I_p^* \cap I_q} d_i \tag{31}$$

$$\sum_{I_0} d_i = \sum_{I_0 \cap I_0^*} d_i + \sum_{I_0 \cap I_p^*} d_i. \tag{32}$$

Adding up equations (31) and (32) and using $i \in (I_0^* \cap I_p) \implies d_i = 0$ we get

$$\begin{aligned}
\sum_{I_0} d_i + \sum_{I_q} d_i &= \sum_{I_q^*} d_i + \sum_{i \in I_p^*, i \notin I_p} d_i + \left(\sum_{I_0^* \cap I_q} d_i + \sum_{I_0^* \cap I_0} d_i \right) \\
&= \sum_{I_q^*} d_i + \sum_{i \in I_p^*, i \notin I_p} d_i + \left(\sum_{I_0^*} d_i - \sum_{I_0^* \cap I_p} d_i \right) \\
&= \sum_{I_q^*} d_i + \sum_{i \in I_p^*, i \notin I_p} d_i + \sum_{I_0^*} d_i \\
&\geq \sum_{I_0^*} d_i + \sum_{I_q^*} d_i, \qquad (33)
\end{aligned}$$

and it is easy to see that

$$\sum_{I_q} d_i \geq \sum_{I_q^*} d_i. \tag{34}$$

Finally, using inequalities (33) and (34) we get

$$\begin{aligned}
D_c(\mathbf{u}^*, \mathbf{d}) &= \frac{p}{q} \left(\sum_{I_0^*} d_i + \sum_{I_q^*} d_i \right) + \left(1 - \frac{p}{q}\right) \sum_{I_q^*} d_i \\
&\leq \frac{p}{q} \left(\sum_{I_0} d_i + \sum_{I_q} d_i \right) + \left(1 - \frac{p}{q}\right) \sum_{I_q^*} d_i \\
&\leq \frac{p}{q} \left(\sum_{I_0} d_i + \sum_{I_q} d_i \right) + \left(1 - \frac{p}{q}\right) \sum_{I_q} d_i \\
&= \frac{1}{q} \left(p \sum_{I_0} d_i + q \sum_{I_q} d_i \right) \\
&= D_c(\mathbf{u}, \mathbf{d}). \quad \square
\end{aligned}$$

5.2.2 Problems with Fine Measure

In section 4.2 three candidates for the fine measure $D_f(\mathbf{u}, \mathbf{d})$ were introduced. None of them, however, is such that the decrease of distance with positive feedback is guaranteed. The situation is as complicated for negative feedback and increasing of distance.

The problem originates from the different objectives of the learning method and the distance measure. The goal of the learning method is to keep track of all the interests the user has. The distance measure, on the other hand, looks for partial matches to the profile, not especially for an overall match. Therefore, the best possible document $\mathbf{d}$ for a profile $\mathbf{u}$ according to the distance measure is not $\mathbf{d} = \mathbf{u}$.

If we applied the positive learning method for a profile $\mathbf{u}$ with the same document $\mathbf{d}$ infinitely many times, the profile would finally converge to $\mathbf{u} = \mathbf{d}$. So, in order to get the distance decrease definitely for positive feedback one should choose a distance measure that equals to zero for $\mathbf{u} = \mathbf{d}$ and not otherwise.

The convergence of $\mathbf{u}$ to $\mathbf{d}$ is quite obvious. Using equation (23), after N iterations of learning method the profile is

$$\begin{aligned} \mathbf{x}_N &= \frac{1}{f^N}\mathbf{x}_0 + \lambda \left(\sum_{k=0}^{N-1} \frac{1}{f^k} \right) \mathbf{d} \qquad && \text{with forgetting } (f > 1) \text{ , and} \\ \mathbf{x}_N &= \mathbf{x}_0 + N\lambda\mathbf{d} && \text{without forgetting } (f = 1) \text{ ,} \end{aligned}$$

where $\mathbf{x}_0$ is the initial profile and λ is the learning parameter. With forgetting, and letting N to infinity we get

$$\lim_{N\to\infty} \mathbf{x}_N = \left(\lim_{N\to\infty} \frac{1}{f^N} \right) \mathbf{x}_0 + \lambda \frac{f}{f-1} \mathbf{d} = \lambda \frac{f}{f-1} \mathbf{d} \ .$$

So, either the positive or the negative profile vanishes. For positive feedback ($\lambda > 0$) the resulting normalized profile is $\mathbf{u} = \mathbf{d}$, and for negative feedback ($\lambda < 0$) the profile is $\mathbf{u} = -\mathbf{d}$. If no forgetting is applied, the profile must be normalized before letting N to infinity. (For details, see appendix A.1.2)

Actually, it can be shown that the following measure $D_f(\mathbf{x}, \mathbf{d})$ decreases with positive feedback and increases with negative feedback

$$D_f(\mathbf{x}, \mathbf{d}) = \frac{1}{2}\left(1 - \frac{\sum_{i\in I} x_i \cdot d_i}{\sqrt{\sum_{i\in I} x_i^2}\sqrt{\sum_{i\in I} d_i^2}}\right) = \frac{1}{2}\left(1 - \cos(\mathbf{x}, \mathbf{d})\right) . \qquad (35)$$

Here the index set I consists of all the indices. The profile $\mathbf{x}$ must be the unnormalized one, so that it includes both positive and negative components. This measure would, unfortunately, be completely inappropriate as a distance measure used to find documents for the user profiles, above all because of its symmetry. The proof is nevertheless included in appendix A.1.3.

5.2.3 When Does Combined Measure Decrease?

Let us discuss the problem with both methods of combining the coarse and the fine measure. The first combination method introduced in section 4.5.1 was

$$D(\mathbf{u}, \mathbf{d}) = \begin{cases} \frac{1}{n}\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor + \frac{1}{n} \cdot \mathcal{D}_f(\mathbf{u}, \mathbf{d}) \quad , & \text{if } \mathcal{D}_c(\mathbf{u}, \mathbf{d}) < 1 \\ 1 \quad , & \text{if } \mathcal{D}_c(\mathbf{u}, \mathbf{d}) = 1 \ , \end{cases}$$

where $\lfloor\cdot\rfloor$ denotes the floor function. Let us denote the profile after feedback with the superscript *. If the combined distance $D(\mathbf{u}, \mathbf{d})$ is already equal to 1, it cannot increase.

If $\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor$ decreases, it decreases by at least 1, i.e.,

$$\lfloor n \cdot \mathcal{D}_c(\mathbf{u}^*, \mathbf{d})\rfloor \le \lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor - 1 .$$

From this it immediately follows that the combined distance D($\mathbf{u}$,$\mathbf{d}$) decreases as well:

$$\begin{aligned} D(\mathbf{u}^*, \mathbf{d}) &= \frac{1}{n}\lfloor n \cdot \mathcal{D}_c(\mathbf{u}^*, \mathbf{d})\rfloor + \frac{1}{n} \cdot \mathcal{D}_f(\mathbf{u}^*, \mathbf{d}) \\ &\le \frac{1}{n}\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor - \frac{1}{n} + \frac{1}{n} \cdot \mathcal{D}_f(\mathbf{u}^*, \mathbf{d}) \\ &\le \frac{1}{n}\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor \le \frac{1}{n}\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d})\rfloor + \frac{1}{n} \cdot \mathcal{D}_f(\mathbf{u}, \mathbf{d}) \\ &= D(\mathbf{u}, \mathbf{d}) . \end{aligned}$$

So, we conclude that if the coarse measure decreases by at least $\frac{1}{n}$, the combined measure definitely decreases.

Usually, slightly less is required as is shown below. Let us denote

$$n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d}) = a + \delta ,$$

where $a \in \mathbb{N}$ and $\delta \in [0, 1)$. Then

$$\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d}) \rfloor = a .$$

For $\lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d}) \rfloor$ to decrease, the condition

$$n \cdot \mathcal{D}_c(\mathbf{u}^*, \mathbf{d}) < a$$

is required. So, what is actually needed is

$$\mathcal{D}_c(\mathbf{u}, \mathbf{d}) - \mathcal{D}_c(\mathbf{u}^*, \mathbf{d}) > \frac{1}{n} (a + \delta - a) = \frac{\delta}{n} . \tag{36}$$

If the inequality (36) does not hold, $\lfloor n \cdot \mathcal{D}_c(\mathbf{u}^*, \mathbf{d}) \rfloor = \lfloor n \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d}) \rfloor$, and the increase or decrease of the combined distance measure is determined solely by the fine measure.

As explained in the previous section 5.2.2 the fine measure candidates can also increase in some cases. The maximal increase of the combined measure is

$$\begin{aligned} &\frac{1}{n} (\mathcal{D}_f(\mathbf{u}^*, \mathbf{d}) - \mathcal{D}_f(\mathbf{u}, \mathbf{d})) \\ &\leq \frac{1}{n} . \end{aligned}$$

So, with this method of combining the measures the effect of the problematic fine measure can be bounded to $\frac{1}{n}$.

We now continue with the alternative method of combining the measures introduced in section 4.5.2

$$D(\mathbf{u}, \mathbf{d}) = p \cdot \mathcal{D}_c(\mathbf{u}, \mathbf{d}) + (1 - p) \cdot \mathcal{D}_f(\mathbf{u}, \mathbf{d}) .$$

The worst case would be when $\mathcal{D}_f(\mathbf{u}, \mathbf{d}) < \mathcal{D}_f(\mathbf{d}, \mathbf{d})$ and the coarse measure remains the same, i.e., $\mathcal{D}_c(\mathbf{u}^*, \mathbf{d}) = \mathcal{D}_c(\mathbf{u}, \mathbf{d})$. Then, the change in the combined distance would be

$$\begin{aligned} D(\mathbf{u}^*, \mathbf{d}) - D(\mathbf{u}, \mathbf{d}) &= (1 - p) (\mathcal{D}_f(\mathbf{u}^*, \mathbf{d}) - \mathcal{D}_f(\mathbf{u}, \mathbf{d})) \\ &\leq (1 - p) (\mathcal{D}_f(\mathbf{d}, \mathbf{d}) - \mathcal{D}_f(\mathbf{u}, \mathbf{d})) \\ &\leq (1 - p) \mathcal{D}_f(\mathbf{d}, \mathbf{d}) \\ &\leq 1 - p . \end{aligned}$$

Note, however, that the extreme value $1 - p$ cannot be achieved, since for it to hold we need $\mathcal{D}_f(\mathbf{u}, \mathbf{d}) = 0$, which only holds when both profile $\mathbf{u}$ and document $\mathbf{d}$ have all their weights in one node, the same node for both of them. If this is the case, the normalized profile $\mathbf{u}$ is not affected by the application of learning method with this document, and the change in the combined distance is zero.

6 Matching and Learning in Other Dimensions of Metadata

In the previous chapters we have focused on matching and learning in a hierarchical dimension of metadata. Examples of such dimensions are the description of content (*Subject*) and geographical location (*Location*). Next, we shall introduce learning methods and distance measures for other types of metadata dimensions.

6.1 *Priority* of a Document

Some sort of priority label (denoted *doc* in this chapter) could be attached to documents indicating the urgency of the document or the newsworthiness judged by the journalist. This is, of course, a subjective judgement and could in theory be totally contradictory to a particular user's point of view. But for some users, it could be the most relevant aspect in regarding which documents should be recommended.

Let us have a priority label called *Priority* with $2M + 1$ possible discrete values. For practical purposes it is obvious that M cannot be very large, but it is surely less than 5. The range does not really matter, but it is useful to have an odd number of values, so that the middle one can be interpreted semantically as neutral. For convenience the range is next selected to be

$$doc \in \{-M, -(M-1), \ldots, -1, 0, 1, \ldots, M-1, M\}.$$

6.1.1 Learning of *Priority*

Let us first focus on positive feedback only and denote the user profile for *Priority* by x_+ . The simplest way to capture user's preferences according to *Priority* is to take an average of the *Priority* labels of those documents judged positively by the user so far:

$$x_+ = \frac{\sum_{i=1}^{N} doc_i}{N}. \tag{37}$$

To enable the calculation of the next value of profile x_+ recursively, both previous x_+ and N must be kept in the profile representation:

$$x_+^* = \frac{N \cdot x_+ + doc_{N+1}}{N+1}. \tag{38}$$

This way all the documents are considered equally important. If we allow the user to give various amounts of feedback for different documents, the priorities must be weighted accordingly. Let us denote the learning parameter for the ith document by λ_i. Furthermore, we shall tentatively write $W_N = \sum_{i=1}^{N} \lambda_i$. We can choose the profile to be the weighted average of the *Priority* labels of those documents judged by the user so far:

$$x_+ = \frac{\sum_{i=1}^{N} \lambda_i \cdot doc_i}{W_N}. \tag{39}$$

Again, the next value of the profile can be calculated recursively as long as we keep the information x_+ and W_N:

$$x_+^* = \frac{W_N \cdot x_+ + \lambda_{N+1} \cdot doc_{N+1}}{W_{N+1}}\ , \tag{40}$$

where $W_{N+1} = \sum_{i=1}^{N+1} \lambda_i = W_N + \lambda_{N+1}$.

Since we are gradually forgetting the earlier feedback in the *Subject* dimension as time passes, it would be consistent to do so in other dimensions, too. Following the same logic as in (23) the forgetting parameter $f \geq 1$ can be appended in (40) resulting in

$$x_+^* = \frac{\frac{W_N \cdot x_+}{f} + \lambda_{N+1} \cdot doc_{N+1}}{\frac{W_N}{f} + \lambda_{N+1}}\quad . \tag{41}$$

Here the effects of forgetting must be taken into account in W_N:

$$\begin{cases} W_1 & = \lambda_1 \\ W_{N+1} & = \frac{W_N}{f} + \lambda_{N+1} \end{cases} \tag{42}$$

or, equivalently

$$W_N = \sum_{i=1}^{N} \left(\frac{1}{f}\right)^{N-i} \cdot \lambda_i. \tag{43}$$

It is easy to see that assigning $f = 1$ in (41) yields (40)

$$x_+^* = \frac{\left(\sum_{i=1}^{N} \lambda_i\right) x_+ + \lambda_{N+1} \cdot doc_{N+1}}{\sum_{i=1}^{N+1} \lambda_i}$$

and further setting all feedback coefficients equal ($\forall i \quad \lambda_i = a$) yields (38), the average value we started with

$$x_+^* = \frac{N \cdot a \cdot x_+ + a \cdot doc_{N+1}}{(N+1)a} = \frac{N \cdot x_+ + doc_{N+1}}{N+1} = \frac{\sum_{i=1}^{N+1} doc_i}{N+1}.$$

6.1.2 Matching of *Priority*

Technically, we could just compute the difference between the *Priority* profile x_+ and *Priority* label of a document *doc*. Instead, we try to make the distance measure meaningful. Suppose a user is normally satisfied with documents labeled "neutral" i.e. $doc = 0$. We can still assume that he or she does not think there is something wrong in getting documents with "better" ranking ($doc > 0$) by the journalists. Rather, we could infer that documents with worse *Priority* ranking ($doc < 0$) are undesirable.

Let us assume that the user profile for *Priority* is learned using positive feedback in some of the ways described in previous section 6.1.1 and is thus a real number x_+ in the range $[-M, M]$. The document has a *Priority* label $doc \in \{-M, \ldots, M\}, \quad doc \in \mathbb{Z}$. A distance measure for *Priority* could be

$$D_{abs}(x_+, doc) = \begin{cases} 0 & \text{if} \quad doc \geq x_+ \\ x_+ - doc & \text{if} \quad doc < x_+. \end{cases} \tag{44}$$

There is no penalty for exceeding the profile value x_+ but only for going below it. This measure has the advantage of being absolute in the sense that the values can be compared between different profile values x_+. But, of course, it has different range for every possible value of x_+ and can be therefore hard to interpret (see figure 20). Its range is $[0, x_+ + M]$, which is $[0, 2M]$ in the maximal case.

