Receipt Distance Metric

Author One

January 2020

1 Introduction

In this document we explain the concept of comparing receipts using categories and hierarchical data. The global idea is that we do not only want to compare receipts within each super-category (*intra*-super-category) but also between different super-categories (*inter*-super-category). The document first describes an example in Section 2 and then formalizes this in Section 3. The formalization will require some learning based on available data as explained in Section 4, optimizations to improve the learning speed are discussed in Section 5. We conclude this document with a more elaborate example that step-by-step covers all steps and optimizations proposed in this document in Section 6.

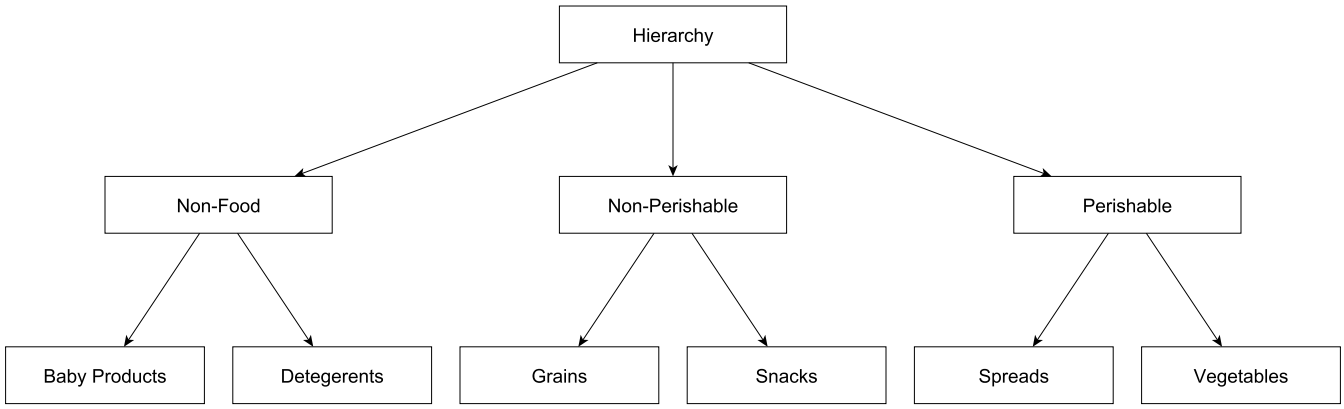

Figure 1: Example Hierarchy

2 Example

Figure 1 shows an example of a hierarchy. From the hierarchy we see that there are three *super-categories*, Non-Food, Non-Perishable, and Perishable. Each of these three super-categories has two *categories*. Non-Food has Baby Products and Detergent, Non-Perishable has Grains and Snacks, and Perishable has Spreads and Vegetables.

We want a way to compare two receipts. Tables 1 and 2 show two example receipts. Note that we only show which categories are bought; for this comparison we do not care about the price, quantity or products bought; just about the categories. We make the following observations. From the super-category Non-Food both receipts contain Baby Products, but only receipt 2 contains Detergent. From the super-category Non-Perishable both contain *a* category, but the specific category they contain differs. Finally, from the super-category Perishable receipt 1 contains nothing, whereas receipt 2 contains Spreads.

We make a total of four comparisons between these two receipts. The first three come from the super-categories. For the super-category Non-Food the two receipts are somewhat similar as they both contain Baby Products, but they are also somewhat different since only one of the receipts contains Detergent. For the super-category Non-Perishable the receipts are completely different: the contain no overlapping categories. The same holds for the super-category Perishable: there are no categories that are found in both receipts. Somewhat different and completely different are actually well defined; we return to this in Section 3. We call these comparisons *Non-Food comparison*, *Non-Perishable*

Baby Products
Detergent
Grains

Table 1: Example receipt 1

Baby Products
Snacks
Spreads

Table 2: Example receipt 2

comparison and *Perishable comparison* respectively. These comparisons are referred to as *intra-super-category* comparisons, because they make the comparison within a super-category.

The fourth comparison we make is a little trickier. We noted how both of the super-categories Non-Perishable and Perishable are completely different. There is however the fact that both receipt still bought *something* from Non-Perishable; this is not true for Perishable. We want to incorporate this in our total comparison. Just like we compared which categories are present in each super-category, we will also compare which super-categories are present in the receipt. More specifically, on the one hand both receipts have both super-categories Non-Food as well as Non-Perishable as at least one category in each of these super-categories was bought. On the other hand, only receipt 2 has the super-category Perishable. As such from this point of view the receipts are somewhat similar. We call this fourth comparison the *inter-super-category* comparison, since it makes a comparison between multiple super-categories.

When determining the difference between the two receipts we want to account for all of the comparisons: Non-Food comparison, Non-Perishable comparison, Perishable comparison and inter-super-category comparison. It may however be that some of these comparisons are more important than others. For example; we stress more on the Non-Food comparison and inter-super-category comparison than the other two. These difference have a variety of reasons. They may result from the objective we have, be based on domain expert knowledge, or be decided from the structure of the data.

Finding out the *exact* importance of each comparison is a crucial yet non-trivial task. The search for the importance of each comparison can however be made significantly faster if we at least know the *order* of importance, i.e. if we for example know that the comparison made on the super-category Non-Food is the more important than the inter-super-category comparison, which in turn might be more important than the super-category comparison of Non-Perishable. For this ordering of we rely on the knowledge provided by domain experts.

3 Formalization

In this Section we formalize the distance metric we described in section 2. We first do so by formalizing the concepts. We then formalize the intra- and inter-super-category, and conclude with the complete distance metric. We refer back to the example in each of these four parts. We conclude this Section with a proof that the defined distance between two receipts is indeed a distance metric. A more elaborate example of the computations done in this Section can be found in Section 6.1.

3.1 Definitions

Let $\mathcal{C} = C_1 \cup C_2 \cup \ldots \cup C_h$ be a set of *categories*. Each C_i is a *super-category* containing categories. We further have that $C_i \cap C_j = \emptyset$ for $i \neq j$, such that each category is in exactly one super-category, and $C_i \neq \emptyset$ for all i, such that each super-category is non-empty. The set $\{C_1, C_2, \ldots C_h\}$ is called a *hierarchy*. Note that the hierarchy is a partition over $\mathcal{C}$.

Let $\mathcal{R} \subseteq \{0,1\}^{|\mathcal{C}|}$ be a set of receipts. A *receipt* is a description of the products bought in a *visit*. More specifically, for a receipt $\vec{r} \in \mathcal{R}$ we have that $\vec{r}_c = 1$ if a product of category c was bought, and $\vec{r}_i = 0$ otherwise. As a consequence we have that multiple visits may be represented by the same value of $\vec{r}$. Let $\mathcal{D}$ be a bag over $\mathcal{R}$. We have that a receipt $\vec{r}$ represents a total of $\mathcal{D}(\vec{r})$ visits.

