Modeling Label Space Interactions in Multi-label Classification using Box Embeddings

Abstract

Multi-label classification is a challenging structured prediction task in which a set of output class labels are predicted for each input. Real-world datasets often have natural or latent taxonomic relationships between labels, making it desirable for models to employ label representations capable of capturing such taxonomies. Most existing multi-label classification methods do not do so, resulting in label predictions that are inconsistent with the taxonomic constraints, thus failing to accurately represent the fundamentals of problem setting. In this work we introduce the multi-label box model (MBM), a multi-label classification method that combines the encoding power of neural networks with the inductive bias and probabilistic semantics of box embeddings (Vilnis, et al 2018). Box embeddings can be understood as trainable Venn-diagrams based on hyper-rectangles. Representing labels by boxes rather than vectors, MBM is able to capture taxonomic relations among labels. Furthermore, since box embeddings allow these relations to be learned by stochastic gradient descent from data, and to be read as calibrated conditional probabilities, our model is endowed with a high degree of interpretability. This interpretability also facilitates the injection of partial information about label-label relationships into model training, to further improve its consistency. We provide theoretical grounding for our method and show experimentally the model's ability to learn the true latent taxonomic structure from data. Through extensive empirical evaluations on both small and large-scale multi-label classification datasets, we show that BBM can significantly improve taxonomic consistency while preserving or surpassing state-of-the-art predictive performance.

1 Introduction

Multi-label classification is a machine learning task in which an input is associated with multiple categories. Many real-world multi-label classification data sets in modalities such as text categorization [17], image classification [19, 16], entity typing[20, 22], functional genomics [1, 5], and so on, have a rich inter-dependent label structure that can be expressed using a taxonomy graph or a hierarchy. To be useful in practice, a model should produce predictions that are consistent with the label taxonomy. For example, if a book is classified as *drama*, it should also be classified as *fiction*, as shown by the label taxonomy of book genres in the left-hand side of Figure 1. More formally, given a label taxonomy in form of a directed acyclic graph $G = (\mathcal{L}, \mathcal{T})$, where the set of node $\mathcal{L}$ represents the labels and an edge $(a, b) \in \mathcal{T}$ implies that a is the parent of b in the taxonomy, if a model assigns scores s_a and s_b to these labels. Then, if $s_a \geq s_b$, these scores are deemed to be *consistent* with the taxonomy. In the case of book genre classification example, this implies that s_{fiction} must be greater than or equal to s_{drama}, regardless of the input.

The problem of producing consistent predictions for multi-label classification has garnered a lot of attention in the machine learning literature [30, 12, 20, 3]. Most methods that proposed to improve the consistency in predictions explicitly require complete label taxonomy either at inference time or both at training as well as inference time, making these models hard to scale to large label spaces

Preprint. Under review.

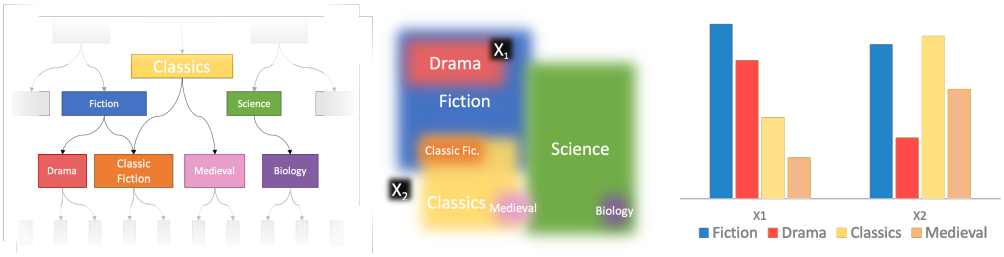

Figure 1: On the left we have an example of a label taxonomy that is represented as a DAG. The figure in the centre shows a possible layout of 2-dimensional box embeddings that capture this taxonomy accurately. The rightmost figure shows the scores assigned by the box embedding model to each label for two different inputs X_1 and X_2.

[12, 30]. This brings forth a question: Can we utilize representation learning to model the label-label relationships implicitly in the embedding space? Recent advances in non-Euclidean representation learning suggest optimism.

Vilnis et al. [28] show that probabilistic box embeddings, which represent concepts as high dimensional hyper-rectangles, can embed DAGs efficiently using the explicit information about the edges. Box embeddings represent edges by box-box containment as shown in Figure 1b. Representing the input and output labels in the same geometric space of boxes allows the multi-label taxonomy to be learned without an explicit taxonomic training signal. Moreover, there exists a large space of possible configurations that represent the same taxonomy, and if the label embeddings in the model gets close to any such configuration, then the model will always produce classifications that are consistent with the taxonomy, regardless of the input. We show through empirical evidence this is the case, and provide a formal proof for latter.

In this work, we propose the multi-label box model (MBM) that utilizes the geometry and probabilistic semantics of box embeddings to model label-label interactions in multi-label classification. MBM represents labels as boxes using free parameters and uses deep neural network to embed the input objects in the same space. We propose a new metric called consistency constrained mean average precision (CMAP) that combines mean average precision, which measures predictive performance, with consistency conditions. CMAP can be used to jointly measure the predictive and consistency performance of a model. Using this metric we show that MBM not only achieves state-of-the-art predictive performance but it also significantly improves the consistency of predicted scores w.r.t latent label taxonomy. Our analysis further shows that it is possible to retrieve the latent label-label relationships solely by analysing the learnt label representations inside the MBM, endowing the model with high degree of interpretability. Finally, we also present a way to utilize the interpretability of MBM to inject partial information about label-label relationships into the model thereby improving the consistency even further.