To fix the scale of the measure it can be normalized (see also figure 20)

$$D_{norm}(x_+, doc) = \begin{cases} 0 & \text{if} \quad doc \geq x_+ \\ \frac{x_+ - doc}{x_+ + M} & \text{if} \quad doc < x_+. \end{cases} \tag{45}$$

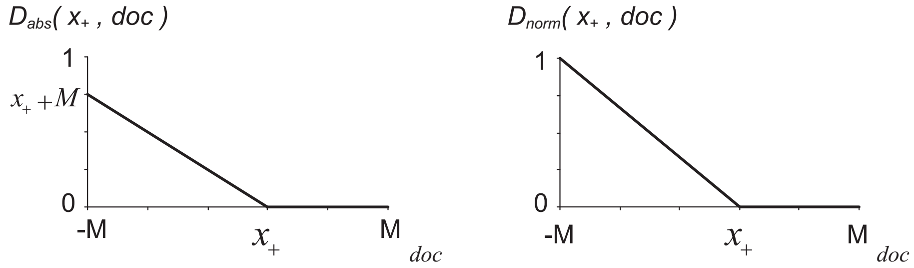

Figure 20: Distance measures $D_{abs}(x_+, doc)$ (eq. (44)) and $D_{norm}(x_+, doc)$ (eq. (45)). The normalization fixes the scale of the measure but makes the values profile dependent.

6.1.3 Negative Feedback and Negative Profile for *Priority*

If a negative profile is considered necessary, it must be totally separate from the positive one, aiming at averaging information of those *Priority* values that relate to the documents with negative feedback. Let us denote the magnitude of the given feedback $\lambda_i < 0$ by $\beta_i = |\lambda_i|$. The equation for the negative profile x_- that corresponds to the general case for the positive profile (41) is

$$x_-^* = \frac{\frac{W_N \cdot x_-}{f} + \beta_{N+1} \cdot doc_{N+1}}{\frac{W_N}{f} + \beta_{N+1}} , \tag{46}$$

where W_N is defined analogously

$$\begin{cases} W_1 & = \beta_1 \\ W_{N+1} & = \frac{W_N}{f} + \beta_{N+1} \end{cases} \tag{47}$$

which is equivalent to

$$W_N = \sum_{i=1}^{N} \left(\frac{1}{f}\right)^{N-i} \cdot \beta_i \; . \tag{48}$$

Again, setting the forgetting parameter f equal to 1 the weighted sum results and setting all feedback equal ($\forall\, i \quad \beta_i = b$) we get the average value of the priorities of the negatively judged documents.

If the positive and negative profiles of a user appear in correct order, some reasonable assumptions of the user's preferences can be made. The absolute distance measure (analogous to equation (44)) could be

$$D_{abs}(x_+, x_-, doc) = \begin{cases} 0 & \text{if} \quad doc \geq x_+ \\ x_+ - doc & \text{if} \quad doc \in [x_-, x_+] \\ 1 & \text{if} \quad doc < x_- \end{cases} \tag{49}$$

and this can be normalized to the range $[0, 1]$ if desired, which yields equation (50) (see also figure 21)

$$D_{norm}(x_+, x_-, doc) = \begin{cases} 0 & \text{if} \quad doc \geq x_+ \\ \frac{x_+ - doc}{x_+ - x_-} & \text{if} \quad doc \in [x_-, x_+] \\ 1 & \text{if} \quad doc < x_- \quad . \end{cases} \tag{50}$$

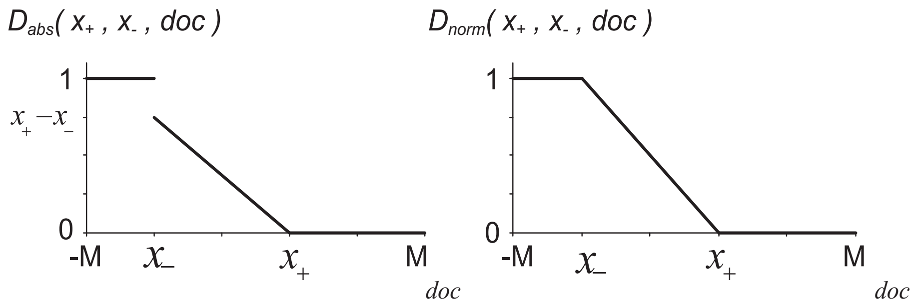

Figure 21: Distance measures $D_{abs}(x_+, x_-, doc)$ and $D_{norm}(x_+, x_-, doc)$ (equations (50) and (49)). It is assumed that $x_- \leq x_+$.

Note that in the special case where positive and negative profiles coincide, $x_+ = x_-$, both measures (49) and (50) reduce to

$$D(x_+, x_-, doc) = \begin{cases} 0 & \text{if} \quad doc \geq x_+ \\ 1 & \text{if} \quad doc < x_- \end{cases} . \tag{51}$$

If, on the other hand, the order of positive and negative profiles is not correct, i.e. $x_+ < x_-$, the reasonable interpretation would be that *Priority* is not essential to the user's preferences.

6.1.4 Compatibility of Learning and Matching for *Priority*

We would like to find that positive feedback decreases the distance between the profile and the document considered. This issue was already dealt with in sections 5.2.1, 5.2.2 and 5.2.3 for the *Subject* dimension of metadata. We now prove it for the *Priority* dimension.

Let us first examine the case with only the positive profile for *Priority*, x_+. We have to consider separately the cases 1) $doc \geq x_+$ and 2) $doc < x_+$.

Case 1)

$$doc \geq x_+ \quad \Longrightarrow D_{abs}(x_+, doc) = D_{norm}(x_+, doc) = 0.$$

$$x_+^* = \frac{\frac{W_N}{f}\, x_+ + \lambda\, doc}{\frac{W_N}{f} + \lambda} \leq \frac{\frac{W_N}{f}\, doc + \lambda\, doc}{\frac{W_N}{f} + \lambda} = doc,$$

so, after feedback we still have $doc \geq x_+^*$ and thus, the distance remains zero, $D_{abs}(x_+^*, doc) = D_{norm}(x_+^*, doc) = 0$.

Case 2)

$doc < x_+ \quad \Longrightarrow D_{abs}(x_+, doc) = x_+ - doc$ by (44)

or, by the normalized version (45)

$D_{norm}(x_+, doc) = \dfrac{x_+ - doc}{x_+ + M}.$

After applying positive feedback once the profile is

$$x_+^* = \frac{\frac{W_N}{f}\, x_+ + \lambda\, doc}{\frac{W_N}{f} + \lambda} < \frac{\frac{W_N}{f}\, x_+ + \lambda\, x_+}{\frac{W_N}{f} + \lambda} = x_+ \tag{52}$$

and, on the other hand,

$$x_+^* > \frac{\frac{W_N}{f}\, doc + \lambda\, doc}{\frac{W_N}{f} + \lambda} = doc\ . \tag{53}$$

Combining inequalities (52) and (53) above, we get

$doc < x_+^* < x_+,$ which yields

$D_{abs}(x_+^*, doc) = x_+^* - doc < x_+ - doc = D_{abs}(x_+, doc)$.

We then continue with the normalized measure (45)

$$D_{norm}(x_+^*, doc) = \frac{x_+^* - doc}{x_+^* + M}$$

$$= \frac{\left(\frac{W_N}{f}\, x_+ + \lambda\, doc\right) - doc\left(\frac{W_N}{f} + \lambda\right)}{\left(\frac{W_N}{f}\, x_+ + \lambda\, doc\right) + M\left(\frac{W_N}{f} + \lambda\right)}$$

$$= \frac{x_+ - doc}{x_+ + M + \underbrace{\frac{\lambda f}{W_N}}_{>0}\underbrace{(doc + M)}_{\geq 0}} \leq \frac{x_+ - doc}{x_+ + M}$$

$$= D_{norm}(x_+, doc).$$

So, for both measures (44) and (45) the distance after positive feedback is less than or equal to the distance before feedback.

□

Next, we take the negative profile into consideration as well and examine the distance measures (49) and (50). We assume that $x_- \leq x_+$ since otherwise the distance is undefined and we assume it has no relevance. When positive feedback is applied the negative profile remains unchanged, $x_-^* = x_-$. There are three separate cases to be dealt with: 1) $doc \geq x_+ \geq x_-$,
2) $doc \leq x_- \leq x_+$ and 3) $x_- < doc < x_+$.

Case 1)

$doc \geq x_+ \geq x_- \quad \Longrightarrow D_{abs}(x_+, x_-, doc) = D_{norm}(x_+, x_-, doc) = 0.$
With the same argument as in the proof for the positive profile, $x_+^* \leq doc.$
So, the distance remains zero, $D_{abs}(x_+^*, x_-^*, doc) = D_{norm}(x_+^*, x_-^*, doc) = 0.$

Case 2)

$doc \leq x_- \leq x_+ \quad \Longrightarrow D_{abs}(x_+, x_-, doc) = D_{norm}(x_+, x_-, doc) = 1.$
Since $x_-^* = x_-$, everything that is true for x_- holds for the new negative profile x_-^* as well. In particular, $doc \leq x_-^*$, thus the distance does not change, i.e., $D_{abs}(x_+^*, x_-^*, doc) = D_{norm}(x_+^*, x_-^*, doc) = 1.$

Case 3)

$x_- < doc < x_+.$
With the same arguments as for the proof for the positive profile
$x_+^* < x_+$ and $x_+^* > doc$, and thus, we get the following ordering
$x_- < doc < x_+^* < x_+$ .Therefore, the absolute distance (49) decreases:
$D_{abs}(x_+^*, x_-^*, doc) = x_+^* - doc < x_+ - doc = D_{abs}(x_+, x_-, doc).$
We shall next examine the normalized measure (50)

$$D_{norm}(x_+^*, x_-^*, doc) = \frac{x_+^* - doc}{x_+^* - x_-^*}$$
$$= \frac{\left(\frac{W_N}{f}\, x_+ + \lambda\, doc\right) - doc\left(\frac{W_N}{f} + \lambda\right)}{\left(\frac{W_N}{f}\, x_+ + \lambda\, doc\right) - x_-\left(\frac{W_N}{f} + \lambda\right)}$$
$$= \frac{x_+ - doc}{x_+ - x_- + \underbrace{\frac{\lambda f}{W_N}}_{>0}\underbrace{(doc - x_-)}_{>0}} < \frac{x_+ - doc}{x_- x_-}$$
$$= D_{norm}(x_+, x_-, doc).$$

So, applying positive feedback results in decrease in the distance measure for *Priority*.

$\square$

We still need to check whether negative feedback increases distance analogously. Negative feedback affects only the negative profile and leaves the positive profile untouched, so $x_+^* = x_+$. Again, we have three cases to consider, 1) $doc \geq x_+ \geq x_-$, 2) $doc \leq x_- \leq x_+$, and 3) $x_- < doc < x_+$.

Case 1)

$doc \geq x_+ \geq x_- \quad \Longrightarrow \quad D_{abs}(x_+, x_-, doc) = D_{norm}(x_+, x_-, doc) = 0.$
If $x_-^* \leq x_+^* = x_+$, the distance remains zero,
$D_{abs}(x_+^*, x_-^*, doc) = D_{norm}(x_+^*, x_-^*, doc) = 0.$
If $x_-^* > x_+^*$, we conclude that there is a contradiction and
Priority should not be used for this user.

Case 2)

$doc \leq x_- \leq x_+ \quad \Longrightarrow \quad D_{abs}(x_+, x_-, doc) = D_{norm}(x_+, x_-, doc) = 1.$

$$x_-^* = \frac{\frac{W_N}{f}\, x_- + \beta\, doc}{\frac{W_N}{f} + \beta} \geq \frac{\frac{W_N}{f}\, doc + \beta\, doc}{\frac{W_N}{f} + \beta} = doc \tag{54}$$

and, on the other hand,

$$x_-^* \leq \frac{\frac{W_N}{f}\, x_- + \beta\, x_+}{\frac{W_N}{f} + \beta} \leq \frac{\frac{W_N}{f}\, x_+ + \beta\, x_+}{\frac{W_N}{f} + \beta} = x_+ = x_+^* \,. \tag{55}$$

Using, inequalities (54) and (55) we get the following ordering
$doc \leq x_-^* \leq x_+ = x_+^*$ and, therefore, the distance remains equal to 1 ,
$D_{abs}(x_+^*, x_-^*, doc) = D_{norm}(x_+^*, x_-^*, doc) = 1$.

Case 3)

$x_- < doc < x_+ \quad \Longrightarrow \quad D_{abs}(x_+, x_-, doc) = x_+ - doc$ and
$D_{norm}(x_+, x_-, doc) = \dfrac{x_+ - doc}{x_+ - x_-}$. The new profiles are

$$x_-^* > \frac{\frac{W_N}{f}\, x_- + \beta\, x_-}{\frac{W_N}{f} + \beta} = x_- \tag{56}$$

and

$$x_-^* < \frac{\frac{W_N}{f}\, doc + \beta\, doc}{\frac{W_N}{f} + \beta} = doc\,. \tag{57}$$

So, by (56) and (57), we get the following ordering
$x_- < x_-^* < doc < x_+ = x_+^*$ and, therefore,
$D_{abs}(x_+^*, x_-^*, doc) = x_+^* - doc = x_+ - doc = D_{abs}(x_+, x_-, doc)$

$$D_{norm}(x_+^*, x_-^*, doc) = \frac{x_+^* - doc}{x_+^* - x_-^*}$$
$$= \frac{x_+ - doc}{x_+ - x_-^*} > \frac{x_+ - doc}{x_+ - x_-}$$
$$= D_{norm}(x_+, x_-, doc).$$

We can conclude that negative feedback increases distance. □

6.2 Other Flat Metadata Structures

The *Industry* dimension is a flat list of industries and the *Company* dimension is a list of company names accordingly (See section 3.1). In these dimensions the documents only get the list of hits, no weights implying the magnitudes of the hits. So, the document representations are Boolean vectors, but this does not imply that the user profile needs to be a Boolean vector.

6.2.1 Learning and Matching in these Dimensions

There are essentially two ways to implement the same kind of learning method as in the *Subject* dimension introduced in chapter 5. Either we take the binary document vector $\mathbf{d}$ as it is and apply the learning method to the profile $\mathbf{x}$ as usual, or we first normalize the document vector $\mathbf{d}$ according to the 1-norm and after that apply the learning method. In both cases the equation for the learning method is

$$\mathbf{x}^* = \frac{\mathbf{x}}{f} + \lambda\mathbf{d} \ ,$$

where λ is the learning parameter and f is the forgetting parameter as before.

If the document vector $\mathbf{d}$ is normalized, every document gives an equal contribution to the profile and if it is not normalized, every appearance of an industry or a company name gives an equal contribution. So the choice is semantic rather than computational or theoretic.

Coarse measure (2) (taking only positive profile into account) or (4) (negative profile considered as well) can be applied in these dimensions but both the documents and the profiles must be normalized before distance calculations. A fine measure is also necessary, if more accuracy is required in discriminating the documents matching to larger weights in the profile from the documents matching to smaller weights in the profile. We face the same problems as in sections 4.2 and 5.2.2 in choosing the fine measure.

Taking the negative profile into account does not introduce any new problems here since there is no hierarchy and hence positive and negative values cannot overlap.

7 Social Information Filtering and Clustering of User Profiles

7.1 Social Filtering

The filtering techniques we have discussed so far are all so called *content based filtering* techniques, since the filtering decisions are based on the content or a description of the content, such as metadata [18]. *Social information filtering* is designed for situations where the content cannot be objectively described. For example, preferring different kinds of music, books or wines is a matter of taste. If two users have approximately same taste, they could successfully recommend new items to each other [19]. The key issue is to find similar users.

Since in (pure) social filtering systems there are no descriptions of the items, the only thing the similarity measure can be based on is likes and dislikes of the users to specific items. This is the weakest point: if there are vast amounts of items or the set of items is expanding all the time, it is rather unlikely that two similar users have explicitly expressed their likes or dislikes of the same set of objects. Those items that the two users in comparison have given no feedback on cannot be used in deciding whether the users are similar.

Another weakness is that when new items become available, no one has given any feedback for them and hence no one recommends them further. Unfortunately, in a typical information filtering system, there are masses of new timely documents on which every user would like to receive recommendations rather than give them.