In our example provided in Section 2 we have that $\mathcal{C}$ = {Baby Products, Detergent, Grains, Snacks, Spreads, Vegetables}, and that C_1 = Non-Food = {Baby Products, Detergent}, C_2 = Non-Perishable = {Grains, Snacks}, C_3 = Perishable = {Spreads, Vegetables}. Receipt 1 of Table 1 is represented as $\vec{a} = (1,1,1,0,0,0)$ and Receipt 2 of Table 2 as $\vec{b} = (1,0,0,1,1,0)$. Our limited example further has $\mathcal{R} = \{\vec{a}, \vec{b}\}$ and $\mathcal{D}(\vec{a}) = \mathcal{D}(\vec{b}) = 1$.

3.2 Intra-super-category distance

We want to compare receipts on each super-category. Let $\vec{a}, \vec{b} \in \mathcal{R}$. For $i = 1 \ldots h$ we define

$$\delta_{C_i}(\vec{a}, \vec{b}) = 1 - \frac{\sum_{c \in C_i} \vec{a}_c \cdot \vec{b}_c}{\sum_{c \in C_i} \vec{a}_c + \vec{b}_c - \vec{a}_c \cdot \vec{b}_c}. \tag{1}$$

Put differently, $\delta_{C_i}(\vec{a}, \vec{b})$ can be thought of as the Jaccard distance between the set of categories in C_i bought in receipts $\vec{a}$ and $\vec{b}$. Each of these distances is an *intra-super-category distance*, since we make a comparison within a super-category. Note that this distance is not properly defined defined if both $\vec{a}$ and $\vec{b}$ contain no categories for C_i (as the denominator will be 0). We will deal with this shortly.

In our example we have that $\delta_{C_1}(\vec{a}, \vec{b}) = \frac{1}{2}$ and $\delta_{C_2}(\vec{a}, \vec{b}) = \delta_{C_3}(\vec{a}, \vec{b}) = 1$.

3.3 Inter-super-category distance

Apart from the intra-super-category distance we also want make a comparison between different super-categories. Let $\vec{s} : \mathcal{R} \to \{0,1\}^h$ be a mapping that assigns a vector of length h to a receipt. The value $\vec{s}(\vec{r})_i$ indicates whether super-category i was bought for receipt $\vec{r}$. More specifically, we have that $\vec{s}(\vec{r})_i = 0$ if $\forall c \in C_i : \vec{r}_c = 0$, and 1 otherwise. We can now define:

$$\delta_s(\vec{a}, \vec{b}) = 1 - \frac{\sum_{i=1\ldots h} \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{a})_i}{\sum_{i=1\ldots h} \vec{s}(\vec{a})_i + \vec{s}(\vec{b})_i - \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{b})_i}. \tag{2}$$

Put differently, $\delta_s(\vec{a}, \vec{b})$ is the Jaccard distance between the sets of super-categories bought in receipts $\vec{a}$ and $\vec{b}$. This distance is alternatively referred to as the *inter-super-category distance* since we make a comparison between super-categories.

For our example we have that $\vec{s}(\vec{a}) = (1,1,0)$ and $\vec{s}(\vec{b}) = (1,1,1)$. This results in $\delta_s(\vec{a}, \vec{b}) = \frac{1}{3}$.

3.4 Distance metric

As stated in the example, we want to be able to give more importance to certain intra-super-category distances, and/or the inter-super-category distance. We do so by assigning weights to each of these distances. Let $\vec{w} = (\vec{w}_s, \vec{w}_1, \vec{w}_2, \ldots \vec{w}_h) \in \mathbb{R}_+ \times \mathbb{R}_0^h$. We define

$$\delta_{\vec{w}}(\vec{a}, \vec{b}) = \frac{\vec{w}_s \cdot \delta_s(\vec{a}, \vec{b}) + \sum_{i:\vec{s}(\vec{a})_i + \vec{s}(\vec{b})_i - \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{b})_i = 1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(\vec{a}, \vec{b})}{\vec{w}_s + \sum_{i:\vec{s}(\vec{a})_i + \vec{s}(\vec{b})_i - \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{b})_i = 1} \vec{w}_i}, \tag{3}$$

as the distance between receipts $\vec{a}$ and $\vec{b}$. The distance is a weighted average of the Jaccard distances $\delta_{\mathcal{C}_i}$ for $i = 1 \ldots h$ and δ_s. It is important to note that in the numerator, we sum over values i for which $\vec{s}(\vec{a})_i + \vec{s}(\vec{b})_i - \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{b})_i = 1$, since otherwise, $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ is not defined. In other words, we do not take the Jaccard distance of a super-category $\mathcal{C}_i$ in account if both receipts have no products from that super-category. We take the sum over the same values i in the denominator. We do not allow $\vec{w}_s = 0$ while we do allow $\vec{w}_i = 0$ for any $i = 1 \ldots h$. This is because for any $\vec{a}$ and $\vec{b}$ there is no guarantee that $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ is defined for all[1] i $= 1 \ldots h$. As such, if for some $\vec{a}$ and $\vec{b}$ we have that $\vec{w}_i = 0$ for all i for which $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ is defined, the denominator is still non-zero, and the distance $\delta_{\vec{w}}(\vec{a}, \vec{b})$ is just equal to $\delta_s(\vec{a}, \vec{b})$. Note that we could alternatively have disallowed $\vec{w}_i = 0$ for all i, but this solution allows certain super-categories to be ignored.

If we assign $\vec{w} = (1, 1, 1, 1)$ we get that

$$\delta_{\vec{w}}(\vec{a}, \vec{b}) = \frac{\frac{1}{3} + \frac{1}{2} + 1 + 1}{1 + 1 + 1 + 1} = \frac{17}{24}.$$

Our example has no super-categories for which both receipts are missing a super-category. We do have that receipt $\vec{a}$ does not have the super-category $\mathcal{C}_3$. Note that nonetheless $\delta_{\vec{w}}(\vec{a}, \vec{a}) = \delta_{\vec{w}}(\vec{b}, \vec{b}) = 0$.

4 Finding $\vec{w}$

To find $\vec{w}$ we use a clustering over $\mathcal{R}$ and internal evaluation metrics. Assuming we have a clustering, we want find the weights that minimize[2] the internal evaluation metric on the clustering.

Let $K = \{K_1 \cup K_2 \cup \ldots \cup K_n\}$ be a partition over $\mathcal{R}$. K is a clustering of the receipts, and as such represents the clustering over the visits. Using a clustering and a value of $\vec{w}$ we can define an internal evaluation metric $\Phi(\vec{w}, K)$ which describes the quality of clustering K with respect to weights $\vec{w}$, the lower the value of $\Phi(\vec{w}, K)$ the better. Let K be given, we want to find the $\vec{w}^*$ that minimizes $\Phi(\vec{w}, K)$.

