# Fusion Distance

The following content uses the notations of each paper.

# HQANN

## Original Definition in the Paper

Suppose $s_i$ is a hybrid datapoint consisting of feature vector $\mathbf{x}_i$ (embeddings) and attributes $\mathbf{v}_i$. The $k$-th attribute of $\mathbf{v}_i$ is $\mathbf{v}_i[k]$.

The definition of fusion distance $Dist(s_i, s_j)$ is

$$Dist(s_i, s_j) = w \cdot g(\mathbf{x}_i, \mathbf{x}_j) + f(\mathbf{v}_i, \mathbf{v}_j) \tag{1}$$

where $g(\mathbf{x}_i, \mathbf{x}_j)$ and $f(\mathbf{v}_i, \mathbf{v}_j)$ denote distance functions for feature vectors and attributes separately.

- $g$ is usual vector distance like L2 distance (Euclidean), cosine ($1 - \cos(\mathbf{x}, \mathbf{y})$), Inner Product ($1 - \mathbf{x} \cdot \mathbf{y}$) (all equivalent when normalized)
- $f$ is defined as follows, where $e(\mathbf{v}_i, \mathbf{v}_j) = \sum_{k=1}^{n} |\mathbf{v}_i[k] - \mathbf{v}_j[k]|$ ($n$ is the number of attributes) and $\lg$ is base $10$:

$$f(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} 0, & \mathbf{v}_i = \mathbf{v}_j \\ bias - \frac{1}{\lg(e(\mathbf{v}_i, \mathbf{v}_j) + 1)}, & \mathbf{v}_i \neq \mathbf{v}_j \end{cases} \tag{2}$$

- The paper states: All embeddings are normalized, so $\max_{i,j}\left(g(\mathbf{x}_i, \mathbf{x}_j)\right) \leq 1$.
- To make $bias - \frac{1}{\lg 2} \gg w \cdot \max_{i,j}\left(g(\mathbf{x}_i, \mathbf{x}_j)\right)$, in most cases, just set $w = 0.25,\ bias = \max_{i,j}\left(g(\mathbf{x}_i, \mathbf{x}_j)\right) + \frac{1}{\lg 2} = 1 + \frac{1}{\lg 2} \approx 4.32$.
- However, for normalized vectors, $\max_{i,j}\left(g(\mathbf{x}_i, \mathbf{x}_j)\right) \leq 2$, not $1$, so I think the bias should be modified as $5.32$.

## Substituted with recommended parameters and for single-label situation

Since with single label, $v_i$ is an integer. The distance is then:

$$Dist(s_i, s_j) = 0.25 \cdot g(\mathbf{x}_i, \mathbf{x}_j) + f(v_i, v_j) \tag{3}$$

$$f(v_i, v_j) = \begin{cases} 0, & v_i = v_j \\ 1, & v_i \neq v_j \end{cases} \tag{4}$$

# NHQ

## Original Definition in the Paper

Suppose $e_i$ represent a datapoint consisting of feature vector $\boldsymbol{\nu}(e_i)$ and attribute vector $\boldsymbol{\ell}(e_i)$.

The fusion distance function is

$\Gamma(e_i, e_j) = w_v \cdot \delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j)) + w_\ell \cdot \chi(\boldsymbol{\ell}(e_i), \boldsymbol{\ell}(e_j))$

where $\delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j))$ is the feature vector distance and $\chi(\boldsymbol{\ell}(e_i), \boldsymbol{\ell}(e_j))$ is the attribute vector distance.

- $\delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j))$ is usual vector distance.
- $\chi(\boldsymbol{\ell}(e_i), \boldsymbol{\ell}(e_j)) = \sum_{k=0}^{m-1} \phi(\boldsymbol{\ell}(e_i)^k, \boldsymbol{\ell}(e_j)^k)$ ($m$ is the number of attributes, $\boldsymbol{\ell}(e_i)^k$ means the $k$-th attribute of $e_i$)

  where

$$\phi(\boldsymbol{\ell}(e_i)^k, \boldsymbol{\ell}(e_j)^k) = \begin{cases} 0, & \boldsymbol{\ell}(e_i)^k = \boldsymbol{\ell}(e_j)^k \\ 1, & \boldsymbol{\ell}(e_i)^k \neq \boldsymbol{\ell}(e_j)^k \end{cases} \tag{5}$$

- The recommended parametes: $w_v = 1, w_l = \frac{\delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j))}{m}$, which makes $\delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j)) \leq \Gamma(e_i, e_j) \leq 2 \cdot \delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j))$
- In a word, use attribute mismatch count to dynamically **scale** the embedding distance.

### Substituted with recommended parameters and for single-label situation

In single-label situation, $m = 1$, $\ell(e_i)$ is an integer, so

$$\Gamma(e_i, e_j) = \begin{cases} \delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j)), & \ell(e_i) = \ell(e_j) \\ 2 \cdot \delta(\boldsymbol{\nu}(e_i), \boldsymbol{\nu}(e_j)) & \ell(e_i) \neq \ell(e_j) \end{cases} \tag{6}$$

# Other Fusion Distance to try

Overall, just modify the weight of different distances.

## 1. modify w in HQANN

For the HQANN fusion distance, adjust $w$ to control the penalty of label mismatch:

$$Dist(s_i, s_j) = w \cdot g(\mathbf{x}_i, \mathbf{x}_j) + f(v_i, v_j) \tag{7}$$

or

$$Dist(s_i, s_j) = g(\mathbf{x}_i, \mathbf{x}_j) + w \cdot f(v_i, v_j) \tag{8}$$

Same thing, both adjusting the weight of two distances

In the HQANN, the label penalty (attribute distance) is much larger than vector distance, maybe we can try to make vector distance larger and attribute distance smaller.

## 2. Make them comparable, view embedding as a smoother of label.

If $s_i, s_j$ both have labels,

$$Dist(s_i, s_j) = \lambda \cdot \frac{1}{2} g(\mathbf{x}_i, \mathbf{x}_j) + (1 - \lambda) f(v_i, v_j) \tag{9}$$

Then we have $\frac{1}{2} g(\mathbf{x}_i, \mathbf{x}_j) \leq 1, f(v_i, v_j) \leq 1$, and $Dist(s_i, s_j) \leq 1$.

If either of $s_i, s_j$ does not have labels,

$$Dist(s_i, s_j) = \frac{1}{2} g(\mathbf{x}_i, \mathbf{x}_j) \tag{10}$$

In this way, the distance of the two stages have the same scale, making neighbor list more comparable (but same scale does not mean same distribution, so this may not be good, just some thought).