2 Related Work

Multi-label classification tasks that exhibit strong label space structure in the form of explicit label taxonomy are termed hierarchical multi-label classification (HMLC) in machine learning literature. Most approaches for such tasks make use of the complete hierarchy at the training time. These approaches can be categorized into two buckets [26]: (1) *Local approaches* that focus on local information for each label or clusters of labels in the hierarchy and classify them independently [3, 14], and (2) *Global approaches* that treat the problem as structured classification task and take global interactions into account [2]. In the most general setting, however, both local and global interactions between labels exist. The recent advances in deep learning [29] propose a specialized neural network architecture called Hierarchical Multi-Label Classification Network (HMCN-R and HMCN-F) that takes into account both local and global interactions by creating an ensemble of classifiers that can be trained using end-to-end gradient based training. However, HMCN does not try to enforce consistency strongly and focuses solely on predictive performance. In order to improve prediction consistency, recent works employ special loss functions on top of a neural network

classifier to enforce consistency w.r.t label taxonomy [20, 12]. While effective, these approaches still use the label taxonomy explicitly, making them difficult to scale to very large label spaces.

The recent advances in representation learning provide various methods to embed large graphs and taxonomies parsimoniously in non-euclidean spaces [21, 28, 27, 13]. The most prominent of these embedding methods include hyperbolic embeddings [21, 9, 10] and box embedding [28, 18, 6]. The use of non-euclidean embeddings for improving the consistency of multi-label classification has been limited to specific domains like text [4] or specific tasks like entity typing [22]. Moreover, while both hyperbolic and box embedding can model hierarchical relationships, it has been shown that the box embedding can also model more general graphs like DAGs much more efficiently than hyperbolic embeddings [24]. Hence, we propose a model that uses box embedding to capture general label-label relationships without the explicit use of label taxonomy to improve the consistency of model predictions.

3 Overview of Box Embeddings

Notations: In the problem of multi-label classification, we are given a set of labels $\mathcal{L}$ where $L = |\mathcal{L}|$, and an instance can be labeled with an element $s \in \{0,1\}^L$, where projection to the ith coordinate $\pi_i(s) = 1$ means that the ith label is true. We call the set of all such labelings S, and the associated probability space $(S, \mathcal{P}(S), P_S)$. We use $\mathbb{I}$ to denote the set of all finite closed intervals $[\mu^-, \mu^+]$ in $\Omega \subset \mathbb{R}$ plus the empty set, i.e. $\mathbb{I} := \{[\mu^-, \mu^+] \subset \Omega \,|\, \mu^+ \geq \mu^-\} \cup \emptyset$, We denote the smallest σ-algebra containing $\mathbb{I}$ as $\sigma(\mathbb{I})$ and, given a valid finite measure ν, we consider the measure space $(\Omega, \sigma(\mathbb{I}), \nu)$. As a high dimensional generalization, $\mathbb{I}^d$ will denote a d-dimensional Cartesian product of $\mathbb{I}$.

Definition 1 (Box Embedding [28])**.** Let $B : \mathbb{I}^d \to S$ be a measurable function such that $B^{-1} \circ \pi_i^{-1}(1) = \prod_i^d [\mu_i^-, \mu_i^+] \in \mathbb{I}^d$. A *box embedding* is defined as the function $\text{Box} : \mathcal{L} \to \mathbb{I}^d$ which maps a label $\ell \in \mathcal{L}$ to $B^{-1} \circ \pi_\ell^{-1}(\{1\}) \in \mathbb{I}^d$.

The definition of *box embeddings* induces a push-forward measure Q on S such that for any $R \subseteq S$, $Q(S) = \nu \circ B^{-1}(R)$. The complete joint probability distribution over the labels can be modeled using Q as defined above; however, computing $B^{-1}(R)$ requires the use of inclusion-exclusion principle and hence is intractable for a general R.

In order to avoid local identifiability issues in training, Dasgupta et al. [7] interpret μ_i^- (resp. μ_i^+) as the location parameters of random variables M_i^- (resp. M_i^+) that are distributed according to GumbelMax (resp. GumbelMin) distributions, leading to a meta-probabilistic generalization of box embedding which they call *Gumbel Box Process*. Since the GumbelMax (resp. GumbelMin) is max (resp. min) stable distribution, it enables the computation of the location parameters of the intersection box as given in the following definition.

Definition 2 (Intersection Box [7])**.** Let $A = \prod_{i=1}^d [a_i^-, a_i^+]$ and $B = \prod_{i=1}^d [b_i^-, b_i^+]$ be two gumbel boxes expressed using their location parameters, then the location parameters of the intersection of these two *gumbel boxes* are given as

$$A \tilde{\cap} B = \prod_{i=1}^{d} \left[\beta \,\text{lse}\left(\frac{a_i^-}{\beta}, \frac{b_i^-}{\beta}\right) \,,\quad -\beta \,\text{lse}\left(-\frac{a_i^+}{\beta}, -\frac{b_i^+}{\beta}\right) \right], \tag{1}$$

where $\text{lse}(x, y) = \log(\exp(x) + \exp(y))$.

The expected volume of Gumbel boxes involves the Bessel Function of the Second Kind, however, as shown in Dasgupta et al. [7], this integral can be reasonably approximated using softplus function leading to the following definition for *approximate bessel volume*.

Definition 3 (Approximate Bessel Volume [7])**.** For a gumbel box $B = \prod_{i=1}^d [b_i^-, b_i^+]$ we define the approximate Bessel volume $\lambda : \mathbb{I}^d \to \mathbb{R}_+$ as

$$\lambda(B) := \prod_{i=1}^{d} \log\left(1 + \exp\left(\frac{b_i^+ - b_i^-}{\beta} - 2\gamma\right)\right).$$

In the next section, we formally demonstrate the suitability of box embeddings for capturing taxonomic label relationships, and for that we first prove a couple of useful facts regarding the Gumbel intersection and Bessel approximate volume.