There are several problems in minimizing $\Phi(\vec{w}, K)$. To discuss these, we give an example of $\Phi(\vec{w}, K)$, though the problems discussed generally hold for any internal evaluation metric. Let $X \subseteq \mathcal{R}$. For brevity of notation, define $\delta_{\vec{w}}(X)$ as the total distance between all visits that are represented in X:

$$\delta_{\vec{w}}(X) = \sum_{(\vec{a}, \vec{b}) \in X^2} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_{\vec{w}}(\vec{a}, \vec{b}). \tag{4}$$

We define $\Phi(\vec{w}, K)$ as:

$$\Phi(\vec{w}, K) = \log \sum_{i=1 \ldots n} \delta_{\vec{w}}(K_i) - \log \delta_{\vec{w}}(\mathcal{R}) \tag{5}$$

Computing $\vec{w}^*$ that minimizes $\Phi(\vec{w}^*, K)$ poses several problems:

1. Since $\delta_{\vec{w}}$ is non-linear in $\vec{w}$, so is $\delta_{\vec{w}}(X)$ and as such also $\Phi(\vec{w}, K)$.
2. Computing $\delta_{\vec{w}}(X)$ is quadratic in $|X|$ and linear in h. As such, evaluation of $\Phi(\vec{w}, K)$ for a given $\vec{w}$ is expensive.
3. We only have one-sided constraints (positive and non-negative) on individual values of $\vec{w}$, not between values.

[1] We do have the guarantee that it is defined for at least one i in $1 \ldots h$, since otherwise both receipts would be empty.

[2] Without loss of generality, we discuss minimization of the internal evaluation metric, the problem and solution is similar for maximization.

We are therefore dealing with a multi-dimensional non-linear minimization problem with linear constraints and an expensive cost function. Before considering solution strategies, we first discuss several optimizations that can in the very least reduce the effect of some of the problems.

5 Computational Optimizations

In this Section we propose several optimizations that are meant to reduce search time for $\vec{w}^*$ regardless of the chosen optimization technique or $\Phi(\vec{w}^*, K)$.

5.1 Limitations on the domain of $\vec{w}$

For any optimal value $\vec{w}^*$, any $a \cdot \vec{w}^*$ with $a \in \mathbb{R}_+$ is also an optimal value, since we take the weighted average. As such, we can limit our search space for the individual values of $\vec{w}$, as long as each of the domains of $\vec{w}_i$ is the same interval $[0, z]$ for some $z \in \mathbb{R}_+$, and $\vec{w}_s$ has domain $(0, z]$. We can further limit this search space by allowing only a finite domain for each weight; i.e. $\vec{w}_i \in 0...z$ and $\vec{w}_s \in 1...z$ for some $z \in \mathbb{N}_+$. This will lead to an approximation of $\vec{w}^*$, with a better approximation but higher computation time for higher values of z.

5.2 Relations between values of $\vec{w}$

We can make estimations about how each value in $\vec{w}$ relates to each other from domain knowledge. More specifically, we can base an ordering of $\vec{w}$ on whether some intra-super-category distances are considered more important than others, and also where in this ordering the inter-super-category distance is. This will add up to h additional constraints to our problem, and as such it will limit our search space. This will also imply that some $\vec{w}_i$ need to be non-negative too, if the respective intra-super-category distance is considered more important than the inter-super-category distance.

5.3 Computation of $\delta_{\vec{w}}(X)$

Most of the computational effort in Equation (4) comes from having to sum over all combinations of $\vec{a}$ and $\vec{b}$. In the optimization we propose in this Subsection, we precompute parts of this equation before optimizing $\vec{w}$. In order to do that we need to be able to isolate the parts that do not depend on $\vec{w}$. The idea is that we compute the sums of $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ over a subset of X^2, such that each of the summed $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ values is defined in that subset. A more elaborate example of the optimizations of this and the next section can be found in Section 6.2. Let $\vec{z}$ be a vector in $\{0, 1\}^h$ and let

$$X^2|_{\vec{z}} = \{(\vec{a}, \vec{b}) | \vec{a}, \vec{b} \in X \wedge \forall i = 1 \dots h : \vec{z}_i = \vec{s}(\vec{a})_i + \vec{s}(\vec{b})_i - \vec{s}(\vec{a})_i \cdot \vec{s}(\vec{b})_i\}. \tag{6}$$

Put differently, $X^2|_{\vec{z}}$ is the set of all pairs $\vec{a}, \vec{b}$ such that $\delta_{\mathcal{C}_i}(\vec{a}, \vec{b})$ is either defined for all pairs in $X^2|_{\vec{z}}$ ($\vec{z}_i = 1$) or for none of the pairs ($\vec{z}_i = 0$).

In our example we have that $X^2|_{(1,1,1)} = \{(\vec{a}, \vec{b}), (\vec{b}, \vec{a})\}$ and $X^2|_{\vec{z}} = \emptyset$ for all other $\vec{z}$.

We compute the sum of the distances over these pairs:

$$\sum_{(\vec{a},\vec{b}) \in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_{\vec{w}}(\vec{a}, \vec{b}).$$

Using Equation (3) and the definition of $\vec{z}$:

$$\delta_{\vec{w}}(\vec{a}, \vec{b}) = \frac{\vec{w}_s \cdot \delta_s(\vec{a}, \vec{b}) + \sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(\vec{a}, \vec{b})}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i}.$$

We substitute the latter in the former:

$$\sum_{(\vec{a},\vec{b}) \in X^2|\vec{z}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \frac{\vec{w}_s \cdot \delta_s(\vec{a}, \vec{b}) + \sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(\vec{a}, \vec{b})}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i}$$

Because the denominator of the third product term does not depend on $\vec{a}$ or $\vec{b}$, we can rewrite this as:

$$\frac{1}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i} \sum_{(\vec{a},\vec{b})\in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \left(\vec{w}_s \cdot \delta_s(\vec{a},\vec{b}) + \sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(\vec{a},\vec{b}) \right).$$

Since the weights do also no depend on $\vec{a}$ and $\vec{b}$ we have can rewrite this as:

$$\frac{1}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i} \left(\left[\vec{w}_s \cdot \sum_{(\vec{a},\vec{b})\in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_s(\vec{a},\vec{b}) \right] + \left[\sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \sum_{(\vec{a},\vec{b})\in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_{\mathcal{C}_i}(\vec{a},\vec{b}) \right] \right). \quad (7)$$

Two observations are important. The first is that the sums over $X^2|_{\vec{z}}$ contain the largest part of the computation. The second is that those sums do not depend on $\vec{w}$, they only depend on $\vec{z}$. As such we can precompute them before optimizing $\vec{w}$. To shorten notation, let

$$\delta_s(X^2|_{\vec{z}}) = \sum_{(\vec{a},\vec{b})\in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_s(\vec{a},\vec{b}) \quad (8)$$

$$\delta_{\mathcal{C}_i}(X^2|_{\vec{z}}) = \sum_{(\vec{a},\vec{b})\in X^2|_{\vec{z}}} \mathcal{D}(\vec{a}) \cdot \mathcal{D}(\vec{b}) \cdot \delta_{\mathcal{C}_i}(\vec{a},\vec{b}). \quad (9)$$