A certain mass of profiles is needed for them to overlap sufficiently and a certain mass of feedback is required from each user for the profile to be reliable. One or two shared items among two users does not provide enough information to decide whether they have similar tastes.

7.1.1 Social Filtering with Content Based Profiles

Despite the fact that the profiles in our system are content based, we can still enjoy the advantages of social filtering. There are essentially two different ways to do it.

1. Searching new areas of interest for a user on the basis of the interests of similar users.

2. Searching particular interesting items (in our case, documents) based on the feedback similar users have given to particular items.

The same measures for similarity between users can be used for both approaches but otherwise they are quite different. Appropriate distance measures are discussed below, in section 8.2.3. Next, we elaborate a little on both approaches. In the following, we shall denote by $\mathbf{v}$ the profile we are making the recommendation to.

Searching of areas of interest

The main idea is to search for profiles that are somehow similar to $\mathbf{v}$, and recommend subjects that are covered in the positive profiles of the similar users but are uncovered in profile $\mathbf{v}$. We interpret zero weights as neutral, unknown nodes of the profile. This could introduce serendipity to the system. In practice, the recommended objects would be particular documents that match the recommended subjects well.

So, how are these recommended items to be selected? We suggest first constructing a query profile based on the similar profiles. Then the query profile is used in the same way as normal profiles to find matching documents. A query profile $\mathbf{q}_\ell$ constructed of one of the neighbors $\mathbf{y}_\ell$, could be

$$q_{\ell\, i} = \begin{cases} y_{\ell\, i} & , \quad \text{if} \quad v_i = 0 \wedge y_{\ell\, i} > 0 \\ 0 & , \quad \text{otherwise} . \end{cases} \tag{58}$$

The query profile $\mathbf{q}_\ell$ is then normalized according to the 1-norm to get $\hat{\mathbf{q}}_\ell$. The neighbors' query profiles could be weighted by a coefficient w_ℓ that depends on the distance $D(\mathbf{v}, \mathbf{y}_\ell)$ between the neighbor $\mathbf{y}_\ell$ and profile $\mathbf{v}$. The coefficient w_ℓ could be defined as

$$w_\ell = \begin{cases} 1 - D(\mathbf{v}, \mathbf{y}_\ell) & , \quad \text{if} \quad D(\mathbf{v}, \mathbf{y}_\ell) \leq L \\ 0 & , \quad \text{otherwise} , \end{cases} \tag{59}$$

where L defines the radius of neighborhood, and the distance measure $D(\mathbf{v}, \mathbf{y})$ is scaled to the range $[0, 1]$. The size of the neighborhood can also be set in terms of the number of neighbors. Finally, the query profile could be

$$\mathbf{q}_v = \sum_\ell w_\ell \cdot \hat{\mathbf{q}}_\ell . \tag{60}$$

The weighting according to the distance (or similarity) is a common practice in social filtering and a well known technique for long in other research areas utilizing neighborhoods, e.g., neural networks [20].

Searching of particular interesting documents

Here, the main idea is not to find uncovered areas in profile **v** but simply good individual documents. Human beings can classify documents with subjective but very efficient arguments like "well written", "fun", "lucid", and so on. Including this kind of information to the representation model is impossible since the model needs to be fairly objective to serve all users at the same time. In theory, it is possible to find these subjectively good documents to profile **v** just by taking those documents the similar users have judged positively. The most similar users' recommendations would have priority to those judged by less similar users.

7.2 Clustering Profiles

In many applications it is of interest to make fewer clusters of similar user profiles instead of dealing separately with them all. It is ambiguous what is meant by similar users. Actually, it depends on the purpose of the resulting clusters what kind of similarity is meaningful. A service provider could, e.g., like to know a good way to design and price its services according to customer segments. These segments can be found by clustering users on the basis of certain relevant features of their profiles. Relevant information could be how often a customer uses the service, at what time a customer uses the service, how much a customer is prepared to pay for the quality of the content etc. If, on the other hand, the issue is how to target advertising to potential customers, the *Subject* could be more relevant as classifying information.

In Encyclopedia of Artificial Intelligence [21] R. S. Michalski and R. E. Stepp define clustering as follows

> **Definition**
> *Clustering is usually viewed as a process of grouping physical or abstract objects into classes of similar objects. According to this view, in order to cluster objects, one needs to define a measure of similarity between the objects and then apply it to determine classes. Classes are defined as collections of objects whose intraclass similarity is high and interclass similarity is low.*

Choosing relevant features for representing user profiles and constructing a meaningful distance measure between the profiles are the key issues in clustering of profiles. The features and the similarity measure must be tailored for each purpose. In the next section we introduce the Maximin-Distance algorithm [22] and the suitable distance measures for clustering are discussed in chapter 8.

7.2.1 Maximin-Distance Clustering Method

In our tests we have used a simple, heuristic, unsupervised clustering method, called the Maximin-Distance algorithm [22]. The algorithm selects a number of centers for the clusters, called *centroids*. When a given end condition is satisfied the rest of the items that have not been selected are each assigned to a cluster on the basis of their distances to the centroids. Let us define the distance between an item and a set of items as follows: if x is an item and $C = \{m_1, m_2, \ldots, m_n\}$ is a set of items, the distance between x and C is

$$d(x, C) = \min_k d(x, m_k).$$

The algorithm proceeds as follows

step 1 Choose an arbitrary starting point as the first centroid m_1.

step 2 Find the item that has the greatest distance to m_1 and select that item as the second centroid m_2, provided that the end condition is not yet satisfied.

step 3 Denote by C_k the set of centroids selected so far and choose as the next candidate that item that has the greatest distance to C_k among the items that have not been selected yet.

step 4 Check whether the end condition is fulfilled and if not, select the candidate as the next centroid m_{k+1} and continue with step 3.

step 5 Each non-centroid item x belongs to the cluster i, whose centroid is closest to the item x, i.e., $i = \arg\min_k d(x, m_k)$. Each centroid belongs to the cluster it represents.

There are several options for end conditions, e.g.,

- The desired number of clusters, which simply fixes the number of iterations beforehand.
- A limit L for the distance between the centroid to be selected and the set of previously selected centroids. If $d(m_{k+1}, C_k) < L$, the algorithm stops.
- A limit L for the quotient of the distances of the item under evaluation m_{k+1} and the previous one m_k. If $d(m_{k+1}, C_k)/d(m_k, C_{k-1}) < L$, the algorithm stops.
- A limit L for the difference of the distance of the item under evaluation m_{k+1} from the previous one m_k. If $d(m_k, C_{k-1}) - d(m_{k+1}, C_k) < L$, the algorithm stops.

8 Measuring Distance between Two User Profiles

8.1 Different Measures for Different Purposes

The requirements for an adequate similarity measure between two profiles depend on what the measure is used for. When trying to find clusters of similar profiles it is important that the measure is symmetric and for most of the well known clustering methods to work the triangle inequality must also be fulfilled. On the other hand, in social filtering one typically searches the nearest neighbors to a single chosen profile at a time. In this case all other profiles are compared to this center point but not amongst themselves. Suppose the distances have been scaled to the interval $[0, 1]$. Now, whichever profile x is chosen as the center point, it should be possible to construct another profile y with the maximal distance 1 to x. When clustering profiles, however, it is obvious that one cannot get very far from a profile that is "in the middle" even though the most extreme profiles can be very far away from each other.

8.2 Similarity Measures for Social Filtering

8.2.1 Discussion about Objective of Similarity Measure

Let us have a setting where users are asked to give ratings for items (books, music, etc.) they are familiar with according to their own likes and dislikes. Let the scale be for example $\{1, 2, 3, 4, 5\}$, 5 being the best and 1 the worst rating. As a first approximation one could just calculate the plain distances between two profiles using the 1-norm

$$\sum_{i=1}^{N} |a_i - b_i|$$

where a_i = person A's rating for item i and b_i = person B's rating for item i.

But this sum naturally has a maximum value and should be seen in relation to that maximum. For each item i:

$$0 \leq |a_i - b_i| \leq |\text{max rating - min rating }|$$

So,

$$0 \leq \sum_{i=1}^{N} |a_i - b_i| \leq N \cdot |\text{max rating - min rating }| \quad (= 4N \quad \text{in this case})$$

However, once a person has given his or her ratings in most cases it is impossible to achieve the difference $N \cdot |\text{max rating - min rating }|$. If person A has fixed his or her preferences $\{a_1, a_2, \ldots, a_N\}$ (A's profile), this already sets a limit to the value $\sum_{i=1}^{N} |a_i - x_i|$. Let us denote the greatest possible difference to A's profile by A_{max}. For each component i there is a maximum value A_{imax}

$$A_{imax} = \begin{cases} |\max - a_i|\,, & \text{if} \quad a_i \leq \frac{\lfloor \max + \min \rfloor}{2} \\ |\min - a_i|\,, & \text{if} \quad a_i > \frac{\lfloor \max + \min \rfloor}{2} \end{cases}$$

and

$$A_{max} = \sum_{i=1}^{N} A_{imax}.$$

Given A's profile $\{a_1, a_2, \ldots, a_N\}$ this is straightforward calculation. For example, using the scale $\{1, 2, 3, 4, 5\}$ we get

$$\begin{aligned} A_{max} &= 4 \cdot |\{i \mid a_i = 1 \quad \text{or} \quad a_i = 5\}| \\ &+ 3 \cdot |\{i \mid a_i = 2 \quad \text{or} \quad a_i = 4\}| \\ &+ 2 \cdot |\{i \mid a_i = 3\}| \\ &\leq 4N. \end{aligned}$$

To get a percentage of similarity $\text{sim}(a, x)$ compared to A's profile, write

$$\text{sim}(a, x) = \left(1 - \frac{\sum_{i=1}^{N} |a_i - x_i|}{A_{max}}\right) \cdot 100\% \,. \tag{61}$$

This measure is not symmetric but sees things in A's perspective. If we want the measure between two profiles a and b to be symmetric, both measures $\text{sim}(a, b)$ and $\text{sim}(b, a)$ must be taken into account.

<u>Example:</u>

If person A has only 2's and 4's as ratings, the maximum difference to each rating is 3. So $A_{max} = 3N$. The most differing profile b must consist only of ratings 1 and 5. To profile b, the maximum difference for each rating is 4 $\implies \quad B_{max} = 4N$.

$$\text{sim}(a, b) = 1 - \frac{\sum_{i=1}^{N} |a_i - b_i|}{A_{max}} = 1 - \frac{3N}{3N} = 0$$

since $\forall\, i \quad |a_i - b_i| = 3$ but in B's perspective

$$\text{sim}(b, a) = 1 - \frac{\sum_{i=1}^{N} |a_i - b_i|}{B_{max}} = 1 - \frac{3N}{4N} = \frac{1}{4}.$$

If a symmetric summary of both a's and b's points of view is needed, the arithmetic (62) or the geometric mean (63) could be used:

$$\mathrm{sim}_{AB} = \frac{\mathrm{sim}(a,b) + \mathrm{sim}(b,a)}{2} \tag{62}$$

$$\mathrm{sim}_{AB} = \sqrt{\mathrm{sim}(a,b) \cdot \mathrm{sim}(b,a)}\ . \tag{63}$$

The geometric mean would imply that if either one of the profiles differs as much as possible from the other one (i.e. $\mathrm{sim}(a,b) = 0$ or $\mathrm{sim}(b,a) = 0$) the unified similarity would also be 0. If that is not desirable but both differences ought to be maximal for the unified difference to be maximal ($\mathrm{sim}_{AB} = 0$), one could also use

$$\mathrm{sim}_{AB} = 1 - \sqrt{\frac{\sum_{i=1}^{N} |a_i - b_i|}{A_{max}} \cdot \frac{\sum_{i=1}^{N} |a_i - b_i|}{B_{max}}} = 1 - \frac{\sum_{i=1}^{N} |a_i - b_i|}{\sqrt{A_{max} \cdot B_{max}}}. \tag{64}$$

8.2.2 Cosine Measure

A popular way for this kind of comparisons is to map the ratings to such a scale that 0 is in the middle and the best ratings are positive and the worst ratings negative. After the mapping the *cosine measure* (see below) is used to see which of the profiles point to similar directions [19]. Using, e.g., a linear mapping the original profile a could be mapped to $\hat{a}_i = a_i - 3$. For ratings in the earlier mentioned scale $\{1, \ldots, 5\}$ the new values are

x_i	$\hat{x}_i = x_i - 3$
1	-2
2	-1
3	0
4	1
5	2

.

If we denote the original vectors by a and b and their mapped counterparts by $\hat{a}$ and $\hat{b}$, the cosine based similarity measure will be

$$\mathrm{sim}(a,b) = \cos(\hat{a}, \hat{b}) = \frac{\sum_{i=1}^{N} \hat{a}_i \cdot \hat{b}_i}{\sqrt{\sum_{i=1}^{N} \hat{a}_i^2 \sum_{i=1}^{N} \hat{b}_i^2}} \qquad \in [-1, 1],$$

where a negative value means dissimilarity and a positive value similarity. For comparison, let us put it in the same scale as the previously mentioned similarity measures

$$\mathrm{sim}_{cos} = \frac{1}{2} \cdot \Big(1 + \cos(\hat{a}, \hat{b})\Big) \qquad \in [0, 1].$$

Whatever the ratings $\hat{a}_i$, it is always possible to choose a profile $\hat{x}$ that points to the exactly opposite direction simply by choosing $\hat{x}_i = -\hat{a}_i$. Then $\cos(\hat{a}, -\hat{a}) = -1$ implies $\text{sim}_{cos} = \frac{1}{2} \cdot (1 - 1) = 0$.

On the other hand, the semantics is clearly not the same as in the beginning. If person A chooses a rating $a_i = 4$ meaning "I like it just a little bit" and the mapped rating is $\hat{a}_i = 1$, the totally disagreeing opinion (according to cosine measure logic) is $x_i = 2 \approx$ "I dislike it a little bit" instead of the more reasonable choice $x_i = 1 \approx$ "I hate it". Hence, although being easy to handle the cosine measure fails to capture the meaningful semantics desirable for a similarity measure.

Inadequacy of the cosine measure can be seen clearly using a simple example.

Example:

Consider the profile

$$a = \{5, 1, 4, 2\} \implies \hat{a} = \{2, -2, 1, -1\}.$$

Now, in order to achieve the maximal distance to a the other profile has to consist of opposite numbers $\{-2, 2, -1, 1\}$. Ratings 2 and -2 are surely opposites to one another ("likes a lot" and "hates it") but rationally thinking $\hat{a} = 1 \approx$ "likes a little" differs the most from "hates it" rather than "dislikes a little". The opposites of the ratings in this semantic sense are displayed in table 2.

$\hat{a}_i$	opposite
-2	2
-1	2
0	0
1	-2
2	-2

Table 2: Opposite ratings in a semantic sense.

8.2.3 Distance Measure for Social Filtering with Content Based Profiles

In order to compare content based user profiles we could apply the same idea as in section 8.2.1. Once profile a is fixed, it should be possible to achieve the maximal distance 1. An analogue to the measure defined in equation (61) on page 47 would be

$$D(\mathbf{a}, \mathbf{b}) = 1 - \mathrm{sim}_{ab} = \frac{\sum_{I_{ab}} |a_i - b_i|}{A_{max}} \ . \tag{65}$$

We shall first examine the case with only positive profiles. If ε is set as a limit to the weight being significant and profile a is already fixed, we have

$$I_{ab} = \{i \mid a_i > \varepsilon \wedge b_i > \varepsilon\} \ .$$

The maximum of the sum in equation (65) is

$$A_{max} = \begin{cases} 1 - \varepsilon \ , & \text{if} \quad a_{min} = 1 \\ 2(1 - a_{min} - N_a \varepsilon) \ , & \text{if} \quad a_{min} < 1 \ , \end{cases}$$

where we have defined

$$a_{min} = \min\{a_i \mid a_i > \varepsilon\} \quad \text{and} \quad N_a = \begin{cases} |\{i \mid a_i > 2\varepsilon\}| & \text{if } a_{min} \leq 2\varepsilon \\ |\{i \mid a_i > 2\varepsilon\}| - 1 & \text{if } a_{min} > 2\varepsilon \ . \end{cases}$$

If we allow arbitrarily small weights to be significant, we can let $\varepsilon \to 0$. Then

$$I_{ab} = \{i \mid a_i > 0 \wedge b_i > 0\} \quad \text{and} \quad A_{max} = \begin{cases} 1 \ , & \text{if} \quad a_{min} = 1 \\ 2(1 - a_{min}) \ , & \text{if} \quad a_{min} < 1 \ , \end{cases}$$

where $a_{min} = \min\{a_i \mid a_i > 0\}$.