Proposition 1. Approximate bessel volume *is monotonic with respect to set containment. That is for two Gumbel boxes* A, B*,*

$$a_i^- \geq b_i^- \quad \textit{and} \quad a_i^+ \leq b_i^+, \quad \forall i \in \{1, \ldots, d\} \iff \lambda(A) \leq \lambda(B). \tag{2}$$

Proof. Follows from the monotonicity of $\log(1 + \exp(.))$. □

Proposition 2. *For any two Gumbel boxes* A, B*,* $\lambda(A \tilde{\cap} B) \leq \lambda(B)$.

Proof. The fact that $\max(x, y) \leq \text{lse}(x, y)$, and the statement of proposition 1 together imply the desired result. □

Since, λ is neither normalized nor additive, it cannot be used as a probability measure on $(\Omega, \sigma(\mathbb{I}^d))$. However, we can use proposition 1 and 2 to define a conditional probability model as follows.

Corollary 1. *For two gumbel boxes* A, B*, let* $P_{\text{Box}}(A \mid B) = \frac{\lambda(A \tilde{\cap} B)}{\lambda(B)}$*, then*

(i) For any two gumbel boxes A, B*, we have* $0 \leq P_{\text{Box}}(A \mid B) \leq 1$.

(ii) $P_{\text{Box}}(A \mid C) \leq P_{\text{Box}}(B \mid C)$ *for any three gumbel boxes* A, B, C*, with* $a_i^- \geq b_i^-, a_i^+ \leq b_i^+$.

4 Multi-label Box Model

In order to perform the task of multi-label classification we need to model the conditional probabilities $P(Y|X)$ where $Y \in S$ and X is the input. Using definition 1, we define label *box embeddings* $\text{Box}_\psi : \mathcal{L} \to \mathbb{I}^d$ as

$$\text{Box}_\psi(\ell_i) := \prod_{j=1}^{d} [\psi_{i,j}^-, \ \psi_{i,j}^- + \log(1 + \exp \psi_{i,j}^+)],$$

where $\psi^-, \psi^+ \in \mathbb{R}^{L \times d}$ are trainable parameters. The input instance X is encoded into an element of $\mathbb{I}^d$ using a parametric *instance box embedding* $\text{Box}_\theta = I^d \circ \mathcal{F}_\theta : \mathcal{X} \to \mathbb{I}^d$, where $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^d$ is a neural network with parameters θ and $I^d : \mathbb{R}^d \to \mathbb{I}^d$ defined as

$$I^d(x) := \prod_{i=1}^{d} [f_i(x), f_{d+i}(x)], \quad \text{with} \quad f_i(x) := \begin{cases} x_i - \delta, & \text{for } i \leq d \\ x_i + \delta, & \text{for } d < i \leq 2d \end{cases},$$

where $\delta = 10^{-5}$. The conditional probability for $Y \in \mathcal{S}$ given input X is computed using conditional probability under the Gumbel box model as

$$P_{\text{MBM}}(Y|X; \psi, \theta) = \prod_{i=1}^{L} P_{\text{MBM}}(Y_i|X, \psi, \theta) := \prod_{i=1}^{L} P_{\text{Box}}(B^{-1} \circ \pi_i^{-1}(\{Y\}) \mid \text{Box}_\theta(X))$$

Using the definition of P_{Box} as stated through corollary 1, we get the following expression for the conditional probability of Y_i under the model

$$P_{\text{MBM}}(Y_i = 1|X; \psi, \theta) = \frac{\lambda\left(\text{Box}_\psi(\ell_i) \ \tilde{\cap} \ \text{Box}_\theta(X)\right)}{\lambda(\text{Box}_\theta(X))},$$

where the intersection $\tilde{\cap}$ is the *Gumbel Intersection* and measure λ is *Approximate Bessel Volume*.

4.1 Modeling label-space interactions

In Section 1, we alluded to the fact that the inductive bias of MBM allows it to efficiently model partially specified first-order label interactions. Now we make this remark more concrete. If the partial specification of label interaction is defined using a taxonomy that can be represented as a directed acyclic graph (DAG), the following proposition shows that MBM has a strong inductive bias towards maintaining consistency in its scores.

Proposition 3. *Let $G = (\mathcal{L}, \mathcal{T})$ denote a DAG defined over the labels where $\mathcal{L}$ is the set of all labels and $\mathcal{T} = \{(\ell_i, \ell_j) \,|\, \ell_i, \ell_j \in \mathcal{L},\ P_D(y_i = 1 \,|\, y_j = 1) = 1\}$ is the set of edges. Then there exists some ψ such that $P_{(\psi,\theta)}(y_i = 1 \,|\, x) \geq P_{(\psi,\theta)}(y_j = 1 \,|\, x)$, for all x, θ.*

Proof. For all $(\ell_i, \ell_j) \in \mathcal{T}$, let ψ be such that $\text{Box}_\psi(\ell_j) \subseteq \text{Box}_\psi(\ell_i)$. Note that such ψ exists since for each $i \in \{1, \ldots, L\}$, $\text{Box}_\psi(\ell_i)$ is defined using only ψ_i. It follows from corollary 1 that $P_{\text{Box}}(\text{Box}_\psi(\ell_i) \,|\, \text{Box}_\theta(X)) \geq P_{\text{Box}}(\text{Box}_\psi(\ell_j) \,|\, \text{Box}_\theta(X))$ for any X, θ. □

4.1.1 Learning

The entire MBM is specified using parameters (ψ, θ) where $\psi \in \mathbb{R}^{2d \times L}$ are the label embedding parameters and θ are the parameters of the instance encoder neural network $\mathcal{F}_\theta$. Given data $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$, the model parameters are learnt by minimizing negative log-likelihood loss L_{nll} using ADAM optimizer [15].

$$L_{\text{nll}}(\psi, \theta; D) = -\sum_{i=1}^{D} \sum_{j=1}^{L} \log P(y_j^{(i)} \,|\, x^{(i)}; \psi, \theta). \tag{3}$$