We can now rewrite (7) as:

$$\frac{1}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i} \left(\vec{w}_s \cdot \delta_s(X^2|_{\vec{z}}) + \sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(X^2|_{\vec{z}}) \right). \quad (10)$$

Next, note that $\bigcup_{\vec{z}\in\{0,1\}^h} X^2|_{\vec{z}} = X^2$ and $X^2|_{\vec{z}} \cap X^2|_{\vec{z}'} = \emptyset$ if $\vec{z} \neq \vec{z}'$. In other words, $X^2|_{\vec{z}}$ for $\vec{z} \in \{0,1\}^h$ forms a partition[3] over X^2. We can therefore rewrite Equation (4) as:

$$\delta_{\vec{w}}(X) = \sum_{\vec{z}\in\{0,1\}^h} \frac{1}{\vec{w}_s + \sum_{i:\vec{z}_i=1} \vec{w}_i} \left(\vec{w}_s \cdot \delta_s(X^2|_{\vec{z}}) + \sum_{i:\vec{z}_i=1} \vec{w}_i \cdot \delta_{\mathcal{C}_i}(X^2|_{\vec{z}}) \right). \quad (11)$$

Note that this creates a small memory overhead: we need to pre-compute on average $\frac{h}{2}+1$ values for each of the 2^h values of $\vec{z}$ for each of the $K+1$ values of X in order to compute Equation (5). The main advantage here is that only the precomputation scales (quadratic) with $|\mathcal{R}|$, but the number of values only depends on h. In other words; given $\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_h$, the value of $|\mathcal{R}|$ only influences the precomputation cost, but not the cost of the actual minimization of $\vec{w}$. As a result we gain a huge decrease in computational cost, with only a little added cost in terms of memory.

5.4 Bitwise computations

The equations discussed so far involve many vectors from a binary space. This allows a final optimization: using bitwise operations. We will first introduce some additional binary vectors and add some vector manipulations (Section 5.4.1), and then transform the problem to do bitwise computations instead (Section 5.4.2). This transformation introduces an additional optimization (Section 5.4.3).

5.4.1 Additional vectors and vector manipulation

Given a set S', we can define a binary vector representation for its subsets. More specifically, if S is some subset of a set $S' = \{S'_1, S'_2, \ldots S'_n\}$, then $\vec{S} \in \{0,1\}^n$ is the vector with $\vec{S}_i = 1$ if $S'_i \in S$ and 0 otherwise[4]. We can therefore represent each of the super-categories $\mathcal{C}_i$ as a vector $\vec{\mathcal{C}}_i \in \{0,1\}^n$, such that $\vec{\mathcal{C}}_{ic} = 1$ if $c \in \mathcal{C}_i$, and 0 otherwise. With this, we can rewrite Equation (1) as:

[3] Actually, there might be datasets $\mathcal{D}$ for which some of $X^2|_{\vec{z}}$ are empty (or all $\mathcal{D}$ for $\vec{z} = \vec{0}$), but these will result in zero values for their sums anyway.

[4] We implicitly already used this in our definition of receipts. This further assumes some order on the elements of a set. We omit the details of that in this document, but note that $\mathcal{C}$ is sorted alphabetically.

$$\delta_{\mathcal{C}_i}(\vec{a},\vec{b}) = 1 - \frac{\sum_{c\in\mathcal{C}}(\vec{\mathcal{C}_i}\odot\vec{a}\odot\vec{b})_c}{\sum_{c\in\mathcal{C}}(\vec{\mathcal{C}_i}\odot\vec{a}+\vec{\mathcal{C}_i}\odot\vec{b}-\vec{\mathcal{C}_i}\odot\vec{a}\odot\vec{b})_c},$$

where $\odot$ represents the element-wise multiplication. We can further abbreviate this as:

$$\delta_{\mathcal{C}_i}(\vec{a},\vec{b}) = 1 - \frac{||\vec{\mathcal{C}_i}\odot\vec{a}\odot\vec{b}||_1}{||\vec{\mathcal{C}_i}\odot\vec{a}+\vec{\mathcal{C}_i}\odot\vec{b}-\vec{\mathcal{C}_i}\odot\vec{a}\odot\vec{b}||_1}, \tag{12}$$

where $||\vec{x}||_1$ is the 1-norm of $\vec{x}$, or the sum over the elements of $\vec{x}$, which for binary vectors is known the *Hamming weight*.

We can apply the same rewriting to Equation (10) if we first define a vector $\vec{\delta_{\mathcal{C}}}(X^2|_{\vec{z}})$ where $\vec{\delta_{\mathcal{C}}}(X^2|_{\vec{z}})_i = \delta_{\mathcal{C}_i}(X^2|_{\vec{z}})$ if $z_i = 1$ and 0 otherwise. Using this, Equation (11) becomes:

$$\delta_{\vec{w}}(X) = \frac{1}{\vec{w}_s + ||\vec{z}\odot\vec{w}||_1}\left(\vec{w}_s\cdot\delta_s(X^2|_{\vec{z}}) + ||\vec{w}\odot\vec{\delta_{\mathcal{C}}}(X^2|_{\vec{z}})||_1\right). \tag{13}$$

In our example, we have that $\vec{\mathcal{C}_1} = (1,1,0,0,0,0), \vec{\mathcal{C}_2} = (0,0,1,1,0,0)$, and $\vec{\mathcal{C}_3} = (0,0,0,0,1,1)$.

5.4.2 Bitwise operations

The final improvement can be gained by performing bitwise operations on Equation 12. For this we redefine some of our notations. If $\vec{x}$ is some binary vector of length n (i.e. $\vec{x}\in\{0,1\}^n$, then $x\in\mathbb{N}$ is the integer such that its binary representation is the concatenation of $\vec{x}$, i.e. $\vec{x}_1\vec{x}_2\cdots\vec{x}_n$. Similarly, if S is a subset of S', and $\vec{S}$ is its binary vector representation (as discussed in Section 5.4.1), then $s\in\mathbb{N}$ is the integer representing the set S.

We next define several important operations. Let $\vec{a},\vec{b},\vec{c}\in\{0,1\}^n$, and let their integer representations be $a,b,c\in\mathbb{N}$, respectively. We have that the disjunction $a\wedge b$ is equal to c if $\forall_{i=1\ldots n}:\vec{a}_i\cdot\vec{b}_i = \vec{c}_i$. We further have that the conjunction[5] $a\vee b$ is equal to c if $\forall_{i=1\ldots n}:\vec{a}_i+\vec{b}_i-\vec{a}_i\cdot\vec{b}_i = \vec{c}_i$. Finally, we define the Hamming weight $H(a)$ as $||\vec{a}||_1$.