With negative profiles the situation is much simpler, since $A_{max} = 4$ always. Then, the index set I_{ab} must be redefined as follows

$$I_{ab} = \{i \mid |a_i| > \varepsilon \wedge |b_i| > \varepsilon\} \ .$$

8.2.4 Reliability of Distance

We shall next introduce a measure for the reliability of the distance between two profiles. As we must restrict the comparison of two user profiles to those nodes they have in common, the number of common nodes is a kind of reliability measure of the distance. To be more exact, the amount of weights the profiles have in the common area should be considered.

Let us denote by W_A the sum of weights profile a has in the common area, i.e.,

$$W_A = \sum_{I_{ab}} |a_i| ,$$

where $I_{ab} = \{i \mid |a_i| > 0 \wedge |b_i| > 0\}$. The reliability r_{AB} of a compared to b can be defined as the proportion of W_A to the total weight. With just positive profiles we get $r_{AB} = W_A$, while with the negative profiles considered as well we have $r_{AB} = \frac{1}{2}W_A$.

<u>Example:</u>

Let the two profiles in comparison be $\mathbf{u} = (0, 0, 0.9, 0.1, -1, 0, 0, 0)$ and $\mathbf{v} = (0.9, -0.1, 0, 0.1, 0, 0, 0, -0.9)$, shown also in figure 22. Since only those areas where both profiles have nonzero weights are compared, profiles $\mathbf{u}$ and $\mathbf{v}$ are considered identical. All meaningful distances between two user profiles equal to zero when the profiles are identical. So, the fact that the profiles have very little in common ensures that the distance is small. Here we have $W_U = 0.1$ and $W_V = 0.1$. The reliability is $r_{UV} = r_{VU} = 0.05 = 5\%$ if negative profiles are taken into consideration and $r_{UV} = r_{VU} = 0.1 = 10\%$ with only positive profiles.

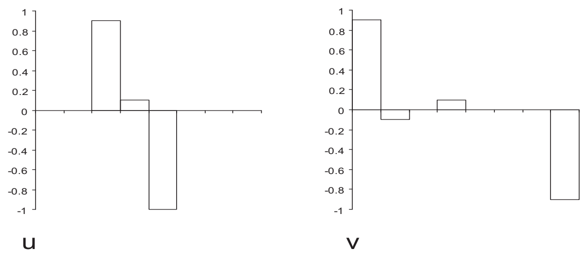

Figure 22: Profiles $\mathbf{u}$ and $\mathbf{v}$ have nothing in common except the fourth weight.

If we consider the situation symmetric, the total reliability should be a symmetric combination of r_{AB} and r_{BA}, e.g., the arithmetic or the geometric mean

$$r_{arith} = \frac{1}{2}(r_{AB} + r_{BA}) \quad \text{and} \quad r_{geom} = \sqrt{r_{AB} \cdot r_{BA}} \,.$$

8.3 Distance Measure Candidates for Clustering of User Profiles

As stated at the beginning of this chapter, for clustering of user profiles the conditions posed for the distance measure differ from those for social filtering. Next, we shall discuss the conditions for clustering and find some measures that fulfill those conditions.

8.3.1 Requirements for Distance Measures for Clustering of User Profiles

The distance measure for clustering user profiles must meet at least the following requirements

$$\begin{cases} 1. & x = y \implies D(x,y) = 0 & \\ 2. & D(x,y) = D(y,x) & \text{(symmetry)} \\ 3. & D(x,y) + D(y,z) \geq D(x,z) & \text{(triangle inequality)} \end{cases} \quad (66)$$

Note that condition

$$D(x,y) = 0 \implies x = y$$

as in the conditions of a metric is not needed here. If two distinct profiles have a zero distance, they are as close to each other as possible and they need not be identical. If they are identical, it is naturally required that their distance is zero. The triangle inequality guarantees that if two profiles are close to a certain third profile they are close to each other as well. Many clustering methods use centroids as representatives for clusters. The cluster associated to a centroid includes those profiles that are closest to the centroid according to some distance measure. For clustering to make sense, profiles included in the same cluster should be near each other, not just near the centroid.

Any measure that is a metric fulfills these conditions but may not be suitable in the semantic sense. Distance measure based on the familiar 1-norm introduced below in equation (67), e.g., is too strict in demanding that the corresponding weights x_i and y_i in the two profiles x and y should be almost equal in order to get a small distance:

$$D_1(x,y) = \frac{1}{2}\|x-y\|_1 = \frac{1}{2}\sum_i |x_i - y_i| \quad (67)$$

Here, the factor $\frac{1}{2}$ is introduced in order to scale the measure into the range $[0,1]$.

A better idea could be to focus on those weights for which the profiles do not match at all. An appropriate measure concerning the semantics would be

$$D_{SymCoarse}(x, y) = \frac{1}{2}\left(\sum_{y_i=0} x_i + \sum_{x_i=0} y_i\right) \qquad \in [0, 1].$$

$D_{SymCoarse}$ is actually the average value of two coarse measures as defined in (2), the first seeing x as a document and y as a profile and vice versa for the second.

Example:

> Profiles $u = (0.2, 0.8, 0, 0, 0, 0)$ and $v = (0.8, 0.2, 0, 0, 0, 0)$ are surely very similar despite the small difference in their weights (figures 23 and 24).

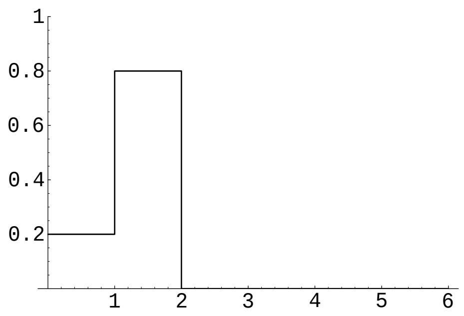

Figure 23: Profile u.

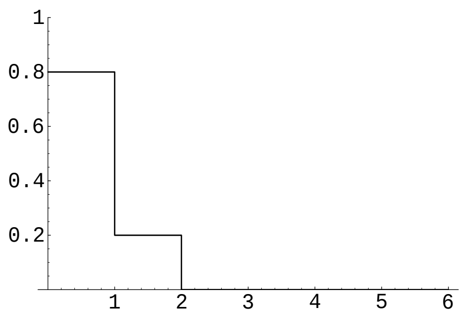

Figure 24: Profile v.

An example of a pair that is not so similar in the semantic sense is $u = (0.2, 0.8, 0, 0, 0, 0)$ and $w = (0.2, 0, 0, 0, 0, 0.8)$ which disagree in the second and the sixth weight (figures 25 and 26).

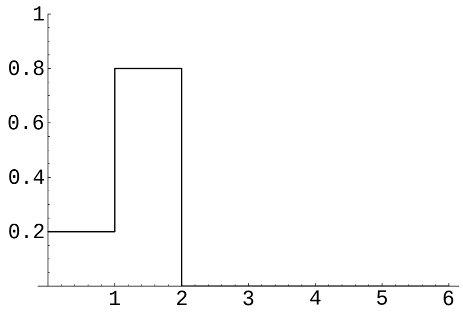

Figure 25: Profile u.

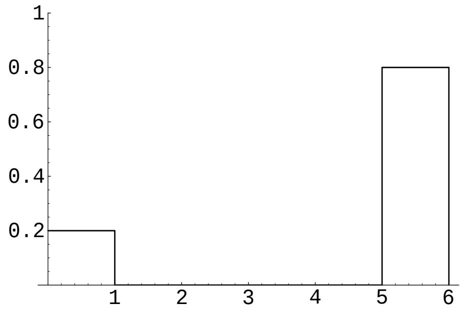

Figure 26: Profile w.

Let us compare the suggested measures $D_1(x, y)$ and $D_{SymCoarse}(x, y)$:

With the 1-norm based measure (67) we get

$$D_1(u, v) = \frac{1}{2}\|u - v\|_1 = 0.6 \qquad \text{and} \qquad D_1(u, w) = \frac{1}{2}\|u - w\|_1 = 0.8\ ,$$

and with $D_{SymCoarse}$ we get

$$D_{SymCoarse}(u, v) = 0 \qquad \text{and} \qquad D_{SymCoarse}(u, w) = 0.8$$

Let us see how well $D_{SymCoarse}$ meets the requirements posed in (66):

1.
$$x = y \implies \quad D_{SymCoarse}(x, y) = \frac{1}{2}\left(\sum_{x_i=0} x_i + \sum_{y_i=0} y_i\right) = 0$$

2. The symmetry of $D_{SymCoarse}(x, y)$ is trivial.

3. Let us denote $I_x = \{i \mid x_i > 0\}$, with corresponding definitions for I_y and I_z. Then
$$D_{SymCoarse}(x, y) = \frac{1}{2}\left(\sum_{i \notin I_y} x_i + \sum_{i \notin I_x} y_i\right) .$$

   We can show by counterexample that the triangle inequality does not hold for $D_{SymCoarse}(x, y)$. With the profiles
$$\begin{cases} x = (0.1, 0.9, 0, 0) \\ y = (0, 0.5, 0.5, 0) \\ z = (0, 0, 0.9, 0.1) \end{cases}$$

   we get the distances
$$\begin{aligned} D_{SymCoarse}(x, y) &= \frac{1}{2}(0.1 + 0.5) = 0.3 \\ D_{SymCoarse}(y, z) &= \frac{1}{2}(0.5 + 0.1) = 0.3 \\ D_{SymCoarse}(x, z) &= \frac{1}{2}(1 + 1) = 1 . \end{aligned}$$

   Thus,
$$D_{SymCoarse}(x, y) + D_{SymCoarse}(y, z) = 0.6 < 1 = D_{SymCoarse}(x, z) ,$$

   which is a pity, because this measure was otherwise very well suited for the task.

8.3.2 Measures Based on the Angle between Vectors

Assuming that the angle between two vectors is a good indicator of similarity, we next discuss some feasible ways to get hold of it. A widely used measure of similarity is the cosine measure that could be converted into a distance measure, e.g., in the following ways (both measures are also normalized to the range $[0, 1]$):

$$D1_{cos}(x, y) = 1 - \cos(x, y)$$

or

$$D2_{cos}(x, y) = \frac{2}{\cos(x, y) + 1} - 1.$$

Both these measures, however, fail to fulfill the triangle inequality, as we shall prove later on.

The angle itself meets the requirements neatly. However, we have to compute the measure using the vector weights and it would be far too complicated to compute the actual angle, even though computing trigonometric functions is feasible.

Proposition 1

If $f(\alpha)$ is a continuous and increasing function of the angle α between two vectors on an interval $[0, h]$ and f is concave on $(0, h)$ and $f(0) = 0$, the triangle inequality holds for f, that is

$$f(\angle(x, y)) + f(\angle(y, z)) \geq f(\angle(x, z))\ .$$

If $f(\alpha) \geq 0$ is a continuous function of the angle between two vectors on an interval $[0, h]$ and f is strictly convex on $(0, h)$ and $f(0) = 0$, there is always a set of three vectors that contradicts the triangle inequality.

Proof:

Let us denote $\alpha = \angle(x, y)$, $\beta = \angle(y, z)$ and $\gamma = \angle(x, z)$. If we use only positive weights, each of these angles is restricted to the interval $[0, \frac{\pi}{2}]$ and if we allow both positive and negative weights the angles lie in the range $[0, \pi]$. It is obvious from basic geometry that $\alpha + \beta \geq \gamma$. We use the most suitable of the equivalent definitions of a convex function [23]:

Definition

A function f is *convex* on an interval (a, b) if for all $s, t, u \in (a, b)$ with $a < s < t < u < b$

$$\frac{f(t) - f(s)}{t - s} \leq \frac{f(u) - f(t)}{u - t}.$$

If the inequality is reversed, we get the definition of a *concave* function. If the inequality is strict, we say that f is *strictly convex* on (a, b).

Let us first assume that f is a continuous and increasing function on $[0, h]$ and that it is concave on $(0, h)$ and $f(0) = 0$. If angle γ is smaller than either of the angles α and β, it is easy to see that the triangle inequality holds

$$\gamma \leq \alpha \quad \Longrightarrow f(\gamma) \leq f(\alpha) \leq f(\alpha) + f(\beta) \ ,$$

since $f(\beta) \geq 0$ and f is increasing implies $\gamma \leq \alpha \implies f(\gamma) \leq f(\alpha)$. The deduction is essentially the same for $\gamma \leq \beta$.

If we allowed $\alpha = 0$, it would lead to the case $\gamma \leq \beta$ that was already considered, and if β were 0 it would lead to the case $\gamma \leq \alpha$. Technically, the remaining possibilities are: 1) $0 < \alpha < \beta < \gamma < h$, 2) $0 < \beta < \alpha < \gamma < h$, 3) $0 < \alpha = \beta < \gamma < h$, 4) $0 < \alpha < \beta < \gamma = h$, 5) $0 < \beta < \alpha < \gamma = h$, and 6) $0 < \alpha = \beta < \gamma = h$. Since the roles of α and β can always be interchanged, it suffices to prove 1), 3), 4) and 6).

<u>Case 1)</u>

$0 < \varepsilon < \alpha < \beta < \gamma < h$. Now, using the definition of a concave function, we have both

$$\frac{f(\beta) - f(\alpha)}{\beta - \alpha} \geq \frac{f(\gamma) - f(\beta)}{\gamma - \beta}$$

and

$$\frac{f(\alpha) - f(\varepsilon)}{\alpha - \varepsilon} \geq \frac{f(\beta) - f(\alpha)}{\beta - \alpha} \ ,$$

from which we get

$$f(\alpha) + f(\beta) \geq f(\gamma) + \frac{\gamma - \beta}{\alpha - \varepsilon} f(\varepsilon) + \frac{\alpha + \beta - \gamma - \varepsilon}{\alpha - \varepsilon} f(\alpha). \tag{68}$$

If we now have $\alpha + \beta > \gamma$, we can denote $\alpha + \beta = \gamma + \delta$, where $0 < \varepsilon < \delta < \alpha$. ($\varepsilon$ can always be chosen small enough and $\beta < \gamma$ ensures that $\delta < \alpha$.) With this notation (68) reduces to

$$f(\alpha) + f(\beta) \geq f(\gamma) + \frac{\alpha - \delta}{\alpha - \varepsilon} f(\varepsilon) + \frac{\delta - \varepsilon}{\alpha - \varepsilon} f(\alpha) \geq f(\gamma) \ .$$

Thus, the triangle inequality holds for this case.

If we, instead, have $\alpha+\beta=\gamma$, (68) reduces to

$$f(\alpha)+f(\beta) \geq f(\gamma)+\frac{\alpha}{\alpha-\varepsilon} f(\varepsilon)-\frac{\varepsilon}{\alpha-\varepsilon} f(\alpha). \tag{69}$$

Since f is continuous on the closed interval $[0, h]$ we can take the limit for $\varepsilon \rightarrow 0^{+}$ on both sides of the inequality (69) and get

$$\begin{aligned}
&\lim_{\varepsilon \rightarrow 0^{+}}(f(\alpha)+f(\beta)) \geq \lim_{\varepsilon \rightarrow 0^{+}} f(\gamma)+\lim_{\varepsilon \rightarrow 0^{+}} \frac{\alpha}{\alpha-\varepsilon} \lim_{\varepsilon \rightarrow 0^{+}} f(\varepsilon)-f(\alpha) \lim_{\varepsilon \rightarrow 0^{+}} \frac{\varepsilon}{\alpha-\varepsilon} \\
&\Longrightarrow f(\alpha)+f(\beta) \geq f(\gamma)+f(0)-f(\alpha) \cdot 0=f(\gamma) .
\end{aligned}$$

Accordingly, the triangle inequality holds in case 1).