In order to empirically verify the intuition behind proposition 3, we also propose the use of label interaction loss

$$L_{\text{G}}(\psi) = -\sum_{(\ell_i, \ell_j) \in \mathcal{T}} s(\ell_i, \ell_j) + \sum_{(\ell_i, \ell_j) \notin \mathcal{T}} s(\ell_i, \ell_j) \tag{4}$$

that utilizes the geometry of box embeddings to inject partial information about label interactions specified using a label taxonomy $G = (\mathcal{L}, \mathcal{T})$. For the Box model, label interaction score for a pair of labels is defined as

$$s_{\text{MBM}}(\ell_i, \ell_j) := \log P_{\text{Box}}(\text{Box}_\psi(\ell_i) \,|\, \text{Box}_\psi(\ell_j)). \tag{5}$$

5 Baselines

Our choice of baselines reflects the focus of this work, i.e., introducing prediction consistency using suitable representation spaces. To this end, our baselines consist of two models–one a high-performing neural network that only uses Euclidean vector representations, and other that uses hyperbolic representations. The base input encoder architecture $\mathcal{F}_\theta$ in both these models is same as the one used in MBM.

5.1 Multi-label Vector Model

An input encoder neural network $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^d$ is used to encode the inputs and a label embedding matrix ψ is used to represent the labels. The conditional probability of labels given the input is modeled as

$$P_{\text{MVM}}(y_l = 1 \,|\, x; \psi, \theta) = \sigma(\mathcal{F}_\theta(X) \cdot \psi_l),$$

where σ is the logistic sigmoid function. The parameters (θ, ψ) are learnt through negative log-likelihood loss (Eq. 3).

5.2 Multi-label Hyperbolic Model

This model uses the hyperbolic projection $\Pi : \mathbb{R}^d \to \mathcal{B}^d$ and distance $d : \mathcal{B}^d \times \mathcal{B}^d \to \mathbb{R}_+$ as defined in Nickel and Kiela [21] to project Euclidean vectors to d-dimensional Poincaré ball

$\mathcal{B}^d = \{x \in \mathbb{R}^d \mid \|x\| < 1\}$, and to compute distance between two points in $\mathcal{B}^d$.

$$\Pi(x) := \frac{x}{1 + \sqrt{1 + \|x\|_2^2}} \ , \qquad d(u, v) := \operatorname{arcosh}\left(1 + 2\frac{\|u - v\|}{(1 - \|u\|_2^2)(1 - \|v\|_2^2)}\right).$$

The input is first encoded using $\mathcal{F}_\theta$ and then projected into $\mathcal{B}^d$. The unnormalized score for each label is computed as the negative of the distance between the hyperbolic projections of encoded input and label representation. Since the hyperbolic distance function consists of arcosh, the negative distance is interpreted as log-probability score

$$\log P_{\text{MHM}}(y_l = 1 \mid x) = -d(\Pi(\mathcal{F}_\theta(x)), \Pi(\psi_l)),$$

which is used to learn the parameters (ψ, θ) using negative log likelihood loss (Eq. 3). Further the hyperbolic distance is used to define label interaction scores in the hyperbolic space as

$$s_{\text{MHM}}(l_i, l_j) := d(\Pi(\psi_{l_i}), \Pi(\psi_{l_j})).$$

6 Evaluation and Results

In this section we evaluate the performance of MBM using various real-world multi-label classification datasets.[1] The performance on multi-label classification is usually measured with Mean Average Precision (MAP), that is the mean of the average precision values across instances. MAP, however, does not take into account inconsistencies in predicted scores w.r.t label taxonomy. For instance, recalling the earlier example in Figure 1, a consistent model would always assign higher score to *fiction* when compared to *drama*, since a book classified as *drama* should also be classified as *fiction*. Since, MAP is incapable of capturing such consistency conditions, we introduce two new metrics, namely, Constraint Violation and Constrained Mean Average Precision.

Constrained Mean Average Precision (CMAP) computes MAP after accounting for the scores violating latent label taxonomy constraints. This is done by modifying the score for each label to be the minimum of the scores of its ancestor in the taxonomy (including itself) before computing the MAP. That is, given the taxonomy G, $\text{CMAP}(s) = \text{MAP}(\tilde{s})$. Here, the modified scores $\tilde{s}$ are computed as:

$$\tilde{s}_i^{(k)} = \min_{l_j \in \text{Anc}_G(l_i) \cup \{l_i\}} s_i^{(k)}, \tag{6}$$

where $\text{Anc}_G(l)$ is the set of ancestors of l in graph G.

Constraint violation measures the extent to which the label scores generated by the model violate the partial ordering of the latent label taxonomy regardless of true labels for the instances. Hence, lower value of CV implies higher taxonomic consistency in the predictions. CV is computed as

$$\text{CV} = \frac{1}{|D||\mathcal{T}|} \sum_{k=1}^{|D|} \sum_{(l_i, l_j) \in \mathcal{T}} \mathbb{1}\left(s_i^{(k)} - s_j^{(k)} < 0\right). \tag{7}$$

6.1 Task1: Feature based multi-label classification

In order to assess whether the neural network encoder can encode any kind of input well into box space, we use 7 small-scale multi-label classification datasets spanning across three domains: text, images [8], and functional genomics [5].[2] The characteristics of these datasets are summarized in the top section of table 1. These datasets are ideal test bed for our model as they explicitly provide label space taxonomy. Moreover, all the labels of all training and test instances respect the label taxonomy. The inputs for the datasets are either categorical features or continuous feature vectors. We convert the categorical features into one-hot feature vectors and standardize all continuous features.

The input encoder $\mathcal{F}_\theta$, common for MBM and the baselines, consists of a MLP with 3 layers. We perform grid search over activations, hidden dimensions, dropout, learning rate and use the best parameters for each model.

[1] Links to the sources for all 10 datasets are provided in the Appendix.

[2] These datasets do not require a licence and are available for public usage.