With the above definitions we can rewrite our distance computation. Let $c = c_1 + c_2 + ... + c_h = 2^{|\mathcal{C}|} - 1$ be the integer representing all categories, i.e. $H(c) = |\mathcal{C}|$, where $c_i\in\mathbb{N}_+$ is the integer representation of $\mathcal{C}_i$ for $i = 1\ldots h$. $\{c_1,c_2,\ldots,c_h\}$ is the hierarchy[6].

Let $\mathcal{R}\subseteq\{n\in\mathbb{N}_+|n<2^{|\mathcal{C}|}\}$ be a set of receipts. Let $\mathcal{D}$ be a bag over $\mathcal{R}$. We have that the integer $r\in\mathcal{R}$ represents $\mathcal{D}(r)$ visits.

Define $s:\mathcal{R}\to\{n\in\mathbb{N}_+|n<2^h\}$ as a mapping that assigns an integer i with $H(i)<h$ to a receipt. We have that $s(r) = i$ if $\vec{s}(\vec{r}) = \vec{i}$.

Let $a,b\in\mathcal{R}$. For $i = 1\ldots h$ we redefine

$$\delta_{\mathcal{C}_i}(a,b) = 1 - \frac{H(c_i\wedge a\wedge b)}{H(c_i\wedge(a\vee b))}, \tag{14}$$

$$\delta_s(a,b) = 1 - \frac{H(s(a)\wedge s(b))}{H(s(a)\vee s(b))}. \tag{15}$$

In the example provided in Section 2 we have that $c = 63$ and that $c_1 = 48, c_2 = 12$, and $c_3 = 3$. Receipt 1 of Table 1 is represented as $a = 56$ and Receipt 2 of Table 2 as $b = 38$. From this we can compute $s(a) = 6$ and $s(b) = 7$. We can then compute that $\delta_{\mathcal{C}_1}(a,b) = \frac{1}{2}, \delta_{\mathcal{C}_2}(a,b) = \delta_{\mathcal{C}_3}(a,b) = 1$, and $\delta_s(a,b) = \frac{1}{3}$.

We rewrite Equations (6), (8) and (9) as:

$$X^2|_z = \{(a,b)|a,b\in X\wedge z = s(a)\vee s(b)\}, \tag{16}$$

[5] A more common definition of the conjunction operator would be $\forall_{i=1\ldots n}:a\vee b = c\Leftrightarrow max(\vec{a}_i,\vec{b}_i) = \vec{c}_i$, but this computation is in line with the rest of this document.

[6] Note that the constraints defined on $\mathcal{C}_i$ in Section 3 that make the hierarchy a partition over $\mathcal{C}$ are captured by the requirements that c is the sum over c_i, and that none of the c_i may be 0.

$$\delta_s(X^2|_z) = \sum_{(a,b)\in X^2|_z} \mathcal{D}(a)\cdot\mathcal{D}(b)\cdot\delta_s(a,b), \tag{17}$$

$$\delta_{c_i}(X^2|_z) = \sum_{(a,b)\in X^2|_z} \mathcal{D}(a)\cdot\mathcal{D}(b)\cdot\delta_{\mathcal{C}_i}(a,b), \tag{18}$$

respectively. Let $\vec{\delta_c}(X^2|_z)$ be a vector such that $\vec{\delta_c}(X^2|_z)_i = \delta_{c_i}(X^2|_z)$ if $\vec{z}_i = 1$ and 0 otherwise. We can the rewrite Equation (11) as:

$$\delta_{\vec{w}}(X) = \sum_{z\in\{n\in\mathbb{N}_+|n<2^h\}} \frac{1}{\vec{w}_s + ||\vec{z}\odot\vec{w}||_1}\left(\vec{w}_s\cdot\delta_s(X^2|_z) + ||\vec{w}\odot\vec{\delta_c}(X^2|_z)||_1\right). \tag{19}$$

5.4.3 Using symmetry

Note that for all $a, b \in X$ we have that $(a,b)\in X^2|_z \Leftrightarrow (b,a)\in X^2|_z$ and $\delta_s(a,b) = \delta_s(b,a)$. We can therefore also only consider $\delta_s(a,b)$, skip $\delta_s(b,a)$ and double the value of $\delta_{c_i}(X^2|_z)$ and $\delta_s(X^2|_z)$. The integer representation of receipts also introduces[7] a way to assign an ordering to $\mathcal{R}$; i.e. we can say that receipt a is before receipt b if $a < b$. Using this and the fact that $\delta_{c_i} = \delta_s(a,a) = 0$, we can rewrite Equations (17) and (18) as:

$$\delta_{c_i}(X^2|_z) = 2\sum_{a\in X}\sum_{b\in\{r\in X|(a,r)\in X^2|_z\wedge r>a\}} \mathcal{D}(a)\cdot\mathcal{D}(b)\cdot\delta_{c_i}(a,b), \tag{20}$$

and

$$\delta_s(X^2|_z) = 2\sum_{a\in X}\sum_{b\in\{r\in X|(a,r)\in X^2|_z\wedge r>a\}} \mathcal{D}(a)\cdot\mathcal{D}(b)\cdot\delta_s(a,b). \tag{21}$$

This saves about half the time needed to compute $\delta_{c_i}(X^2|_z)$ and $\delta_s(X^2|_z)$.

5.5 Ease of implementation

This part is definitely for the hard-core only, and it is unlikely to be part of the publication. It serves no purpose but to explain how the inter-super-category distance can be interpreted as an intra-super-category distance using a special super-category that contains all other super-categories. It serves no improvement on memory or time complexity, but it makes the implementation much easier. Read this part only if you are either interested in the mathematical analogousness between the inter-super-category distance and the intra-super-category distances, if you enjoy the beauty of the logical computations behind it, or if you are trying to make sense of the implementation. A more elaborate example accompanying this Subsection can be found in Section 6.3.

From everything we have discussed so far, the arithmetic involved for the inter-super-category distance and weight is nearly identical to that of the intra-super-category distances and weights. In this Subsection we rewrite the work done so far, where we make use of this similarity. More specifically, with a few modifications, we can assign a designated super-category to take care of the inter-super-categories distances of the hierarchy, as if each elements of this designated super-category are the super-categories of the hierarchy. The idea is that we adapt $r\in\mathcal{R}$ such that its first h bits represent the value of $s(r)$, followed by the original bits in r.

Let $\ll$ and $\gg$ be the left- and right-shift bit operator respectively, and let $c_0 = (2^h - 1) \ll |\mathcal{C}|$. We adapt $r, \mathcal{R}$, and $\mathcal{D}$ to $\mathcal{R}' = \{0,1\}^h \times \mathcal{R}$, $r' = r + (s(r) \ll |\mathcal{C}|)$, and $\mathcal{D}'$ a mapping from $\{0,1\}\times\mathcal{R}$ to $\{0,1\}^h$, such that $\mathcal{D}'(r') = \mathcal{D}(r)$. We also define $s'(r') = s(r) = r' \gg |\mathcal{C}|$. Using this we can drop Equation (15) and make Equation (14) also hold for $i = 0$.