Case 3)

$0<\varepsilon<\alpha=\beta<\gamma<h$. Using the definition of a concave function we get

$$\frac{f(\alpha)-f(\varepsilon)}{\alpha-\varepsilon} \geq \frac{f(\gamma)-f(\alpha)}{\gamma-\alpha},$$

from which we get with a short manipulation

$$f(\alpha)+f(\beta)=2 f(\alpha) \geq f(\gamma)+\frac{\gamma-\alpha}{\alpha-\varepsilon} f(\varepsilon)+\frac{2 \alpha-\gamma-\varepsilon}{\alpha-\varepsilon} f(\alpha). \tag{70}$$

If $2 \alpha=\gamma$, (70) will be exactly the same as in case 1).

If $2 \alpha>\gamma$, denote again $2 \alpha=\gamma+\delta$, where $0<\varepsilon<\delta<\alpha$. With this notation (70) reduces to the same inequality as in case 1). So, the triangle inequality holds for case 3).

Case 4)

$0<\varepsilon<\alpha<\beta<w<\gamma=h$. Using the definition of a concave function, we have both

$$\frac{f(\beta)-f(\alpha)}{\beta-\alpha} \geq \frac{f(w)-f(\beta)}{w-\beta}$$

and

$$\frac{f(\alpha)-f(\varepsilon)}{\alpha-\varepsilon} \geq \frac{f(\beta)-f(\alpha)}{\beta-\alpha},$$

from which we get

$$f(\alpha)+f(\beta) \geq f(w)+\frac{w-\beta}{\alpha-\varepsilon} f(\varepsilon)+\frac{\alpha+\beta-w-\varepsilon}{\alpha-\varepsilon} f(\alpha). \tag{71}$$

Since f is continuous on the closed interval $[0, h]$ we can take the limit for $w \to \gamma^-$ on both sides of the inequality (71) and get

$$f(\alpha) + f(\beta) \geq f(\gamma) + \frac{\gamma - \beta}{\alpha - \varepsilon} f(\varepsilon) + \frac{\alpha + \beta - \gamma - \varepsilon}{\alpha - \varepsilon} f(\alpha),$$

from which on the proof is identical with case 1).

Case 6)

$0 < \varepsilon < \alpha = \beta < w < \gamma = h$. Again, using the definition of a concave function we have

$$\frac{f(\alpha) - f(\varepsilon)}{\alpha - \varepsilon} \geq \frac{f(w) - f(\alpha)}{w - \alpha},$$

from which we get with a short manipulation

$$f(\alpha) + f(\beta) = 2f(\alpha) \geq f(w) + \frac{w - \alpha}{\alpha - \varepsilon} f(\varepsilon) + \frac{2\alpha - w - \varepsilon}{\alpha - \varepsilon} f(\alpha). \qquad (72)$$

Since f is continuous on the closed interval $[0, h]$ we can take the limit for $w \to \gamma^-$ on both sides of the inequality (72) as in the proof for case 4) and get

$$2f(\alpha) \geq f(\gamma) + \frac{\gamma - \alpha}{\alpha - \varepsilon} f(\varepsilon) + \frac{2\alpha - \gamma - \varepsilon}{\alpha - \varepsilon} f(\alpha),$$

from which on the proof is identical with case 3). $\square$

Let us then prove that the triangle inequality does not hold for a strictly convex function f. We now assume that f is continuous on $[0, h]$ and strictly convex on $(0, h)$. Furthermore, we assume that $f \geq 0$ and $f(0) = 0$. This time it suffices to find one case for which the triangle inequality does not hold. Let us take, e.g., case 1).

Case 1)

$0 < \varepsilon < \alpha < \beta < \gamma \leq h$. Now, using the definition of a strictly convex function, we have both

$$\frac{f(\beta) - f(\alpha)}{\beta - \alpha} < \frac{f(\gamma) - f(\beta)}{\gamma - \beta}$$

and

$$\frac{f(\alpha) - f(\varepsilon)}{\alpha - \varepsilon} < \frac{f(\beta) - f(\alpha)}{\beta - \alpha},$$

from which we get

$$f(\gamma) > f(\alpha) + f(\beta) + \frac{\gamma - \alpha - \beta - \varepsilon}{\alpha - \varepsilon} f(\alpha) - \frac{\gamma - \beta}{\alpha - \varepsilon} f(\varepsilon). \tag{73}$$

If we now choose α, β and γ so that $\alpha + \beta = \gamma$, equation (73) yields

$$f(\gamma) > f(\alpha) + f(\beta) - \frac{\varepsilon}{\alpha - \varepsilon} f(\alpha) - \frac{\alpha}{\alpha - \varepsilon} f(\varepsilon) \tag{74}$$

and since f is continuous on the closed interval $[0, h]$ we can take the limit for $\varepsilon \to 0^+$ on both sides of the inequality (74) to get

$$f(\gamma) > f(\alpha) + f(\beta) - 0 \cdot f(\alpha) - f(0) = f(\alpha) + f(\beta)$$

which contradicts the triangle inequality. $\square$

The following simple proposition can be found in any elementary textbook of calculus. It tells us how to check whether a twice differentiable function is convex or concave.

Proposition 2

For a twice differentiable function $f(\alpha)$

f *is convex on an interval* $(a, b) \Leftrightarrow f''(\alpha) \geq 0$ *on the interval* (a, b)

and

f *is concave on an interval* $(a, b) \Leftrightarrow f''(\alpha) \leq 0$ *on the interval* (a, b).

Now we have the tools to check easily that the triangle inequality holds neither for $D1_{cos}$ nor $D2_{cos}$, since they are both strictly convex on $(0, \frac{\pi}{2})$ as can be seen from the second derivative. (See appendix A.2 for details.)

Since the cosine can be computed from weights of the profiles and the sine (in our restricted space, with all components ≥ 0) is simply

$$D1_{sin}(x,y) = \sin(x,y) = \sqrt{1-\cos^2(x,y)}, \tag{75}$$

we can compute the sine also in a simple manner using vector weights.

Let us check the conditions (66) for $D1_{sin}(x,y)$

1. $x = y \implies \quad D1_{sin}(x,y) = \sin(x,y) = 0$
2. The symmetry of $D1_{sin}(x,y)$ is trivial.
3. The triangle inequality holds for $D1_{sin}(x,y)$. **Proof:** Denote $\alpha = \angle(x,y)$. The second derivative of $D1_{sin}(\alpha)$ is $D1''_{sin}(\alpha) = -\sin\alpha \leq 0$ for $\alpha \in [0, \frac{\pi}{2}]$, so by Proposition 2, $D1_{sin}(\alpha)$ is concave on $(0, \frac{\pi}{2})$, and by Proposition 1, the triangle inequality holds.

To make the distance measure more linear as a function of the angle we could try to replace the sine by the tangent function. Of course, the tangent has a singularity at $\frac{\pi}{2}$ where we would like to have the distance equal to 1. This is easily solved by taking a half of the angle:

$$D1_{tan}(x,y) = \tan\frac{\angle(x,y)}{2} = \sqrt{\frac{1-\cos(x,y)}{1+\cos(x,y)}} \qquad \in [0,1].$$

But $D1_{tan}(x,y)$ does not fulfill the conditions proposed in (66). (See appendix A.2 for proof.)

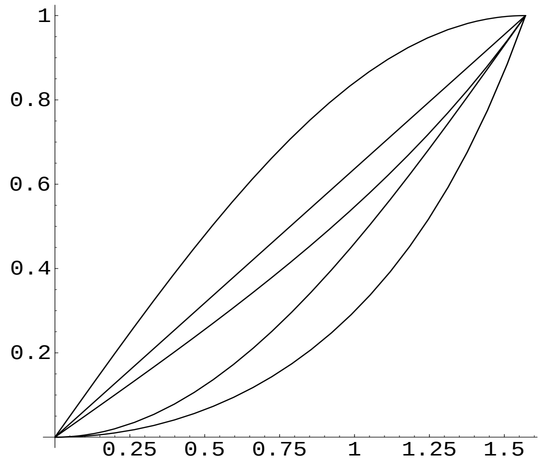

Figure 27: Measures as functions of the angle $\alpha = \angle(x,y)$. From above: $D1_{sin}(x,y)$, the linear function $g(\alpha) = \frac{2}{\pi}\alpha$, $D1_{tan}(x,y)$, $D1_{cos}(x,y)$ and $D2_{cos}(x,y)$.

Knowing that the concavity of the function used as a measure is decisive we could choose a concave part of the tangent function as follows.

$$D2_{tan}(x,y) = \tan\left(\frac{\angle(x,y)}{2} - \frac{\pi}{4}\right) + 1 \tag{76}$$
$$= \frac{2\sqrt{1-\cos(x,y)}}{\sqrt{1+\cos(x,y)} + \sqrt{1-\cos(x,y)}} \quad \in [0,1].$$

This measure fulfills the requirements (66) as is proved in appendix A.2. However, it is rather complicated computationally.

In a similar way as the tangent was scaled by taking a half of the angle one could try to make the sine a better approximation to the linear function $g(\alpha) = \frac{2}{\pi}\alpha$ by halving the argument $\angle(x,y)$ (see figure 28):

$$D2_{sin}(x,y) = \sqrt{2}\sin\left(\frac{\angle(x,y)}{2}\right) \tag{77}$$

$$= \sqrt{2}\sqrt{\frac{1-\cos\angle(x,y)}{2}} = \sqrt{1-\cos\angle(x,y)} = \sqrt{D1_{cos}}\ ,$$

which turns out to be the square root of the first of the previously tried cosine based distance measure $D1_{cos}$. The conditions (66) hold for $D2_{sin}(x,y)$.(See appendix A.2 for details.)

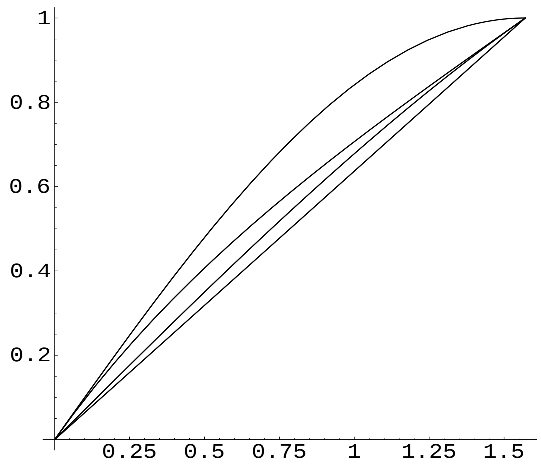

Figure 28: Measures as functions of the angle $\alpha = \angle(x,y)$. From above: $D1_{sin}(x,y)$, $D2_{tan}(x,y)$, $D2_{sin}(x,y)$ and the linear function $g(\alpha) = \frac{2}{\pi}\alpha$.

8.3.3 Measures that Allow Negative Weights

So far the considerations of suitable distance measures for clustering have been restricted to profiles with positive weights only. Let us make a brief remark of measures that allow negative weights as well. First of all, note that those measure candidates that failed to meet the requirements for positive profiles fail here as well, since this is just an extension and includes the previously handled positive case. The successful ones in the positive case were

$$D1_{sin}(x,y) = \sin(x,y) \qquad \text{and} \qquad D2_{sin}(x,y) = \sqrt{2}\sin\left(\frac{\angle(x,y)}{2}\right)$$

Allowing negative weights extends the range of possible angles from $[0, \frac{\pi}{2}]$ to $[0, \pi]$, and on that range the sine is no longer monotonic. The equivalent variants for this range are

$$D3_{sin}(x,y) = \sin\left(\frac{\angle(x,y)}{2}\right) \qquad \text{and} \qquad D4_{sin}(x,y) = \sqrt{2}\sin\left(\frac{\angle(x,y)}{4}\right)$$

The conditions (66) hold for both measures $D3_{sin}$ and $D4_{sin}$. The proofs are very similar to those of $D1_{sin}$ and $D2_{sin}$ and they can be found in appendix A.2.

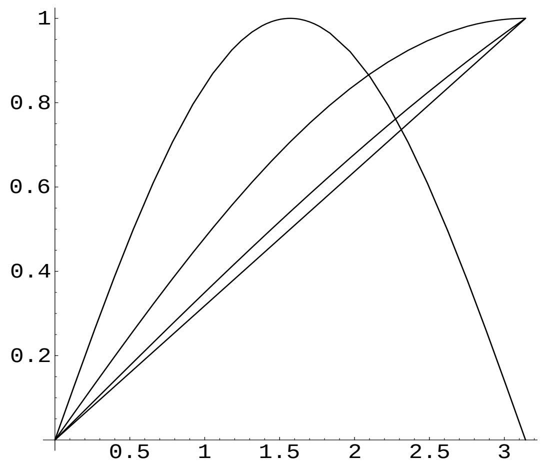

Figure 29: Measures as functions of the angle $\alpha = \angle(x,y)$ of range $[0,\pi]$. From above: $D1_{sin}(x,y)$, $D3_{sin}(x,y)$, $D4_{sin}(x,y)$ and the linear function $g(\alpha) = \frac{1}{\pi}\alpha$.

9 Testing of Methods

9.1 Tests with Clustering User Profiles

As real document metadata and logs of actual users have not yet been available, we have so far used as a testbed a collection of news flashes. It is owned by STT, the Finnish News Agency, and it dates at a period of time in 1995. We have made a concept hierarchy for the content description of these news flashes. This model is by no means comprehensive but only an outline for preliminary testing of the methods. The hierarchy is introduced in table 3.

<table>
<tr><th colspan="4">Concept Hierarchy used with test material</th></tr>
<tr><td rowspan="5">Accidents</td><td colspan="3">Drownings
Disasters
Fires</td></tr>
<tr><td rowspan="3">Damages and Injuries</td><td rowspan="2">Deaths and Injuries</td><td>Injuries</td></tr>
<tr><td>Deaths</td></tr>
<tr><td colspan="2">Material Damages</td></tr>
<tr><td colspan="3">Radiation
Earthquakes
Environmental Damages
Other Accidents</td></tr>
<tr><td colspan="4">Science</td></tr>
<tr><td colspan="4">Sports</td></tr>
<tr><td>Environment</td><td colspan="3">Water System
Forest
Other Environmental Issues</td></tr>
<tr><td>Automobiles</td><td>Automobiles</td><td>Automobiles</td><td>Race Cars
Formula 1
Other Cars</td></tr>
</table>

Table 3: Concept hierarchy used with test material (continues on the next page)

<table>
<tr><th colspan="4">Concept Hierarchy, continued</th></tr>
<tr><td rowspan="6">Economy</td><td>Industries</td><td>Industries</td><td>Mining and Metallurgy
Commerce
Service Industry
Banking and Insurance
Agriculture and Forestry
Construction
Manufacturing</td></tr>
<tr><td rowspan="2">National Economy</td><td rowspan="2">National Economy</td><td>Indicators</td></tr>
<tr><td>Other National Economy Issues</td></tr>
<tr><td colspan="3">International Trade</td></tr>
<tr><td>Money and Stock Market</td><td colspan="2">Interest Rates and Indexes
Stock Market
EMU
Other Finance</td></tr>
<tr><td>Companies</td><td>Companies</td><td>Acquisitions and Mergers
Profit and Loss
Other Corporate Issues</td></tr>
<tr><td>Traffic</td><td colspan="3">Road Traffic
Water Traffic
Air Traffic
Railway Traffic</td></tr>
<tr><td>Society</td><td colspan="3">Education
Health Care
Social Services</td></tr>
<tr><td rowspan="4">Politics</td><td colspan="3">Domestic Politics</td></tr>
<tr><td rowspan="3">Foreign and International Politics</td><td rowspan="2">Alliances</td><td>EU</td></tr>
<tr><td>Other Alliances</td></tr>
<tr><td colspan="2">Other Foreign and International Politics</td></tr>
</table>

There were 319 documents altogether in the document collection and most of them were somehow related to either economy or accidents. The documents were assigned a fuzzy classification to the nodes of the hierarchy as described in chapter 3.