Table 1: Summary of the datasets used in experiments. The first section summarizes the feature based multi-label datasets spanning across 3 domains: functional genomics, image and text. The bottom section summarizes large scale datasets: Blurb Genre Collection(BGC), RCV1 for multi-label text classification and TypeNet for entity typing.

Dataset	Domain	Input/Feature Type	Label Taxonomy	#Labels	#Instances		
					Train	Val	Test
Expr	Genomics	Continuous	Forest	500	1636	849	1288
Cellcycle	Genomics	Continuous	Forest	500	1628	848	1281
Derisi	Genomics	Continuous	Forest	500	1608	842	1275
Spo	Genomics	Continuous	Forest	500	1600	837	1266
Diatoms	Image	Continuous	Tree	399	1400	665	1054
Imclef07a	Image	Continuous	Tree	97	7000	3000	1006
Enron	Text	Binary	Tree	57	650	338	600
BGC	Text	Raw Text	Forest	142	58715	14785	18394
RCV1	Text	Raw Text	DAG	104	13890	9260	781265
TypeNet	Text	Raw Text	Forest	997	295068	16392	16393

Table 2 reports the test set performance of the MBM model along with the baselines. The metric values reported are averaged across 5 runs with different random seeds, and are accompanied by standard error interval. We observe that the predictive performance of MBM measured using MAP is comparable to or better than that of MVM. When consistency constraints are considered along with the predictive performance (CMAP), MBM consistently outperforms MVM. While, on one extreme where we have MVM that exhibits good predictive performance but fails to maintain consistency w.r.t the taxonomy (resulting in higher CV), on the other extreme we have MHM that exhibits lowest constraint violations but gives inadequate predictive performance. MBM, however, demonstrates good characteristics on both fronts–predictive performance as well consistency.

6.2 Task 2: Multi-label text classification and entity typing

We evaluate MBM on large-scale multi-label text classification and entity-typing datasets in addition to small feature-based datasets. All three datasets, TypeNet[20], BGC and RCV1 [17][3] have a rich input space consisting of raw text and have explicit label taxonomies (c.f. Table 1).

The input text for text classification and entity typing datasets can be split into two parts: main text x_m and auxiliary text x_a. While the main text consists of sentences containing entity mentions, text describing the book and complete news articles; the auxiliary text contains tokens for surface mention, books' title and headline of the new articles for TypeNet, BGC and RCV1 respectively. For encoding $x = (x_a, x_m)$ we use CNN based encoder described in Murty et al. [20] that is general enough to encode the inputs for text classification as well as entity typing. The tokens of (x_a, x_m) are encoded using GloVe embeddings [25] to produce (g_a, g_m). A single-layered CNN with $\tanh$ non-linearity is applied to g_m, followed by a max-pooling layer to obtain a single vector representation e_m for the main input. To generate e_a, a mean-pooling layer is applied to g_a. The encoded vector representation of the raw text is obtained by concatenating the two representations (e_a, e_m) and passing it through 2-layer MLP. This encoder setup is common for MBM and the baselines and the best performing hyper-parameters (activation, dropout, hidden dimensions, and learning rate) for each model are tuned using grid search.

As shown in Table 2, we observe similar trend with raw text datasets as with the feature based datasets. While the MVM performs slightly better than MBM comparing MAP, the latter has significantly lower CV and better CMAP. Hence, it can be concluded that even for larger text datasets, the MBM model strikes a fine balance between predictive performance and consistency.

[3]The license for the dataset is acquired.

Table 2: Performance comparison of MBM models with the baselines for feature based datasets and large-scale multi-label classification and entity tying datasets. The left section compares the MVM, MHM and MBM models with the best performing model highlighted in each row. The right section shows the performance when we include the taxonomy information in training through L_G (MHM-T and MBM-T), where the highlighted cells indicated an improvement in performance w.r.t the respective non-T version of the model. All the metrics reported are averaged across five runs with different seeds and includes the standard error interval.