For the example from Section 2, we have that $c_0 = 448$, $a' = 56 + (6 \ll 6) = 56 + 384 = 440$, $b' = 38 + (7 \ll 6) = 38 + 448 = 486$. This checks out with $s'(a') = s(a)$ and $s'(b') = s(b)$, since $384 \gg 6 = 6$ and $486 \gg 6 = 7$. Entering these values in Equation (14) for $i = 0$ indeed gets $\delta_{c_0}(a', b') = \frac{1}{3}$, equivalent to the value of $\delta_s(a,b)$ found previously.

[7] We could have applied the exact same to $\vec{a}$ and $\vec{b}$, but the representation of integers allows for a clearer comparison and implementation.

Let again[8] $X \subseteq \mathcal{R}'$ and let z still be as defined previously[9]. As such Equation (16) becomes:

$$X^2|_z = \{(a', b') | a', b' \in X \wedge z = (s'(a') \vee s'(b'))\}. \tag{22}$$

For $i = 0 \dots h$ Equations (20) and (21) become:

$$\delta_{c_i}(X^2|_z) = \sum_{(a',b') \in X^2|_z} \mathcal{D}'(a') \cdot \mathcal{D}'(b') \cdot \delta_{c_i}(a', b'). \tag{23}$$

To shorten notation, let $\vec{\delta_c}(X^2|_z)$ be the vector of length $h+1$ containing the distances $\vec{\delta_c}(X^2|_z)_i = \delta_{c_i}(X^2|_z)$. Let $z' = z + 2^h$ (i.e. we add a 1 before the bit value of z), and let $\vec{z'}$ be its vector representation[10]. With that, Equation (19) becomes:

$$\delta_{\vec{w}}(X) = \sum_{z \in \{n \in \mathbb{N}_+ | n < 2^h\}} \frac{||\vec{\delta_c}(X^2|_z) \odot \vec{w}||_1}{||\vec{z'} \odot \vec{w}||_1}. \tag{24}$$

For computational purposes, let $\vec{\vec{\delta}}(X^2)$ be a matrix, such that $\vec{\vec{\delta}}(X^2)_{z,i} = \delta_{c_i}(X^2|_z)$. Let furthermore $\vec{\vec{Z'}}$ be a matrix such that $\vec{\vec{Z'}}_{z',i} = \vec{z'}_i$ (i.e. $\vec{\vec{Z'}}$ contains on its rows the bits of z'). Letting $\oslash$ be the element-wise division operator, representing $\vec{w}$ as a row-vector, and letting $\cdot$ be the dot product when operated on vectors or matrices, we can rewrite Equation (24) as:

$$\delta_{\vec{w}}(X) = ||(\vec{\vec{\delta}}(X^2) \cdot \vec{w}) \oslash (\vec{\vec{Z'}} \cdot \vec{w})||_1. \tag{25}$$

5.6 Some final implementation notes

We conclude the optimizations with some notes on how the implementation slightly differs from the logic discussed so far. This section is really only relevant if you want to dive into the actual implementation.
In Section 5.4.3 we discuss that we can save about half the computation time by considering the symmetry of the distance metric. In the text we enforce this by only computing $\delta_s(a, b)$ and $\delta_{c_i}(a, b)$ for $b > a$. In the implementation we treat X as an ordered list, and iterate a over the full list, and b over the elements *after* a.

In Section 5.5, Equation (25) we use a matrix $\vec{\vec{Z'}}$ for the denominator elements. We do not actually compute the matrix, but rather, we have a vector $\vec{Z'}$ of the z' values, and for the denominator result a vector $\vec{d}$ that is initiated as a 0 vector of the same length. We then loop over the $\vec{w}_i$ values, adding to $\vec{d}$: $\vec{w}_i$ times an element-wise bitwise-and between 1 and (the element-wise bit-shift of $\vec{Z'}$ by $h - i$). In other words, we have that

$$\vec{d}_j = \sum_{i=0 \dots h} (1 \wedge (\vec{Z'}_j \gg (h - i)) \cdot \vec{w}_i.$$

This is logically the same, but does not require creating $\vec{\vec{Z'}}$.

Instead of computing Equation (14) only for the pairs (a, b) for which it can be computed, we define the value of Equation (14) as 0 if the the equation could not be computed (i.e. if the denominator is 0). The result of this is in line with the vector notation used in Equations (19) and (24).

In Equation (25), the length of $\vec{\vec{\delta}}(X^2)$ is actually 2^h, we use values $z = 0 \dots 2^h - 1$ despite the fact that $X^2|_0 = \emptyset$. This is easier with indexing. Similarly, the denominator is computed for $\vec{z'}_j \in 2^h \dots 2^{h+1} - 1$. As long as $\vec{w}_0$ is positive, this is no issue.

[8] We abstain from adding an apostrophe to X to avoid cluttering the notation; X is still the same subset of receipts for which we want to compute the total distance, except the receipts are now defined slightly different.

[9] We do not need an adaption z' at this point, as the members of $X^2|_z$ are simply the old receipts (a, b) transformed to (a', b'), i.e. interpreting the inter-super-category distance as a super-category does not change the receipt combinations in $X^2|_z$ as the distance of the super-category representing the inter-super-category distance is always defined.

[10] We now do need the adaption z', or more specifically its vector representation, since it allows us to easily include the weight w_s

6 An elaborate example

In this Section we consider an elaborate example. We will go through all of the steps needed to compute the total distance over the set of transactions. We do so three times; once in Section 6.1 by following the procedure of Section 3, once in Section 6.2 by following the procedure with all optimizations up to, but excluding, Section 5.5, and finally once in Section 6.3 to also include the latter. The purpose of this Section is to verify the logic, to act as a guiding example, and to help any future me understand what I did.

6.1 Regular computation

We keep the hierarchy shown in Section 2. As such we have that $\mathcal{C}$ = {Baby Products, Detergent, Grains, Snacks, Spreads, Vegetables}, and that $\mathcal{C}_1$ = Non-Food = {Baby Products, Detergent}, $\mathcal{C}_2$ = Non-Perishable = {Grains, Snacks}, $\mathcal{C}_3$ = Perishable = {Spreads, Vegetables}.

We suppose that we have the following receipts:

Super Cat.	Non-Food		Non-Perishable		Perishable	
Cat.	Baby Products	Detergent	Grains	Sweets	Spreads	Vegetables
a	X		X	X		X
b	X	X				
c	X	X		X	X	
d						X
e				X		

Table 3: The receipts used in the elaborate example. And X indicates a product was bought in that category.