We used document metadata descriptions as user profiles in the lack of real ones to see how the clustering methods for user profiles might work. The most convenient way to group profiles is by means of the representation hierarchy, i.e., at any level of the hierarchy the greatest weight determines the classification of the profile. This is no longer clustering as we defined it in chapter 7 because the classes are fixed beforehand and the profiles are just classified to these preset classes. Accordingly, this method cannot find any new groupings.

By applying the Maximin-Distance clustering method introduced in chapter 7.2 we tried to cluster documents into sets using various distance measures between "profiles". Since the sine based measures fulfilled the requirements we set for measures used in clustering (66), a few variants of the sine were tested. Moreover, to see if the theoretical drawbacks make any difference in practice, the 1-norm based measure was also tested.

Since no "right" answers exist, the results could not be compared to the "right" clusterings. Therefore, analyzing of the results was plain manual work. When we had as the end condition the number of clusters fixed to six, the resulting clusters were very similar independent of the distance measure or the starting point. These clusters can be described with the concepts of the uppermost level of the hierarchy: Accidents, Economy, Politics, Traffic, Sports, and Environment.

When the number of clusters was increased, the effect of the distance measure was hard to distinguish from the effect of the choice of starting point. Because of the amount of manual work, more than a few starting points were not feasible for thorough testing with different distance measures.

The end conditions based on distances were unpredictable; it might happen that the end condition was never met and the clustering algorithm stopped only after all the profiles were chosen as centroids of the clusters.

9.2 Planned Further Testing

The most interesting tests are still ahead of us, namely tests with real documents and real users with their real interests. However, the tests have already been designed and can be introduced here.

9.2.1 Available Data and Limitations

The test data consists of two parts: documents of an electronic newspaper Kauppalehti Online with their manually added metadata and logs of user behavior. For obvious security reasons there is no connection to the identities of the real users. The metadata model used in producing the test material includes two hierarchical dimensions of metadata, *Subject* and *Location*, one integer valued dimension *Priority*, and two Boolean vector dimensions, *Industry* and *Company*. There are 3 different possible *views* to the documents, one that shows n newest documents, one that shows all documents of the previous day, and one view with all the documents published so far during the current day. A user log contains customer data, e.g., which document each user opened and at what time, and which view he had to the documents. The view helps us to find out, which documents the user had to choose from.

Since we cannot know beforehand which of the dimensions are the most significant or how to combine the dimensions in a sensible way, we shall first test each dimension separately. Only after this we can choose some potentially good ways to combine information from various dimensions. All the possible combinations of more than one dimension would lead to $\sum_{k=2}^{5} \binom{5}{k} = 26$ different combinations, so some selection must be done.

We need an objective measure of the performance of the filtering of documents. The starting point is to compare the documents chosen by the user with the list of documents ranked in the order of increasing distance. In principle, the documents on this ranked list must be exactly the ones that were available to the user at that particular time. The list of available documents evolves all the time, so we need some kind of discretization of time. In practice, we try to divide the user's behavior into separate sessions and assume that those documents that were published before the starting time of the session were available to him. The 3 different possible views to the documents also confuse things, making 3 different possible lists of documents available at the same moment. This affects the set of documents among which the user had to make his decisions.

We have not yet tried on real users, whether the learning process has anything to do with the documents the user would like to have in the future. Thus, we start by testing the main idea, namely, whether these dimensions of metadata seem to capture what is essential in a user's interests, provided that the algorithm has all the time needed for learning the profile. Forgetting must be applied only when it is clear that the profile has learned something about the user's interests in the first place.

9.2.2 Designed Performance Scores

The following performance scores can be applied separately to each dimension of metadata, each with their own distance measures and user representations.

A score based on the ranks of documents

The available documents can be sorted by a distance measure producing a ranking list. Using this list we can base the performance score on the ranks of the documents chosen by the user. If there are M documents available to the user and the user chooses to open k of them, the best possible (minimal) sum of the ranks is

$$best_{rank} = \sum_{i=1}^{k} i = \frac{k(k+1)}{2} \ ,$$

and the worst possible (maximal) sum of ranks is

$$worst_{rank} = \sum_{i=M-k+1}^{M} i = \sum_{j=1}^{k} (M-k+j) = k(M-k) + \frac{k(k+1)}{2} \quad .$$

The score can be defined in the following way

$$score_{rank} = \frac{worst_{rank} - \text{sum of ranks}}{worst_{rank} - best_{rank}} \quad . \tag{78}$$

The best possible set of k documents gives a 100 % score and the worst possible set of k documents gives a 0 % score.

It may, of course, happen that many documents share the same distance. If their equality in the score is required, the ranks can be assigned in any way desired but the best possible sum of ranks and the worst possible sum of ranks must then be computed explicitly from the ranks.

A score based on the distances to the user profile

We are not only interested in the ordering of the documents but also in the distances the chosen documents have to the user profile. Therefore, a score based on the distances is introduced here. Let us denote by x the sum of distances of the chosen documents, i.e., $x = \sum_{i \in CHOSEN} \mathcal{D}(u, doc_i)$, and by $best_{dist}$ the best possible sum of k distances, $best_{dist} = \sum_{i=1}^{k} \mathcal{D}(u, doc_i)$, and by $worst_{dist}$ the worst possible sum of k distances, $worst_{dist} = \sum_{i=M-k+1}^{M} \mathcal{D}(u, doc_i)$. We could proceed analogously to the score for ranks and get the following score candidate

$$score_{distCand} = \frac{worst_{dist} - x}{worst_{dist} - best_{dist}} \tag{79}$$

that gives 100 % in the best case and 0 % in the worst case. Above, we have assumed that $best_{dist} < worst_{dist}$. If they are equal, all the distances in the given set are equal and any choice the user made would result in the same performance score. It is a matter of definition how to deal with such unlikely situations.

The above score (79) has the disadvantage that it only considers the k best and k worst distances and ignores the distribution of the distances between the extreme values. What is more important to know is, e.g., whether the mean of the chosen distances is better or worse than the mean of all the distances.

If

$$\frac{1}{k} \sum_{i \in CHOSEN} \mathcal{D}(u, doc_i) \leq \frac{1}{M} \sum_{all} \mathcal{D}(u, doc_i) \ ,$$

the performance is better than average and if

$$\frac{1}{k} \sum_{i \in CHOSEN} \mathcal{D}(u, doc_i) > \frac{1}{M} \sum_{all} \mathcal{D}(u, doc_i) \ ,$$

the performance is worse than average.

Let us denote the sum of all distances by A, i.e., $A = \sum_{all} \mathcal{D}(u, doc_i)$. Now, we can use the following conditions to determine the performance score

$$\begin{cases} score_{dist} = 0 & \text{if} \quad x = worst_{dist} \\ score_{dist} = 1 & \text{if} \quad x = best_{dist} \\ score_{dist} = 0.5 & \text{if} \quad x/k = A/M. \end{cases}$$

The simplest way is to make the score piecewise linear as below (see figure 30 as well)

$$score_{dist} = \begin{cases} \frac{1}{2}\left(1 + \frac{A/M - x/k}{A/M - best_{dist}/k}\right) & \text{if} \quad \frac{x}{k} \leq \frac{A}{M} \\ \frac{1}{2}\left(\frac{worst_{dist}/k - x/k}{worst_{dist}/k - A/M}\right) & \text{if} \quad \frac{x}{k} > \frac{A}{M}. \end{cases} \tag{80}$$

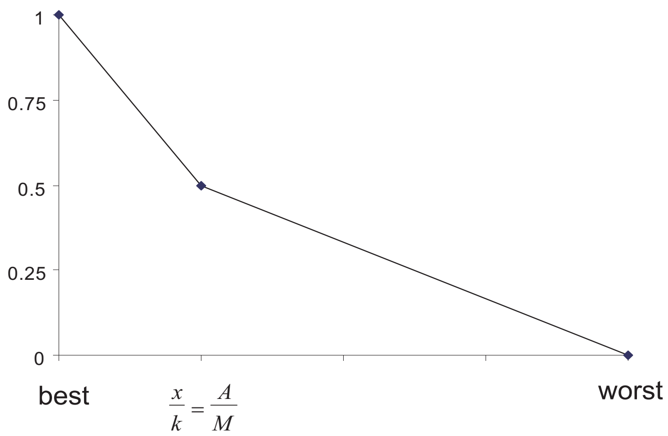

Figure 30: A score that utilizes distances (80). It is a piecewise linear function of the sum of k distances.

The next example shows the difference between the rank based score, $score_{rank}$ (eq. (78)), and the distance based score, $score_{dist}$ (eq. (80)).

Example

Let there be 20 documents available to the user ($M = 20$) and let the distances to the profile be as in table 4.

document	distance	document	distance
doc1	0.0	doc11	0.11
doc2	0.11	doc12	0.11
doc3	0.11	doc13	0.11
doc4	0.11	doc14	1.0
doc5	0.11	doc15	1.0
doc6	0.11	doc16	1.0
doc7	0.11	doc17	1.0
doc8	0.11	doc18	1.0
doc9	0.11	doc19	1.0
doc10	0.11	doc20	1.0

Table 4: Example data for calculating performance scores.

Suppose the user chooses four documents ($k = 4$) that happen to have ranks 10, 11, 12 and 13. The sum of the ranks is $10+11+12+13 = 46$, $worst_{rank} = 74$ and $worst_{rank} - best_{rank} = 64$, so the score based on the ranks is $score_{rank} = \frac{74-46}{64} \approx 43.8$ %. As we can see, the performance is somewhat less than average, if we consider only the ranking.

If, on the other hand, we measure the performance using distances, we have $best_{dist} = 0 + 0.11 + 0.11 + 0.11 = 0.33$ and $worst_{dist} = 1 + 1 + 1 + 1 = 4$. The sum of all distances in the list is $A = 8.32$ and the sum of the distances of the chosen documents is $x = 0.11 + 0.11 + 0.11 + 0.11 = 0.44$. So, the score based on distances is

$$score_{dist} = \frac{1}{2}\left(1 + \frac{4 \cdot 8.32 - 20 \cdot 0.44}{4 \cdot 8.32 - 20 \cdot 0.33}\right) \approx 95.9\ \%,$$

which indicates excellent performance. (See figure 31.)

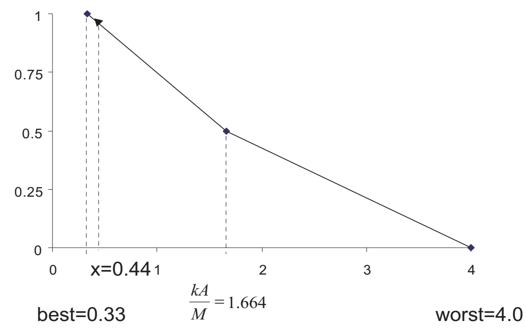

Figure 31: Distance based score (eq. (80)) for the set of documents in table 4.

9.2.3 Combining Scores of Various Sessions and of Various Users

The scores introduced in section 9.2.2 can be applied to each session of a user. A session is a sequence of documents the user opened in such a time interval that we can assume that he had the same set of documents to choose from. To get only one score based on ranks and one based on distances for each dimension of metadata for a user, we have to combine information of various sessions. This can be done in many ways, for example the average of session scores or a weighted average of session scores are possible. Weighting could depend on the number of documents the user opened at each session.

Once we have two scores per user, one based on the rankings, the other based on distances, we could try to combine information of various users to get a score for the system's overall performance. We intend to plot the distribution of users as a function of each score and to conclude some statistical parameters.

9.2.4 Learning Parameters and Amount of Training Data

The overall performance discussed above applies still only for one setting of learning parameters and for a fixed amount of training data. The training data consists of documents the user has chosen before the session. Obviously, we have to vary the parameters of the learning method and the amount of training data used to construct the profile, in order to get comprehensive results.

9.2.5 Tests with Users

When we have adjusted the parameters of the learning method and distance measure using the results of the tests introduced above, we can move on to testing the real recommendation system where documents are recommended according to the distances. Then the users can provide the system with explicit feedback. Therefore, we will be able to distinguish between positive and negative feedback, which was not the case with implicit feedback. With real users, testing of social information filtering techniques discussed in section 7.1 will be possible as well. At this phase the most important thing to be examined is, whether the users find the recommendations useful.

10 Open Issues and Future Work

Future testing will show whether the adopted methods are appropriate and if so, how the parameters should be selected for real use.

10.1 Comparisons with Other Methodologies

It would also be exciting to see, whether either one gives significantly better results for text documents, the basic vector space model or the metadata attached manually to the documents.

Even the vector space model with this suggested asymmetric distance measure could be worth examining.

10.2 Automatic Learning of Metadata from Data

It is obvious that in the long run the production of metadata cannot be counted solely on human resources. As the flow of new information increases at least some kind of assisting tools are needed. In our prototype there exists extensibility for such an assisting system. So far no advanced methods have been developed for a computer to infer the correct metadata from the actual data. There is a lot of research going on in that area but it is mostly limited to one language and plain text — not to talk about having pictures, audio and video under the same representation.

The use of structured document formats, which have a standard way of expressing the author, date, title and other such mechanical metadata, allows these to be easily extracted by computer. Hence, they need no natural language processing unlike the actual content.

Artificial intelligence techniques lend themselves to automated attaching of metadata to textual data provided that there is a sufficient amount of training data. So called supervised learning methods build their model of the data solely on examples of the form: (data, metadata). Of course, if this set of examples is biased or not representative, the result of learning is biased as well.

A statistical approach is to use Bayesian learning and the use of terms and their frequencies as features [1] (see appendix A.3). The needed probabilities are then inferred from statistical results for the set of examples. This could be an assisting tool in the production of metadata even though it only covers textual material in one language at a time, for example Finnish.

An important remark is that not all languages behave like English where you can discard prepositions and certain word endings to get the stems of almost all the words correct. In Finnish most words are inflected and can literally have thousands of different appearances, so a lookup table method is not a feasible solution. Fortunately, there has been a lot of research done by linguists on analyzing sentences in Finnish automatically and extracting the stems from text if needed. This requires special tools available commercially from Lingsoft Inc. Use of these stemming tools costs both money and time and still we only have a list of words appearing in the text — a situation that is taken as the starting point in many methods. The work has to be done separately for texts in different languages. The training data also has to be available separately in each language for the supervised learning method.

11 Conclusions

In order to filter documents according to user interests a common representation model for both the document content and the user profile is needed. The approach used in this thesis is to use metadata, which guarantees potential to a versatile representation.

Although expressed with the same representation model, the semantics of a profile is slightly different from the semantics of a document representation (metadata). Traditionally, symmetric distance measures have been used for their simplicity. Unfortunately, they cannot make the semantic distinction between a document and a profile. Thus, the distance measure between them should be asymmetric to capture individual parts of the overall user profile.

On the other hand, the objective of the learning method is to capture the profile as a whole instead of individual parts. This fundamental difference between the goals causes problems in the convergence of the distance.

Both content based and social information filtering can be utilized with the same profile information. Only different distance measures between two profiles and between a profile and a document must be chosen. The profiles can also be grouped into clusters of similar users. Again, the distance measure for clustering must be tailored for the purpose. In social filtering all distances are measured to one point (profile), which gives a lot of freedom in choosing the similarity measure, whereas in clustering symmetry and satisfaction of the triangle inequality are required.

A summary of the suggested distance measures for each purpose is presented in appendix A.4.

A Appendix

A.1 Compatibility of Learning Method and Distance Measures

A.1.1 Negative Feedback Increases Coarse Measure

Proof for the simple case:

Let us divide the indices into three distinct sets

$$I_p = \{i \mid x_i > 0\} = \{i \mid u_i > 0\}$$

$$I_0 = \{i \mid x_i = 0\} = \{i \mid u_i = 0\}$$

$$I_q = \{i \mid x_i < 0\} = \{i \mid u_i < 0\}.$$

Similarly, we denote the corresponding sets for the resulting profile $\mathbf{x}^*$ by I_p^*, I_0^* and I_q^*. It is easy to see that in the case of negative feedback, where $\lambda \leq -1$, the following is true:

$$i \in I_q \quad \Rightarrow \quad x_i^* = \frac{x_i}{f} + \lambda d_i < 0 \qquad \Rightarrow \quad i \in I_q^*$$

$$i \in I_0 \quad \Rightarrow \quad x_i^* = \lambda d_i \leq 0 \qquad \Rightarrow \quad i \in \left(I_q^* \cup I_0^*\right)$$

$$i \in I_p \quad \Rightarrow \quad x_i^* = \frac{x_i}{f} + \lambda d_i > \lambda d_i \geq \lambda \qquad \Rightarrow \quad i \in \left(I_p^* \cup I_0^* \cup I_q^*\right) \ .$$

Let us use the most general coarse measure

$$D_c(\mathbf{u}, \mathbf{d}) = \frac{1}{q}\left(p \sum_{I_0} d_i + q \sum_{I_q} d_i\right) \tag{81}$$

in the proof. Note also, that we have defined $q \geq p$.