Dataset	Metric	MVM	MHM	MBM	MHM-T	MBM-T
Expr	MAP	49.11 ± 0.08	39.44 ± 0.02	48.35 ± 0.08	38.23 ± 0.21	47.88 ± 0.06
	CMAP	46.47 ± 0.20	37.54 ± 0.06	48.60 ± 0.08	38.47 ± 0.22	48.13 ± 0.06
	CV↓	2.75 ± 0.10	0.72 ± 0.09	2.21 ± 0.07	1.61 ± 0.17	1.52 ± 0.04
Cellcycle	MAP	42.80 ± 0.24	38.68 ± 0.05	44.77 ± 0.24	38.54 ± 0.05	44.91 ± 0.09
	CMAP	42.55 ± 0.21	36.94 ± 0.05	44.94 ± 0.23	36.78 ± 0.07	45.07 ± 0.01
	CV↓	1.73 ± 0.08	0.79 ± 0.05	1.57 ± 0.04	0.96 ± 0.05	0.4 ± 0.01
Derisi	MAP	40.09 ± 0.26	37.96 ± 0.18	40.81 ± 0.03	37.44 ± 0.05	40.67 ± 0.07
	CMAP	39.50 ± 0.43	37.25 ± 0.13	40.93 ± 0.02	37.43 ± 0.03	40.72 ± 0.06
	CV↓	2.01 ± 0.15	0.77 ± 0.05	1.34 ± 0.01	1.23 ± 0.17	0.05 ± 0.01
Spo	MAP	40.12 ± 0.52	38.21 ± 0.15	41.35 ± 0.05	37.24 ± 0.05	41.37 ± 0.08
	CMAP	40.07 ± 0.46	37.23 ± 0.18	41.51 ± 0.04	37.31 ± 0.04	41.53 ± 0.08
	CV↓	1.42 ± 0.06	0.86 ± 0.05	1.53 ± 0.07	2.68 ± 0.51	1.21 ± 0.02
Enron	MAP	83.76 ± 0.16	79.61 ± 0.02	82.08 ± 0.11	79.60 ± 0.01	81.32 ± 0.15
	CMAP	75.65 ± 0.57	79.63 ± 0.02	82.12 ± 0.09	79.63 ± 0.04	81.34 ± 0.11
	CV↓	0.79 ± 0.14	0.35 ± 0.01	0.12 ± 0.04	0.48 ± 0.01	0.04 ± 0.02
Diatoms	MAP	78.40 ± 0.35	47.63 ± 0.03	83.47 ± 0.15	47.71 ± 0.03	82.75 ± 0.24
	CMAP	71.21 ± 0.54	47.65 ± 0.03	83.55 ± 0.15	47.73 ± 0.03	82.66 ± 0.27
	CV↓	7.23 ± 0.21	2.03 ± 0.06	3.48 ± 0.10	1.93 ± 0.17	0.19 ± 0.02
Imclef07a	MAP	73.97 ± 0.48	65.16 ± 0.16	74.3 ± 1.32	65.65 ± 0.20	74.94 ± 1.66
	CMAP	73.87 ± 0.38	64.78 ± 0.38	74.52 ± 1.3	64.56 ± 0.28	75.14 ± 1.63
	CV↓	2.98 ± 0.3	2.27 ± 0.17	2.92 ± 0.25	2.49 ± 0.22	2.83 ± 0.25
BGC	MAP	83.88 ± 0.09	72.44 ± 0.08	83.57 ± 0.16	72.61 ± 0.14	83.53 ± 0.18
	CMAP	78.86 ± 0.47	68.55 ± 0.07	83.70 ± 0.14	68.83 ± 0.08	83.54 ± 0.18
	CV↓	16.17 ± 0.45	0.99 ± 0.04	1.64 ± 0.06	1.2 ± 0.01	0.02 ± 0.0
RCV1	MAP	88.96 ± 0.03	76.19 ± 0.32	87.93 ± 0.08	79.40 ± 0.32	88.23 ± 0.3
	CMAP	86.81 ± 0.02	72.09 ± 0.34	87.81 ± 0.08	75.94 ± 0.31	88.11 ± 0.3
	CV↓	12.65 ± 0.41	2.26 ± 0.16	2.43 ± 0.15	1.19 ± 0.06	1.83 ± 0.11
TypeNet	MAP	88.48 ± 0.03	72.97 ± 0.13	87.84 ± 0.08	73.09 ± 0.07	87.58 ± 0.32
	CMAP	81.70 ± 0.93	72.43 ± 0.12	87.93 ± 0.08	72.49 ± 0.06	87.62 ± 0.32
	CV↓	6.31 ± 0.35	1.46 ± 0.01	0.41 ± 0.06	1.29 ± 0.08	0.01 ± 0.0

7 Analysis of Learned Label Embeddings

In this section we analyse the geometry of the learned label embeddings, finding that the simple geometry of box embeddings endows the MBM model with high degree of interpretability. In order to verify that label box embeddings are producing consistent scores by using inclusion in the box space, we inject into the model, the taxonomy information through the additional loss term (Eq. 4). As seen from the two right most columns in Table 2, injecting explicit taxonomic information into the label embeddings (MBM-T) further reduces the extent of constrain violation in the base MBM. Thus validating our intuition about the arrangement of label embeddings boxes.

To determine the extent to which the label embeddings capture the latent label taxonomy without it being explicitly provided, we perform ancestor-descendant classification solely using the learnt label

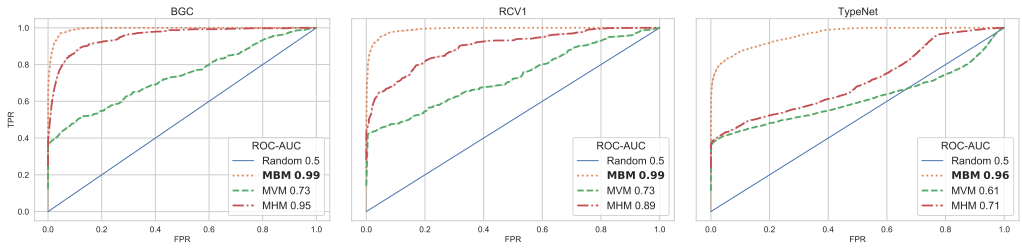

Figure 2: Above figure shows the ROC curves for the ancestor-descendant relationship classification in the label space for the models MBM, MVM and MHM across the large scale datasets: TypeNet, BGC and RCV1. The plot shows that the MBM outperforms MVM and MHM in capturing the label space taxonomy.

embeddings of the MBM model. Each pair of labels (l_i, l_j) get a score $\beta(i, j)$ that is determined using their corresponding label embeddings ψ_i, ψ_j. Since all three models have different geometrical interpretations, we use different scoring for each.[4] Specifically, $\beta_{\text{MVM}}(i, j) = \frac{\psi_i \cdot \psi_j}{\|\psi_j\|}$, $\beta_{\text{MBM}}(i, j) = s_{\text{MBM}}(l_i, l_j)$ and $\beta_{\text{MHM}}(i, j) = -(1+\alpha(\|\psi_i\|-\|\psi_j\|)s_{\text{MHM}}(l_i, l_j)$. These scores are then compared to true ancestor-descendant relations in the taxonomy to obtain respective ROC curves as shown in Figure 2. As seen, MBM captures the true label taxonomy the best ($AUC \geq 0.96$) for all datasets.

Table 3: Spearman rank correlation between the number of descendants in the true label taxonomy compared with each of the following: embedding magnitude for MVM, negative embedding magnitude for MHM and box embedding volume for MBM.