We have that $\vec{a} = (1,0,1,1,0,1), \vec{b} = (1,1,0,0,0,0), \vec{c} = (1,1,0,1,1,0), \vec{d} = (0,0,0,0,0,1)$ and $\vec{e} = (0,0,0,1,0,0)$. We have that $\mathcal{R} = \{\vec{a}, \vec{b}, \vec{c}, \vec{d}, \vec{e}\}$. Furthermore $\mathcal{D}(\vec{a}) = \mathcal{D}(\vec{b}) = \mathcal{D}(\vec{c}) = \mathcal{D}(\vec{d}) = \mathcal{D}(\vec{e}) = 1$. We avoid the multiplicity of receipts, as this is the most trivial part of the computation. We have that $\vec{s}(\vec{a}) = (1,1,1), \vec{s}(\vec{b}) = (1,0,0), \vec{s}(\vec{c}) = (1,1,1), \vec{s}(\vec{d}) = (0,0,1), \vec{s}(\vec{e}) = (0,1,0)$. We can compute the intra- and inter-super-category distances using Equation (1) and Equation (2). The results are presented in Tables 4 to 7, where '$-$' denotes that the value is not valid.

$\delta_{\mathcal{C}_1}(\vec{r_1}, \vec{r_2})$	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{d}$	$\vec{e}$
$\vec{a}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	1
$\vec{b}$	$\frac{1}{2}$	0	0	1	1
$\vec{c}$	$\frac{1}{2}$	0	0	1	1
$\vec{d}$	1	1	1	$-$	$-$
$\vec{e}$	1	1	1	$-$	$-$

Table 4: Non-Food

$\delta_{\mathcal{C}_2}(\vec{r_1}, \vec{r_2})$	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{d}$	$\vec{e}$
$\vec{a}$	0	1	$\frac{1}{2}$	1	$\frac{1}{2}$
$\vec{b}$	1	$-$	1	$-$	1
$\vec{c}$	$\frac{1}{2}$	1	0	1	0
$\vec{d}$	1	$-$	1	$-$	1
$\vec{e}$	$\frac{1}{2}$	1	0	1	0

Table 5: Non-Perishable

$\delta_{\mathcal{C}_3}(\vec{r_1}, \vec{r_2})$	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{d}$	$\vec{e}$
$\vec{a}$	0	1	1	0	1
$\vec{b}$	1	$-$	1	1	$-$
$\vec{c}$	1	1	0	1	1
$\vec{d}$	0	1	1	0	1
$\vec{e}$	1	$-$	1	1	$-$

Table 6: Perishable

$\delta_s(\vec{r_1}, \vec{r_2})$	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{d}$	$\vec{e}$
$\vec{a}$	0	$\frac{2}{3}$	0	$\frac{2}{3}$	$\frac{2}{3}$
$\vec{b}$	$\frac{2}{3}$	0	$\frac{2}{3}$	1	1
$\vec{c}$	0	$\frac{2}{3}$	0	$\frac{2}{3}$	$\frac{2}{3}$
$\vec{d}$	$\frac{2}{3}$	1	$\frac{2}{3}$	0	1
$\vec{e}$	$\frac{2}{3}$	1	$\frac{2}{3}$	1	0

Table 7: inter-super-category

We note that δ_s has no invalid values; as this is not possible. We further note that the elements of each diagonal are either 0 or $-$. We assign the weights $\vec{w} = (1,1,1,1)$ and use Equation (3) to compute Table 8.

$\delta_{\vec{w}}(\vec{r_1}, \vec{r_2})$	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{d}$	$\vec{e}$
$\vec{a}$	$\frac{0}{24}$	$\frac{19}{24}$	$\frac{12}{24}$	$\frac{16}{24}$	$\frac{19}{24}$
$\vec{b}$	$\frac{19}{24}$	$\frac{0}{12}$	$\frac{16}{24}$	$\frac{18}{18}$	$\frac{18}{18}$
$\vec{c}$	$\frac{12}{24}$	$\frac{16}{24}$	$\frac{0}{24}$	$\frac{22}{24}$	$\frac{16}{24}$
$\vec{d}$	$\frac{16}{24}$	$\frac{18}{18}$	$\frac{22}{24}$	$\frac{0}{12}$	$\frac{18}{18}$
$\vec{e}$	$\frac{19}{24}$	$\frac{18}{18}$	$\frac{16}{24}$	$\frac{18}{18}$	$\frac{0}{12}$

Table 8: The total distance

We deliberately did not simplify the fractions; as the denominators are actually an indication of the denominators that one would get in Equation (3). We can for example see that there was one distance value for $(\vec{d}, \vec{e})$ that could not be computed, this is the intra-super-category Non-Food, since $\vec{s}(\vec{d})_1 + \vec{s}(\vec{e})_1 - \vec{s}(\vec{d})_1 \cdot \vec{s}(\vec{e})_1 = 0 + 0 - 0 \cdot 0 = 0$. The total distance adds up to 16.

6.2 Fast and Bitwise computation

When interpreting vectors as integers we have that $c_1 = 110000_2 = 48, c_2 = 001100_2 = 12$ and $c_3 = 000011_2 = 3$. As a result we have that $c = 48 + 12 + 3 = 63 = 111111_2 = 2^{|\mathcal{C}|} - 1$. Next we have $a = 101101_2 = 45, b = 110000_2 = 48, c = 110110_2 = 54, d = 000001_2 = 1$ and $e = 000100_2 = 4$. We can also compute the s values : $s(a) = 111_2 = 7, s(b) = 100_2 = 4, s(c) = 111_2 = 7, s(d) = 001_2 = 1$ and $s(e) = 010_2 = 2$.

For $X^2|_z$ we have that:

$$
\begin{aligned}
X^2|_0 &= \emptyset \\
X^2|_1 &= \{(d,d)\} \\
X^2|_2 &= \{(e,e)\} \\
X^2|_3 &= \{(d,e),(e,d)\} \\
X^2|_4 &= \{(b,b)\} \\
X^2|_5 &= \{(b,d),(d,b)\} \\
X^2|_6 &= \{(b,e),(e,b)\} \\
X^2|_7 &= \{(a,a),(a,b),(a,c),(a,d),(a,e),(b,a),(b,c),(c,a), \\
&\quad (c,b),(c,c),(c,d),(c,e),(d,a),(d,c),(e,a),(e,c)\}
\end{aligned}
$$

As such, using Equations (17) and (18) we that for $z \in \{0,1,2,4\}$: $\delta_j(X^2|_z) = 0$ if $\vec{z}_i = 1$ or not defined if $\vec{z}_i = 0$ for all $j \in \{s, 48, 12, 3\}$ (these contain only distances between the same receipt integer or no distances at all). Table 9 shows the remaining values. Note that some of the elements are blank, for these values we have that $\vec{z}_i = 0$, and as such the distance in $\mathcal{C}_i$ is needed for the pairs in $X^2|_z$. Further note that we get the same results when using Equations (20) and (21).