Before feedback the coarse distance measure between a normalized profile $\mathbf{u}$ and a document $\mathbf{d}$ is $D_c(\mathbf{u}, \mathbf{d})$ as defined in equation (81). After the feedback the coarse distance measure between the new profile $\mathbf{u}^*$ and document $\mathbf{d}$ is

$$D_c(\mathbf{u}^*, \mathbf{d}) = \frac{1}{q}\left(p \sum_{I_0^*} d_i + q \sum_{I_q^*} d_i\right)$$

$$
\begin{aligned}
&= \frac{p}{q}\left(\sum_{I_0^*} d_i + \sum_{I_q^*} d_i\right) + \left(1 - \frac{p}{q}\right)\sum_{I_q^*} d_i \\
&= \frac{p}{q}\left(1 - \sum_{I_p^*} d_i\right) + \left(1 - \frac{p}{q}\right)\sum_{I_q^*} d_i \\
&\geq \frac{p}{q}\left(1 - \sum_{I_p} d_i\right) + \left(1 - \frac{p}{q}\right)\sum_{I_q} d_i \\
&= \frac{p}{q}\left(\sum_{I_0} d_i + \sum_{I_q} d_i\right) + \left(1 - \frac{p}{q}\right)\sum_{I_q} d_i \\
&= \frac{1}{q}\left(p\sum_{I_0} d_i + q\sum_{I_q} d_i\right) \\
&= D_c(\mathbf{u}, \mathbf{d}) \ ,
\end{aligned}
$$

where the facts

$$\sum_{I_p^*} d_i \leq \sum_{I_p} d_i \qquad \text{and} \qquad \sum_{I_q^*} d_i \geq \sum_{I_q} d_i$$

have been used, along with the remark $\left(1 - \frac{p}{q}\right) \geq 0$. $\square$

Proof for the case with an ε-threshold:

In this case only those nonzero weights whose absolute value exceeds certain ε are regarded as hit weights when calculating the distance between the user profile and a document. We can also assume guaranteed that the weights in the documents are always either greater than ε or exactly zero.

The indices are divided into three distinct sets according to the weights of the profile

$$
\begin{aligned}
I_p =& \{i \mid x_i \geq \varepsilon\} \\
I_0 =& \{i \mid -\varepsilon < x_i < \varepsilon\} \\
I_q =& \{i \mid x_i \leq -\varepsilon\}.
\end{aligned}
$$

Similarly, denote the corresponding sets for the resulting profile $\mathbf{x}^*$ by I_p^*, I_0^* and I_q^*.

In the case of negative feedback ($\lambda \leq -1$) the following holds:

$$
\begin{array}{lll}
(i \in I_q) \wedge (d_i \geq \varepsilon) & \Rightarrow & x_i^* = \dfrac{x_i}{f} + \lambda d_i \leq -\dfrac{\varepsilon}{f} + \lambda\varepsilon \leq -\left(1 + \dfrac{1}{f}\right)\varepsilon \leq -\varepsilon \\
 & \Rightarrow & i \in I_q^* \\
(i \in I_q) \wedge (d_i = 0) & \Rightarrow & x_i^* = \dfrac{x_i}{f} \leq -\dfrac{\varepsilon}{f} < 0 < \varepsilon \\
 & \Rightarrow & i \in \left(I_q^* \cup I_0^*\right) \\
 & & \\
i \in I_0 & \Rightarrow & x_i^* < \dfrac{\varepsilon}{f} + \lambda d_i \leq \dfrac{\varepsilon}{f} \leq \varepsilon \\
 & \Rightarrow & i \in \left(I_q^* \cup I_0^*\right) \\
 & & \\
i \in I_p & \Rightarrow & x_i^* \geq \dfrac{\varepsilon}{f} + \lambda d_i \geq \dfrac{\varepsilon}{f} - \lambda \\
 & \Rightarrow & i \in \left(I_p^* \cup I_0^* \cup I_q^*\right) \; .
\end{array}
$$

It is easy to see that

$$\sum_{I_p^*} d_i \leq \sum_{I_p} d_i \; . \tag{82}$$

Let us divide the indices of I_q further as follows

$$\sum_{I_q} d_i = \sum_{I_q \cap I_q^*} d_i + \sum_{I_q \cap I_0^*} d_i = \sum_{I_q \cap I_q^*} d_i \; , \tag{83}$$

where the fact $i \in (I_0^* \cap I_q) \implies d_i = 0$ has been used. From (83) we get

$$\sum_{I_q^*} d_i \geq \sum_{I_q} d_i \; . \tag{84}$$

Using inequalities (82) and (84) the proof for the simple case applies here as well. $\square$

A.1.2 Convergence of Learning Method

We have already proved convergence in the case with forgetting, i.e., $f > 1$. Next, we continue with the case without forgetting, i.e., $f = 1$.

After N iterations of the learning method, the profile is

$$\mathbf{x}_N = \mathbf{x}_0 + N\lambda\mathbf{d} \ .$$

If N is large enough, either the positive or the negative profile has vanished. Let us denote the masses of the positive or the negative part of the initial profile $\mathbf{x}_0$ by

$$M_{0+} = \sum_{x_{0i}>0} x_{0i} \quad \text{and} \quad M_{0-} = \sum_{x_{0i}<0} |x_{0i}| \ .$$

The profile $\mathbf{x}_N$ has either only positive or only negative weights, so the mass of the profile is

$$\begin{aligned} M_N &= \sum_i |x_{0i} + N\lambda d_i| = \left| \sum_i (x_{0i} + N\lambda d_i) \right| \\ &= \left| \sum_i x_{0i} + N\lambda \sum_i d_i \right| = |M_{0+} - M_{0-} + N\lambda| \ . \end{aligned}$$

In the case of positive feedback we get

$$\begin{aligned} \mathbf{u} &= \lim_{N\to\infty} \mathbf{u}_N = \lim_{N\to\infty} \frac{\mathbf{x}_N}{M_N} \\ &= \lim_{N\to\infty} \frac{1}{M_{0+} - M_{0-} + N\lambda} \mathbf{x}_0 + \lim_{N\to\infty} \frac{N\lambda}{M_{0+} - M_{0-} + N\lambda} \mathbf{d} \\ &= \mathbf{d} \ . \end{aligned}$$

In the case of negative feedback the profile will be

$$\begin{aligned} \mathbf{u} &= \lim_{N\to\infty} \mathbf{u}_N = \lim_{N\to\infty} \frac{\mathbf{x}_N}{M_N} \\ &= \lim_{N\to\infty} \frac{1}{M_{0-} - M_{0+} - N\lambda} \mathbf{x}_0 + \lim_{N\to\infty} \frac{N\lambda}{M_{0-} - M_{0+} - N\lambda} \mathbf{d} \\ &= -\mathbf{d} \ . \end{aligned}$$

So, the profile $\mathbf{u}$ converges to document $\mathbf{d}$, i.e. $\mathbf{u} = \mathbf{d}$, in the case of positive feedback and to $-\mathbf{d}$, i.e. $\mathbf{u} = -\mathbf{d}$, in the case of negative feedback. □

A.1.3 Compatibility of Learning Method and Symmetric Cosine Based Measure

The symmetric cosine based measure for an unnormalized profile $\mathbf{x}$ and a document $\mathbf{d}$ was introduced in section 5.2.2 as

$$D_f(\mathbf{x},\mathbf{d}) = \frac{1}{2}\left(1 - \frac{\sum_{i\in I} x_i \cdot d_i}{\sqrt{\sum_{i\in I} x_i^2}\sqrt{\sum_{i\in I} d_i^2}}\right)$$
$$= \frac{1}{2}\left(1 - \frac{(\mathbf{x},\mathbf{d})}{\sqrt{(\mathbf{x},\mathbf{x})}\sqrt{(\mathbf{d},\mathbf{d})}}\right) , \qquad (85)$$

the index set I containing all the indices of $\mathbf{x}$ and $\mathbf{d}$. Now, we shall prove that the learning method decreasing the distance $D_f(\mathbf{x},\mathbf{d})$ in the case of positive feedback and increasing it in the case of negative feedback. As usual,

$$(\mathbf{x},\mathbf{d}) = \sum_{i\in I} x_i \cdot d_i$$

is an inner product, so the Cauchy-Schwarz-Bunjakowski inequality (CSB) holds, i.e.

$$|(\mathbf{x},\mathbf{d})| \leq \sqrt{(\mathbf{x},\mathbf{x})}\sqrt{(\mathbf{d},\mathbf{d})} .$$

Moreover, let us denote $a = (\mathbf{x},\mathbf{x})$, $b = (\mathbf{x},\mathbf{d})$, and $c = (\mathbf{d},\mathbf{d})$ for short. With this notation the CSB is $|b| \leq \sqrt{a}\sqrt{c}$, which yields $b^2 \leq ac$. For the updated profile $\mathbf{x}^*$

$$(\mathbf{x}^*,\mathbf{x}^*) = \sum_I (x_i^*)^2 = \sum_I \left(\frac{x_i}{f} + \lambda d_i\right)^2 = \frac{a}{f^2} + \lambda^2 c + \frac{2\lambda}{f} b$$

and

$$(\mathbf{x}^*,\mathbf{d}^*) = \sum_I x_i^* \cdot d_i = \sum_I \left(\frac{x_i \cdot d_i}{f} + \lambda d_i^2\right) = \frac{b}{f} + \lambda c$$

Since $\sqrt{c} > 0$ it suffices to prove that

$$Q = 2\sqrt{c}\,(D_f(\mathbf{x},\mathbf{d}) - D_f(\mathbf{x}^*,\mathbf{d})) \begin{cases} \geq 0 , & \text{if} \quad \lambda > 0 , \\ \leq 0 , & \text{if} \quad \lambda < 0 . \end{cases}$$

A straightforward substitution into equation (85) yields

$$Q = -\frac{b}{\sqrt{a}} + \frac{b + \lambda f c}{\sqrt{a + (\lambda f)^2 c + 2\lambda f b}} . \qquad (86)$$

It can be easily seen that the sign of a general expression of the form $\frac{p}{\sqrt{q}} + \frac{r}{\sqrt{s}}$ is

$$\mathrm{sign}\left(\frac{p}{\sqrt{q}} + \frac{r}{\sqrt{s}}\right) = \begin{cases} \mathrm{sign}\left(\frac{p^2}{q} - \frac{r^2}{s}\right)\mathrm{sign}(p)\ , & \text{if } \ \mathrm{sign}(pr) = -1 \\ \mathrm{sign}(p)\ , & \text{if } \ \mathrm{sign}(pr) = +1 \text{ or } \\ & \mathrm{sign}(r) = 0 \\ \mathrm{sign}(r)\ , & \text{if } \ \mathrm{sign}(p) = 0\ . \end{cases} \tag{87}$$

In our case $p = b + \lambda f c$, $\quad q = a + 2\lambda f b + c(\lambda f)^2$, $\quad r = -b$, and $\quad s = a$.

Using the fact $\quad \mathrm{sign}\left(\frac{f}{aq}\right) = +1$, $\quad$ we get

$$\begin{aligned} \mathrm{sign}\left(\frac{p^2}{q} - \frac{r^2}{s}\right) &= \mathrm{sign}\left(\frac{\lambda f}{aq}(ac - b^2)(2b + \lambda f c)\right) \\ &= \mathrm{sign}(\lambda)\mathrm{sign}(ac - b^2)\mathrm{sign}(2b + \lambda f c)\ . \end{aligned}$$

So, substituting into (87) we get

$$\mathrm{sign}(Q) = \begin{cases} \mathrm{sign}(\lambda)\mathrm{sign}(2b + \lambda f c)\cdot & \\ \quad \cdot\mathrm{sign}(b + \lambda f c)\mathrm{sign}(ac - b^2)\ , & \text{if } \ \mathrm{sign}(b(b + \lambda f c)) = +1 \\ \mathrm{sign}(b + \lambda f c)\ , & \text{if } \ \mathrm{sign}(b(b + \lambda f c)) = -1 \text{ or } \\ & \mathrm{sign}(b) = 0 \\ -\mathrm{sign}(b)\ , & \text{if } \ \mathrm{sign}(b + \lambda f c) = 0\ . \end{cases} \tag{88}$$

In the special case where $ac = b^2$ and $\mathrm{sign}(b(b + \lambda f c)) = +1$, $\mathrm{sign}(Q)$ will be zero. Otherwise, by CSB $ac - b^2 > 0$ and $\mathrm{sign}(Q) = \mathrm{sign}(\lambda)$,

$$\mathrm{sign}(Q) = \begin{cases} 0\ , & \text{in a certain case} \\ \mathrm{sign}(\lambda)\ , & \text{otherwise.} \end{cases}$$

Thus, with this measure the distance decreases in the case of positive feedback and increases in the case of negative feedback. □

A.2 Proofs of Triangle Inequality for Various Measures

Proof for $D1_{cos}(x, y)$:

1. $x = y \implies \quad D1_{cos}(x, y) = 1 - \cos(x, y) = 0.$
2. The symmetry of $D1_{cos}(x, y)$ is trivial.
3. The triangle inequality does not hold for $D1_{cos}(x, y)$. **Proof:**
   Let us denote $\alpha = \angle(x, y)$. The second derivative of $D1_{cos}(\alpha)$ is $D1''_{cos}(\alpha) = \cos\alpha > 0$ for $\alpha \in (0, \frac{\pi}{2})$, so by Proposition 2, $D1_{cos}(\alpha)$ is strictly convex on $(0, \frac{\pi}{2})$, and by Proposition 1, the triangle inequality does not hold. $\square$

Proof for $D2_{cos}(x, y)$:

1. $x = y \implies \quad D2_{cos}(x, y) = \frac{2}{1+\cos(x,y)} - 1 = 0.$
2. The symmetry of $D2_{cos}(x, y)$ is trivial.
3. The triangle inequality does not hold for $D2_{cos}(\alpha)$. **Proof:**
   The second derivative of $D2_{cos}(\alpha)$ is $D2''_{cos}(\alpha) = 2\frac{2+\cos\alpha-\cos^2\alpha}{(\cos\alpha+1)^3} > 0$ for $\alpha \in (0, \frac{\pi}{2})$, so by Proposition 2, $D2_{cos}(\alpha)$ is strictly convex on $(0, \frac{\pi}{2})$, and by Proposition 1, the triangle inequality does not hold. $\square$

Proof for $D1_{tan}(x, y)$:

1. $x = y \implies \quad D1_{tan}(x, y) = \tan\frac{0}{2} = 0.$
2. Symmetry is trivial.
3. The triangle inequality does not hold. **Proof:**
   The second derivative of $D1_{tan}(\alpha)$ is $D1''_{tan}(\alpha) = \frac{1}{2\cos^2\frac{\alpha}{2}} > 0$ for $\alpha \in (0, \frac{\pi}{2})$, so $D_{tan}(\alpha)$ is strictly convex on $(0, \frac{\pi}{2})$, and therefore, (by Proposition 1) the triangle inequality does not hold. $\square$

Proof for $D2_{tan}(x,y)$:

1. $x = y \implies D2_{tan}(x,y) = \tan\left(\frac{0}{2} - \frac{\pi}{4}\right) + 1 = 0.$
2. The symmetry of $D2_{tan}(x,y)$ is trivial.
3. The triangle inequality holds for $D2_{tan}(x,y)$. **Proof:** The second derivative of $D2_{tan}(\alpha) = \tan\left(\frac{\alpha}{2} - \frac{\pi}{4}\right)$ is $D2''_{tan}(\alpha) = \frac{\tan\left(\frac{\alpha}{2}-\frac{\pi}{4}\right)}{\cos^2\left(\frac{\alpha}{2}-\frac{\pi}{4}\right)} \leq 0$ for $\alpha \in (0, \frac{\pi}{2})$, so $D2_{tan}(\alpha)$ is concave on $(0, \frac{\pi}{2})$, and therefore, the triangle inequality holds. $\square$

Proof for $D2_{sin}(x,y)$:

1. $x = y \implies D2_{sin}(x,y) = \sqrt{2}\sin\frac{0}{2} = 0.$
2. Symmetry is trivial.
3. The triangle inequality holds. **Proof:** The second derivative of $D2_{sin}(\alpha) = \sqrt{2}\sin\frac{\alpha}{2}$ is $D2''_{sin}(\alpha) = -\frac{1}{2\sqrt{2}}\sin\frac{\alpha}{2} \leq 0$ for $\alpha \in (0, \frac{\pi}{2})$, so $D2_{sin}(\alpha)$ is concave on $(0, \frac{\pi}{2})$, and therefore, the triangle inequality holds. $\square$

Proof for $D3_{sin}(x,y)$:

1. $x = y \implies D3_{sin}(x,y) = \sin\frac{0}{2} = 0.$
2. The symmetry of $D3_{sin}(x,y)$ is trivial.
3. The triangle inequality holds for $D3_{sin}(x,y)$. **Proof:** The second derivative of $D3_{sin}(\alpha) = \sin\frac{\alpha}{2}$ is $D3''_{sin}(\alpha) = -\frac{1}{4}\sin\frac{\alpha}{2} \leq 0$ for $\alpha \in (0, \pi)$, so $D3_{sin}(\alpha)$ is concave on $(0, \pi)$, and therefore, the triangle inequality holds. $\square$

Proof for $D4_{sin}(x,y)$:

1. $x = y \implies D4_{sin}(x,y) = \sqrt{2}\sin\frac{0}{4} = 0.$
2. The symmetry of $D4_{sin}(x,y)$ is trivial.
3. The triangle inequality holds for $D4_{sin}(x,y)$. **Proof:** The second derivative of $D4_{sin}(\alpha) = \sqrt{2}\sin\frac{\alpha}{4}$ is $D4''_{sin}(\alpha) = -\frac{1}{8\sqrt{2}}\sin\frac{\alpha}{4} \leq 0$ for $\alpha \in (0, \pi)$, so $D4_{sin}(\alpha)$ is concave on $(0, \pi)$, and therefore, the triangle inequality holds. $\square$

A.3 Bayesian Learning in Information Filtering

Let $\mathbf{x}$ be a feature vector indicating the presence and absence of features (e.g. terms). Let C denote an atomic event "the vector $\mathbf{x}$ belongs to the class C" and denote by $\neg C$ the event "the vector $\mathbf{x}$ does not belong to the class C". The events C and $\neg C$ are mutually exclusive and $P(C) + P(\neg C) = 1$. The Bayesian rule gives

$$P(C \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C)P(C)}{P(\mathbf{x})} = \omega P(\mathbf{x} \mid C)P(C) , \tag{89}$$

with

$$P(\mathbf{x}) = \frac{1}{\omega} = P(\mathbf{x} \mid C)P(C) + P(\mathbf{x} \mid \neg C)P(\neg C).$$

We can first ignore the normalizing factor ω and calculate $a = P(\mathbf{x} \mid C)P(C)$ and $b = P(\mathbf{x} \mid \neg C)P(\neg C)$ since

$$1 = P(C \mid \mathbf{x}) + P(\neg C \mid \mathbf{x}) = \omega a + \omega b ,$$

which implies $\omega = \frac{1}{a+b}$. As a result we get two probabilities $P(C \mid \mathbf{x})$ and $P(\neg C \mid \mathbf{x})$ and a logical inference could be that if $P(C \mid \mathbf{x})$ is greater, $\mathbf{x}$ belongs to class C and otherwise not. Note that we get these two probabilities to every class in our classification. One interpretation could be to allow $\mathbf{x}$ to belong to only one class, the one with the highest probability over all classes. Or one could set some threshold value to the probability and $\mathbf{x}$ would belong to those classes whose probability exceeded the threshold. When $P(C \mid \mathbf{x}) = P(\neg C \mid \mathbf{x})$ the choice has to be made arbitrarily.

Let us study in more detail how the probabilities are estimated from the training data. Let the number of training documents be N. Among these examples k_C documents belong to class C and, therefore, the probability is

$$P(C) = \frac{k_C}{N}$$

(Naturally $N - k_C$ documents belong to $\neg C$ and $P(\neg C) = 1 - \frac{k_C}{N}$.) The conditional probabilities are more cumbersome. Since we expect that an item $\mathbf{x}$ has not appeared in the training data there is no statistics for $\mathbf{x}$. Here the features enter into play. We express probabilities for each feature to appear in each class.

$$P(x_i = 1 \mid C) = \frac{k_{iC}}{k_C} \qquad \text{and} \qquad P(x_i = 0 \mid C) = \frac{k_C - k_{iC}}{k_C} = 1 - \frac{k_{iC}}{k_C},$$

where k_{iC} is the amount of documents that have feature $x_i = 1$ and belong to class C.

A rather strong assumption is made here: as long as we do not have any information about how the features depend on each other we assume them to be stochastically independent of each other, i.e.

$$P(\mathbf{x} \mid C) = \prod_i P(x_i \mid C) \tag{90}$$

Example:

Let the training data be the following set of five documents represented with four features ($N = 5, \quad T = 4$). Let there be three classes $C1$, $C2$ and $C3$ and we suppose that one document can belong to more than one class in this example.

$$\begin{cases} d_1 = (\ 1\ 1\ 0\ 1\ ) & \in C1 \\ d_2 = (\ 0\ 1\ 1\ 0\ ) & \in C2 \\ d_3 = (\ 0\ 0\ 0\ 1\ ) & \in C3 \\ d_4 = (\ 1\ 1\ 1\ 1\ ) & \in C1, C2, C3 \\ d_5 = (\ 0\ 1\ 1\ 1\ ) & \in C2, C3 \end{cases}$$

First of all, we compute the statistics for this set of documents, i.e., the number of documents k_C in each class C

$$\begin{cases} k_{C1} = 2 \\ k_{C2} = 3 \\ k_{C3} = 3 \end{cases}$$

and the number of documents that have the feature i present (i.e. $x_i = 1$) in class C, denoted by k_{iC}.

k_{iC}						
$i \setminus C$	$C1$	$\neg C1$	$C2$	$\neg C2$	$C3$	$\neg C3$
1	2	0	1	1	1	1
2	2	2	3	1	2	2
3	1	2	3	0	2	1
4	2	2	2	2	3	1

Table 5: The number of documents that have the feature i present ($x_i = 1$) in class C, denoted by k_{iC}, for each of the classes and their complements.

Now, we can take any feature vector $\mathbf{x}$ and calculate the probabilities for its belonging to each class. For example, let $\mathbf{x} = (\ 0\ 1\ 0\ 1\ )$.

$$P(C1 \mid \mathbf{x}) = 0 \qquad (\Longrightarrow \quad P(\neg C1 \mid \mathbf{x}) = 1) \qquad \text{since}$$

$$P(x_1 = 0 \mid C1) = 1 - P(x_1 = 1 \mid C1) = 1 - \frac{k_{1C1}}{k_{C1}} = 1 - \frac{2}{2} = 0$$

$$\text{and} \quad P(C2 \mid \mathbf{x}) = 0 \qquad (\implies \quad P(\neg C2 \mid \mathbf{x}) = 1) \qquad \text{since}$$

$$P(x_3 = 0 \mid C2) = 1 - P(x_3 = 1 \mid C2) = 1 - \frac{k_{3C2}}{k_{C2}} = 1 - \frac{3}{3} = 0.$$

$$\begin{aligned}
P(C3 \mid \mathbf{x}) &= \omega \cdot P(C3) \cdot \prod_{i=1}^{4} P(x_i \mid C3) \\
&= \omega \cdot \frac{k_{C3}}{N} \cdot P(x_1 = 0 \mid C3) \cdot P(x_2 = 1 \mid C3) \\
&\quad \cdot P(x_3 = 0 \mid C3) \cdot P(x_4 = 1 \mid C3) \\
&= \omega \cdot \frac{3}{5} \cdot \frac{k_{C3} - k_{1C3}}{k_{C3}} \cdot \frac{k_{2C3}}{k_{C3}} \cdot \frac{k_{C3} - k_{3C3}}{k_{C3}} \cdot \frac{k_{4C3}}{k_{C3}} \\
&= \frac{4}{45}\omega
\end{aligned}$$

$$\begin{aligned}
P(\neg C3 \mid \mathbf{x}) &= \omega \cdot P(\neg C3) \cdot \prod_{i=1}^{4} P(x_i \mid \neg C3) \\
&= \omega \cdot \frac{N - k_{C3}}{N} \cdot P(x_1 = 0 \mid \neg C3) \cdot P(x_2 = 1 \mid \neg C3) \\
&\quad \cdot P(x_3 = 0 \mid \neg C3) \cdot P(x_4 = 1 \mid \neg C3) \\
&= \omega \cdot \frac{2}{5} \cdot \frac{N - k_{C3} - k_{1\neg C3}}{N - k_{C3}} \cdot \frac{k_{2\neg C3}}{N - k_{C3}} \cdot \frac{N - k_{C3} - k_{3\neg C3}}{N - k_{C3}} \cdot \frac{k_{4\neg C3}}{N - k_{C3}} \\
&= \frac{1}{20}\omega
\end{aligned}$$

And since $P(C3 \mid \mathbf{x}) + P(\neg C3 \mid \mathbf{x}) = 1$ we get $\omega = 36/5$ and we have

$$\begin{cases} P(C3 \mid \mathbf{x}) & = \frac{36}{5} \cdot \frac{4}{45} \quad = \frac{16}{25} \\ P(\neg C3 \mid \mathbf{x}) & = \frac{36}{5} \cdot \frac{1}{20} \quad = \frac{9}{25} \end{cases}$$

A.4 Summary of Suggested Distance Measures

Content Based Filtering

In content based filtering where content based representations of documents are matched to a profile, we expect that the best choice is the combination below

- the coarse measure
  (eq. (2) in the case of positive profile or (4) if there is also a negative profile)
- combined with the fine measure D_{f1} (eq. (5))
- using the first method for combining the distances (eq. (14))
- and emphasizing the leaf level of the hierarchy in weighting
  (eq. (9) in the case of positive profile or (12) when there are also negative profiles).

Social Filtering with Content Based Profiles

In social filtering with content based profiles we expect that the best choice for comparing the profiles in order to find the similar ones is

- the measure $D(\mathbf{a}, \mathbf{b})$
  (eq. (65) introduced on page 50 in the case of positive profile or the same equation with the suggested modifications if there is also a negative profile)
- emphasizing the uppermost levels of the hierarchy in weighting
  (eq. (10) in the case of positive profile or with the same idea as in equation (12) but the weighting reversed, when there are also negative profiles).

If new areas of interest are searched for a user profile a, the actual filtering of documents is reduced to the profile–document comparison discussed above, as soon as the query profile $\mathbf{q}_a$ is constructed.

Pure Social Filtering

In pure social filtering, where the profiles are based only on ratings for items, we expect that the best choice for comparing the profiles in order to find the similar ones is

- the measure $\mathrm{sim}(a, x)$ (eq. (61)).

It can be used in the same way as usual in social filtering, i.e., by weighting the similar users' ratings with their similarity measures. The prediction of the rating for the profile is the weighted average of the similar users' ratings.

Clustering of Profiles

In clustering the user profiles into groups of similar users, we do not have a superior candidate for comparing the profiles, but the choice should be

- a symmetric distance measure that satisfies the triangle inequality (For positive profiles such measures are, e.g., $D1_{sin}$ (eq. (75)), $D2_{sin}$ (eq. (77)) or $D2_{tan}$ (eq. (76)). For profiles with negative weights also examples of such measures could be $D3_{sin}$ and $D4_{sin}$ on page 63.)
- and emphasizing the uppermost levels of the hierarchy in weighting. (This can be found in equation (10) in the case of positive profiles. If there are also negative profiles, only the leaf level should be considered and the hierarchy must be ignored to get the same weighting for the levels for all the profiles).

References

[1] van Rijsbergen, C.J. (1981). Information Retrieval. Butterworths.

[2] Salton, G. (1968). Automatic Information Organization and Retrieval. McGraw-Hill.

[3] Belkin, N.J., Croft, W.B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM, Dec. 1992, Vol. 35, No. 12.

[4] Davis, R., Shrobe, H., Szolovits, P. (1993). What is Knowledge Representation? AI Magazine, v. 14, #1.

[5] Sheth, B.(1994). A Learning Approach to Personalized Information Filtering, M.I.T.
http://agents.www.media.mit.edu/groups/agents/papers/newt-thesis/main.html

[6] Yan, T.W.,Garcia-Molina, H. (1994). Index Structures for Information Filtering under the Vector Space Model.
IEEE Conference on Data Engineering.

[7] Moukas, A., Zacharia, G. (1997). Evolving a Multi-agent Information Filtering Solution in Amalthaea.
Agents'97 Conference Proceedings, ACM.

[8] Balabanović, M. (1996). An Adaptive Recommendation Service.
Stanford University Digital Libraries Project Working Paper SIDL-WP-1996-0041.

[9] Foltz, P.W., Dumais, S. (1992). Personalized Information Delivery : An Analysis of Information Filtering Methods.
Communications of the ACM 35(12)
http://www-psych.nmsu.edu/~pfoltz/cacm/cacm.html

[10] Berry, M.W., Dumais S.T., O'Brien, G.W. (1995). Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4), 573-595.
http://superbook.bellcore.com/~remde/lsi/LSIpapers.html

[11] Savia, E. (1998). Latentti semanttinen indeksointi - esimerkkitapaus. Tietojenkäsittelyteorian erikoistyö, TKK.
http://smartpush.cs.hut.fi/pubdocs

[12] Bloedorn, E., Mani, I., MacMillan R. (1996).
Representational Issues in Machine Learning of User Profiles.
AAAI Spring Symposium on Machine Learning in Information Access.

[13] Koller, D., Sahami, M. (1997). Hierarchically Classifying Documents Using Very Few Words. International Conference on Machine Learning -97.

[14] Zimmermann, H.-J. (1986). Fuzzy Set Theory — and its Applications. Kluwer Publishing.

[15] Savia, E. (1998). Etäisyysmittojen vertailu elektronisessa täsmäjakeluympäristössä. Matematiikan erikoistyö, TKK. http://smartpush.cs.hut.fi/pubdocs

[16] Savia, E., Kurki, T., Jokela, S. (1998). Metadata Based Matching of Documents and User Profiles. Proceedings of Finnish Artificial Intelligence Conference. STeP'98. http://smartpush.cs.hut.fi/pubdocs

[17] Rajala, A.(1998). Improved Learning Method for Learning of User Profiles and Analysis of Former Methods. Matematiikan erikoistyö, TKK. http://smartpush.cs.hut.fi/pubdocs

[18] Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., Cohen, M.D. (1987). Intelligent Information-Sharing Systems. Communications of the ACM, May 1987, Vol. 30, no. 5.

[19] Shardanand, U., Maes, P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth". CHI'95: Human Factors in Computing Systems, Conference Proceedings.

[20] Kohonen, T. (1995). Self-Organizing Maps. Springer-Verlag.

[21] Shapiro, S.C. (ed.)(1992). Encyclopedia of Artificial Intelligence. John Wiley & Sons, Inc.

[22] Tou, J.T., Gonzales, R.C. (1974). Pattern Recognition Principles. Addison-Wesley.

[23] Rudin, Walter (1986). Real and Complex Analysis. McGraw-Hill.

[24] Savia, E. (1997). Adaptiivinen oppimismenetelmä käyttäjäprofiilien päivittämiseen. Matematiikan erikoistyö, TKK. http://smartpush.cs.hut.fi/pubdocs