Dataset	**MVM**	**MHM**	**MBM**
BGC, RCV1, TypeNet	0.008, 0.12, -0.13	0.58, 0.49, 0.50	0.54, 0.46, 0.49

It is known that for the hyperberbolic space, the magnitude of embeddings relate to the level of generality in taxonomy[21]. We show that the same observation holds for box embeddings, with the vector embedding magnitude replaced by box embedding volume. To see this, we compute the spearman rank correlation between the number of descendants of a node in the true taxonomy and the embedding magnitude, negative embedding magnitude and embedding volume for MVM, MHM and MBM, respectively. The correlation values reported in Table 3 confirm our intuition regarding box embeddings stated above.

8 Conclusion

In this work, we demonstrate that box embeddings can more effectively capture taxonomic relations present between labels in the multi-label classification setting. This is true both intrinsically, captured via containment relationships between the box embeddings, and with respect to their labeling performance, as observed via improved MAP and CMAP metrics. Furthermore, our experiments validate that graph relationships between labels can be effectively injected via supervision during training, resulting in a consistent reduction in constraint violations (CV) on every dataset we evaluated on. Our model is thus an effective choice for multi-label classification both in settings with and without known taxonomic relations on the labels.

The problem of multi-label classification is very general and is of great practical applicability. Due to the promising performance of the proposed approach we believe that this work might have a broad impact. Hence, it is important to consider ethical issue pertaining to this work. Due to the high interpretability of our model, the most important concern would be regarding fairness, wherein biased data or taxonomic information can percolate the same into the model predictions. In order to avoid this, the data must be thoroughly inspected before the application of our method.

[4]The definition for β_{MHM} is taken from Nickel and Kiela [21], with $\alpha = 10^{-3}$.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2021/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to autmoatically hide this section in the anonymized submission.

References

[1] Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 01 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btk048. URL `https://doi.org/10.1093/bioinformatics/btk048`.

[2] David Belanger and Andrew McCallum. Structured prediction energy networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/belanger16.html`.

[3] Ricardo Cerri, Rodrigo C. Barros, and AndrÃ© C.P.L.F. de Carvalho. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80 (1):39 – 56, 2014. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2013.03.007. URL `http://www.sciencedirect.com/science/article/pii/S0022000013000718`.

[4] Soumya Chatterjee, Ayush Maheshwari, Ganesh Ramakrishnan, and Saketha Nath Jagaralpudi. Joint learning of hyperbolic label embeddings for hierarchical multi-label classification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2829–2841, Online, April 2021. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2021.eacl-main.247`.

[5] Amanda Clare. *Machine learning and data mining for yeast functional genomics*. PhD thesis, Citeseer, 2003.

[6] Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving Local Identifiability in Probabilistic Box Embeddings. 2020. ISSN 23318422. URL `https://www.iesl.cs.umass.edu/http://arxiv.org/abs/2010.04831`.

[7] Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems*, 2020. URL `https://proceedings.neurips.cc/paper/2020/file/01c9d2c5b3ff5cbba349ec39a570b5e3-Paper.pdf`.

[8] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10-11):2436–2449, 2011.

[9] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. Technical report, 2018. URL `https://en.wikipedia.org/wiki/`.

[10] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Neural Networks. 2018. doi: arXiv:1805.09112v2. URL `http://arxiv.org/abs/1805.09112`.

[11] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

[12] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent Hierarchical Multi-Label Classification Networks. 2020. URL http://arxiv.org/abs/2010.10151.

[13] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJxeWnCcF7.

[14] Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *International Conference on Information and Knowledge Management, Proceedings*, pages 1051–1060. ACM, 2019. ISBN 9781450369763. doi: 10.1145/3357384.3357885. URL https://doi.org/10.1145/3357384.3357885.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

[17] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[18] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. Smoothing the geometry of probabilistic box embeddings. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xSNiRcF7.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[20] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. Hierarchical losses and new resources for fine-grained entity typing and linking. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 97–109, 2018. ISBN 9781948087322. doi: 10.18653/v1/p18-1010. URL https://www.aclweb.org/anthology/P18-1010.

[21] Maximilian Nickel and Douwe Kiela. Poincar\'e Embeddings for Learning Hierarchical Representations. 2017. ISSN 19457871. doi: 10.1109/ICME.2013.6607554. URL http://arxiv.org/abs/1705.08039.

[22] Yasumasa Onoe, Michael Boratko, and Greg Durrett. Modeling fine-grained entity types with box embeddings, 2021.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[24] Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Li, Luke Vilnis, and Andrew McCallum. Representing joint hierarchies with box embeddings. In *Automated Knowledge Base Construction*, 2020. URL https://openreview.net/forum?id=J246NSqR_l.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

[26] C. N. Silla and A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 2010.

[27] Luke Vilnis and Andrew McCallum. Word representations via Gaussian embedding. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[28] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. 2018. URL `https://people.cs.umass.edu/~luke/box-lattices.pdfhttp://arxiv.org/abs/1805.06627`.

[29] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/wehrmann18a.html`.

[30] Jônatas Wehrmann, Ricardo Cerri, and Rodrigo C Barros. Hierarchical Multi-Label Classification Networks. Technical report, 2018.

A Implementation Details

In this section, we describe the implementation details of the input encoder for the text based datasets (BGC, RCV1 and Typenet), and the details of the training procedure and hyper-parameter search.

A.1 Input encoder

As discussed in section 6.2, the input text for text classification and entity typing can be split into two parts: main text x_m and auxiliary text x_a. For the raw text based datasets, we use the input encode from Murty et al. [20]. The encoder network $\mathcal{F}_\theta$ consists of GloVe embeddings $M \in \mathbb{R}^{|V| \times h}$ [25], a single layer CNN with $\tanh$ activations $\mathcal{W}$ and a 2-layer MLP P. Both the main and auxiliary inputs are passed through the embedding layer to get $(g_m, g_a) = (M(x_m), M(x_a))$. The the embeddings of the main input are then passed through the CNN and max-pooled to obtain e_m as:

$$e_m = \max_{0 \le i \le n_m - w + 1} \tanh \left(b + \sum_{j=0}^{w} W[j] g_m[i - \lfloor w/2 \rfloor + j] \right),$$

where n_m is the number of tokens in the main text, $W \in \mathbb{R}^{h,h}$ is the weight of the CNN filter, $b \in \mathbb{R}^d$ is the bias and the width w is set to 5. The embedding of the auxiliary input are simply mean pooled to get e_a, that is,

$$e_a = \frac{1}{n_a} \sum_{i=1}^{n_a} g_a.$$

The final vector representation for the input is obtained by concatenating e_a and e_m and passing them through a 2-layer MLP.

$$\mathcal{F}_\theta(x) = P(\text{cat}(e_a, e_m)).$$

Table 4: Example input instances from RCV1, BGC and Typenet datasets.