$\delta_i(X^2\|z)$	$i = 48$	$i = 12$	$i = 3$	$i = s$
$z = 3$		2	2	2
$z = 5$	2		2	2
$z = 6$	2	2		2
$z = 7$	10	10	12	8

Table 9: Values for $\delta_i(X^2|_z)$

We now have that

$$\begin{aligned}
\vec{\delta_c}(X^2|_0) &= (0,0,0)\\
\vec{\delta_c}(X^2|_1) &= (0,0,0)\\
\vec{\delta_c}(X^2|_2) &= (0,0,0)\\
\vec{\delta_c}(X^2|_3) &= (0,2,2)\\
\vec{\delta_c}(X^2|_4) &= (0,0,0)\\
\vec{\delta_c}(X^2|_5) &= (2,0,2)\\
\vec{\delta_c}(X^2|_6) &= (2,2,0)\\
\vec{\delta_c}(X^2|_7) &= (10,10,12)
\end{aligned}$$

Still using $\vec{w} = (1,1,1,1)$, we use Equation (19) to compute that $\delta_{\vec{w}}(X)$ is $0+0+0+2+0+2+2+10 = 16$, as expected.

6.3 Implemented computation

In this Subsection we follow the improvement discussed in Section 5.5.
We now do not use the inter-super-category metric separately; we instead have $c_0 = (2^h - 1) \ll |C| = 7 \ll 6 = 111_2 \ll 6 = 111000000_2 = 448$, c_1 through c_3 remain unchanged. We compute that

$$\begin{array}{rclcrclcrclcrcl}
a' &=& a + (s(a) \ll |C|) &=& 45 + (111_2 \ll 6) &=& 45 + 111000000_2 &=& 45 + 448 &=& 493 &=& 110110111_2\\
b' &=& b + (s(b) \ll |C|) &=& 48 + (100_2 \ll 6) &=& 48 + 100000000_2 &=& 48 + 256 &=& 304 &=& 100110000_2\\
c' &=& c + (s(c) \ll |C|) &=& 54 + (111_2 \ll 6) &=& 54 + 111000000_2 &=& 54 + 448 &=& 502 &=& 111110110_2\\
d' &=& d + (s(d) \ll |C|) &=& 1 + (001_2 \ll 6) &=& 1 + 001000000_2 &=& 1 + 64 &=& 65 &=& 001000001_2\\
e' &=& e + (s(e) \ll |C|) &=& 4 + (010_2 \ll 6) &=& 4 + 010000000_2 &=& 4 + 128 &=& 132 &=& 010000100_2
\end{array}$$

The values for $s'(a')$ through $s'(e')$ are the same as $s(a)$ through $s(e)$ from the previous Subsection, just computed with a different formula. For $X^2|_z$ we have that:

$$\begin{aligned}
X^2|_0 &= \emptyset\\
X^2|_1 &= \{(d',d')\}\\
X^2|_2 &= \{(e',e')\}\\
X^2|_3 &= \{(d',e'),(e',d')\}\\
X^2|_4 &= \{(b',b')\}\\
X^2|_5 &= \{(b',d'),(d',b')\}\\
X^2|_6 &= \{(b',e'),(e',b')\}\\
X^2|_7 &= \{(a',a'),(a',b'),(a',c'),(a',d'),(a',e'),(b',a'),(b',c'),(c',a'),\\
&\quad (c',b'),(c',c'),(c',d'),(c',e'),(d',a'),(d',c'),(e',a'),(e',c')\}
\end{aligned}$$

Note that the sets from the previous Subsection are the same except for the apostrophes.

Using Equation (23) we that for $z \in \{0,1,2,4\}$: $\delta_j(X^2|_z) = 0$ if $\vec{z}_i = 1$ or not defined if $\vec{z}_i = 0$ for all $j \in \{448, 48, 12, 3\}$ (these contain only distances between the same receipt integer or no distances at all). Table 10 shows the remaining values. Note that some of the elements are blank, for these values we have that $\vec{z}_i = 0$, and as such the distance in C_i is not defined for the pairs in $X^2|z$. Further note that we get the same results when using Equations (20) and (21).

$\delta_i(X^2\|z)$	$i=48$	$i=12$	$i=3$	$i=448$
$z=3$		2	2	2
$z=5$	2		2	2
$z=6$	2	2		2
$z=7$	10	10	12	8

Table 10: Values for $\delta_i(X^2|_z)$

We now have that

$$\begin{aligned}
\vec{\delta_c}(X^2|_0) &= (0,0,0,0)\\
\vec{\delta_c}(X^2|_1) &= (0,0,0,0)\\
\vec{\delta_c}(X^2|_2) &= (0,0,0,0)\\
\vec{\delta_c}(X^2|_3) &= (2,0,2,2)\\
\vec{\delta_c}(X^2|_4) &= (0,0,0,0)\\
\vec{\delta_c}(X^2|_5) &= (2,2,0,2)\\
\vec{\delta_c}(X^2|_6) &= (2,2,2,0)\\
\vec{\delta_c}(X^2|_7) &= (8,10,10,12)
\end{aligned}$$

Now using Equation (25), we have that:

$$\vec{\vec{\delta}}(X) = \begin{pmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 2 & 0 & 2 & 2\\ 0 & 0 & 0 & 0\\ 2 & 2 & 0 & 2\\ 2 & 2 & 2 & 0\\ 8 & 10 & 10 & 12 \end{pmatrix}, \vec{w} = \begin{pmatrix} 1\\ 1\\ 1\\ 1 \end{pmatrix}, \vec{\vec{Z'}} = \begin{pmatrix} 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 0\\ 1 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\\ 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Such that:

$$\vec{\vec{\delta}}(X)\cdot\vec{w} = \begin{pmatrix} 0\\ 0\\ 0\\ 6\\ 0\\ 6\\ 6\\ 40 \end{pmatrix}, \vec{\vec{Z'}}\cdot\vec{w} = \begin{pmatrix} 1\\ 2\\ 2\\ 3\\ 2\\ 3\\ 3\\ 4 \end{pmatrix}, \delta_{\vec{w}}(X) = ||(\vec{\vec{\delta}}(X)\cdot\vec{w}) \oslash (\vec{\vec{Z'}}\cdot\vec{w})||_1 = \left|\left| \begin{pmatrix} 0\\ 0\\ 0\\ 2\\ 0\\ 2\\ 2\\ 10 \end{pmatrix} \right|\right|_1 = 16,$$

as expected.

Change Log

Version	Date	Changes
1	20-01-2020	Created document.
2	29-01-2020	Renamed categories to make them alphabetical.
3	03-02-2020	Renamed the super-categories too.
4	05-02-2020	Equation (24) is rewritten to match Equation (25). Renamed (super)categories and categories to be alphabetically and use (Non)-Perishable instead of (Non)-Fresh. Added nicer picture of a hierarchy. Added comment on indexing in Section 5.6.
5	07-04-2020	Anonymized document for PKDD2020 anonymous repository.