center cen_y. Let classification function $f_y(x) = 1$ when $x \in B_{r_O}(cen_y)$, and $f_y(x) = 0$ otherwise. Also let h be the positive half space defined by a binary SVM decision hyperplane Ω obtained using positive and negative training examples, and let the size of ball B_{r_O} be bounded by Ω, $B_{r_O} \cap h = B_{r_O}$. We define open space as

$$O = S_o - B_{r_y}(cen_y)$$

where radius r_O needs to be determined from the training data for each known positive class.

This open space formulation greatly reduces the open space risk compared to traditional SVM and 1-vs-Set Machine in (Scheirer et al., 2013). For traditional SVM, whose classification function $f_y^{SVM}(x) = 1$ when $x \in h$, and positive open space being approximately $h - B_{r_y}(cen_y)$, which is only bounded by the SVM decision hyperplane Ω. For 1-vs-Set Machine in (Scheirer et al., 2013), whose classification function $f_y^{1-vs-set}(x) = 1$ when $x \in g$, where g is a slab area with thickness δ bounded by two parallel hyperplanes Ω and Ψ (Ψ ∥ Ω) in h. And its positive open space is approximately $g - B_{r_y}(cen_y)$. Given open space formulations of traditional SVM and 1-vs-Set Machine, we can see that both methods label an unlimited area as positively labeled space, while our formulation reduces it to a bounded spherical area.

Given the above open space definition, the question is how to estimate radius r_O for the positive class. We show that the center-based similarity space learning (CBS learning) recently proposed in (Fei and Liu, 2015) is suitable for the purpose. It was original proposed to deal with the negative covariate shift problem in binary text classification.

Below, we first introduce CBS learning and then discuss why it is suitable for our problem, as well as its underlying assumptions.

3.2 Center-Based Similarity Space Learning

We now discuss CBS learning for binary text classification. Let $D = \{(\mathbf{d}_1, y_1), (\mathbf{d}_2, y_2), \ldots, (\mathbf{d}_n, y_n)\}$ be the set of training examples, where $\mathbf{d}_i$ is the feature vector (e.g., with unigram features) representing a document d_i and $y_i \in \{1, -1\}$ is its class label. This feature vector is called a document space vector (*ds-vector*). Traditional classification directly uses D to build a binary classifier. CBS learning transforms each *ds*-vector $\mathbf{d}_i$ (no change to its class) to a center-based similarity space feature vector (CBS vector) **cbs-v**$_i$. Each feature in the CBS vector is a similarity between a center c_j of the positive class documents and d_i. CBS learning can use multiple document space representations or feature vectors (e.g., one based on unigrams and one based on bigrams) to represent each document, which results in multiple centers for the positive documents. There can also be multiple document similarity functions used to compute similarity values. The detailed learning technique is as follows.

For a document d_i in D, we have a set R_i of p *ds*-vectors $R_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_p^i\}$. Each *ds*-vector $\mathbf{x}_j^i$ denotes one document space representation of the document d_i, e.g., unigram representation or bigram representation. Then the center of positive training documents can be computed, which is represented as a set of p centroids $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p\}$, each of which corresponds to one document space representation in R_i. Rocchio method in information retrieval (Rocchio, 1971; Manning et al. 2008) is used to compute each center $\mathbf{c}_j$ (a vector), which uses the corresponding *ds*-vectors of all training positive and negative documents.

$$\mathbf{c}_j = \frac{\alpha}{|D_+|} \sum_{d_i \in D_+} \frac{\mathbf{x}_j^i}{\|\mathbf{x}_j^i\|} - \frac{\beta}{|D - D_+|} \sum_{d_i \in D - D_+} \frac{\mathbf{x}_j^i}{\|\mathbf{x}_j^i\|}$$

where D_+ is the set of documents in the positive class and $|.|$ is the size function. α and β are parameters, which are usually set empirically. It is reported that using *tf-idf* representation, $\alpha = 16$ and $\beta = 4$ usually work quite well (Buckley et al. 1994). The subtraction is used to reduce the influence of those terms that are not discriminative (i.e., terms appearing in both classes).

Based on R_i for any document d_i in both training and testing and the previously computed set C of centers using the training data, we can transform a document d_i from its document space representations R_i to one center-based similarity vector **cbs-v**$_i$ by applying a similarity function Sim on each element $\mathbf{x}_j^i$ of R_i and its corresponding center $\mathbf{c}_j$ in C.

$$\textbf{cbs-v}_i = Sim(R_i, C)$$

Sim has a set of similarity measures. Each measure m_j is applied to p document representations $\mathbf{x}_j^i$ in R_i and their corresponding centers $\mathbf{c}_j$ in C to generate p similarity features (*cbs*-features) in **cbs-v**$_i$.