Dataset	main input desc.	auxillary input desc.	main input example	auxillary input example
TypeNet	list of sentences	list of surface mentions (one for each sentence)	[Henry Ford was an Industrialist, Ford was known for is pacifism during WW1, ...]	[Henry Ford, Ford, ...]
BGC	description of the book	book's title	The classic science fiction novel that captures and expands on the vision of Stanley Kubrick's immortal film...	2001: A Space Odyssey
RCV1	text of news article	headline of the article	France's cabinet met for the first time since the winter recess on Monday, amid expectations it could approve a decree....	France's cabinet meets

BGC As seen in Table 4, the main input x_m for BGC is the paragraph containing the description of a book and the auxiliary input x_a is the book's title. Hence, the encoded vector representation for an input instance is $\mathcal{F}_\theta((x_m, x_a))$.

RCV1 In terms of input structure, RCV1 is identical to BGC with a news article instead of a book description as the main input, and the headline of the article instead of a book title as the auxiliary input. Hence, for RCV1, we use the exact same encoder structure as BGC.

Typenet As seen in Table 4, since one input instance for Typenet consists of a list of 10 sentences containing entity mentions $(x_{m,1}, x_{m,2}, \ldots, x_{m,10})$ and the corresponding 10 surface mention phrases $(x_{a,1}, x_{a,2}, \ldots, x_{a,10})$, the final vector for the input representation is the mean of these i.e. $1/10 \sum_{k=1}^{10} \mathcal{F}_\theta((x_{m,k}, x_{a,k}))$.

A.2 Training Details

Frameworks used: We implement all the models described in this work using PyTorch [23]. We also make use of NLP specific abstractions over PyTorch provided by AllenNLP [11].

Data pre-processing: The datasets were pre-processed to remove noisy characters, fix encoding issues, tokenize text input using SpaCy tokenizer and map the labels to binary one hot encoding label vector.

Training: The learning algorithm used for training is minibatch gradient descent with a fixed batch size of 64. The ADAM [15] optimizer was used during training, along with ReduceLROnPlateau learning rate scheduler with early stopping.[5] Since the naive implementation of the label interaction loss described in Eq. 4 is too expensive to compute at each mini-batch step, we approximate it by randomly sampling (without replacement), at each mini-batch step, a subset of edges $\tilde{\mathcal{T}} \sim \mathcal{T}$. The size of the sampled set is a hyper-parameter (last column in Table 5).

Hyper-parameter search: All the results are reported using the best hyper-parameters found using grid search. Table 5 summarizes the search ranges used. The following hyper-parameters were searched based on the models used: optimizer learning rate, each feed forward layer's hidden dimensions and activation functions, dropout probabilities, label space dimensions/box space dimensions(half of hidden dimensions), weight and sampling percentage for the labels for computing the label interaction loss(L_G) for the MHM-T and MBM-T models. All the best model configurations are included in the code folder.

Table 5: Summary of the hyper-parameter search ranges for each dataset and model. The best hyper-parameters for each model and dataset combination were picked using grid search using MAP on the validation set. Except for the L_G *weight* and *label sample percent*, which are only applicable to the MBM-T and MHM-T models, the rest of the parameters are present across all models.

Datasets	lr	hidden dimensions	linear layers	activation	dropout	L_G weight	label sample percent
Feature based datasets	1e-5, 1e-4, 1e-2	250, 500, 1000, 1750	3	sigmoid, relu, tanh	0.0, 0.3, 0.5, 0.7	1e-3, 1e-5, 1e-7	5,10,20
BGC & RCV1	1e-3, 1e-4, 1e-5	150, 300	2	sigmoid, relu, tanh	0, 0.3, 0.5	1e-3, 1e-5, 1e-7	5,10,20
TypeNet	1e-3, 1e-4, 1e-5	310, 620	2	sigmoid, relu, tanh	0, 0.3, 0.5	1e-3, 1e-5, 1e-7	5,10,20

Compute: All the models were trained using Titanx GPUs. It takes less than an hour to train any of the reported models on feature-based datasets. For training any of the reported models on RCV1, BGC, and TypeNet, it takes approximately 2 hours, 4 hours and 22 hours, respectively.

A.3 Code

Completely anonymized and executable code with detailed instructions is provided using Anonymous Github.[6] The instructions cover all the use cases, i.e., obtaining the pre-processed datasets, training a new model from scratch (MBM or any baseline), evaluating a pre-trained model on test set, directly downloading the pre-trained models for larger datasets, and reproducing the graphs reported in the analysis section.

B Datasets

Table 6: The table provides the links to download the data from original source.

Dataset(s)	Download Links
Imclef07a, Enron, Diatoms	`http://kt.ijs.si/DragiKocev/PhD/resources/doku.php?id=hmc_classification`
Expr, Spo, Derisi, Cellcycle (FUN)	`https://dtai.cs.kuleuven.be/clus/hmcdatasets/`
TypeNet	`https://github.com/iesl/TypeNet`
BGC	`https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html`
RCV1 (License required)	`http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm`

[5] `https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.ReduceLROnPlateau`

[6] The anonymized code is available at `https://anonymous.4open.science/r/modeling-label-space-interactions-in-multi-label-classification-using-box-embeddings/README.md`