MODELING LABEL SPACE INTERACTIONS IN MULTI-LABEL CLASSIFICATION USING BOX EMBEDDINGS

Anonymous authors
Paper under double-blind review

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 (Lewis et al., 2004), image classification (Lin et al., 2015; Krishna et al., 2016), entity typing(Murty et al., 2018; Onoe et al., 2021), functional genomics (Barutcuoglu et al., 2006; Clare, 2003), 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 (Wehrmann et al., 2018b; Giunchiglia & Lukasiewicz, 2020; Murty et al., 2018; Cerri et al., 2014). 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 (Giunchiglia & Lukasiewicz, 2020; Wehrmann et al., 2018b). 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.

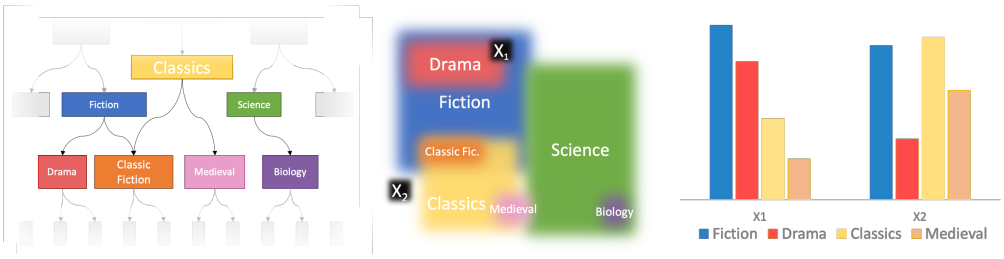


Figure 1: On the left we have an example of a label taxonomy that is represented as a DAG. The figure in the centre shows a possible layout of 2-dimensional box embeddings that capture this taxonomy accurately. The rightmost figure shows the scores assigned by the box embedding model to each label for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

Vilnis et al. (2018) 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 perform consistency analysis of the model using two measures and 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 (Silla & Freitas, 2010): (1) *Local approaches* that focus on local information for each label or clusters of labels in the hierarchy and classify them independently (Cerri et al., 2014; Huang et al., 2019), and (2) *Global approaches* that treat the problem as structured classification task and take global interactions into account (Belanger & McCallum, 2016). In the most general setting, however, both local and global interactions between labels exist. The recent advances in deep learning (Wehrmann et al., 2018a) 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 (Murty et al., 2018; Giunchiglia & Lukasiewicz, 2020). 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 (Nickel & Kiela, 2017; Vilnis et al., 2018; Vilnis & McCallum, 2015; Gu et al., 2019). The most prominent of these embedding methods include hyperbolic embeddings (Nickel & Kiela, 2017; Ganea et al., 2018a;b) and box embedding (Vilnis et al., 2018; Li et al., 2019; Dasgupta et al., 2020a). The use of non-euclidean embeddings for improving the consistency of multi-label classification has been limited to specific domains like text (Chatterjee et al., 2021) or specific tasks like entity typing (Onoe et al., 2021). 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 (Patel et al., 2020). 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 (Vilnis et al., 2018))**.** 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. (2020b) 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 (Dasgupta et al., 2020b))**.** 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

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

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

The expected volume of Gumbel boxes involves the Bessel Function of the Second Kind, however, as shown in Dasgupta et al. (2020b), this integral can be reasonably approximated using softplus function leading to the following definition for *approximate bessel volume*.

Definition 3 (Approximate Bessel Volume (Dasgupta et al., 2020b))**.** 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 the 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 $\mathrm{Box}_\psi(\ell_j) \subseteq \mathrm{Box}_\psi(\ell_i)$. Note that such ψ exists since for each $i \in \{1, \ldots, L\}$, $\mathrm{Box}_\psi(\ell_i)$ is defined using only ψ_i. It follows from corollary 1 that $P_{\mathrm{Box}}(\mathrm{Box}_\psi(\ell_i) \,|\, \mathrm{Box}_\theta(X)) \geq P_{\mathrm{Box}}(\mathrm{Box}_\psi(\ell_j) \,|\, \mathrm{Box}_\theta(X))$ for any X, θ. □

4.2 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

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

using the ADAM optimizer (Kingma & Ba, 2017). 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

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

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 another that uses hyperbolic representations. The base input encoder architecture $\mathcal{F}_\theta$ in both these models is same as the one used in MBM.

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).

Multi-label Hyperbolic Model As discussed in Chatterjee et al. (2021), the isometry between the Lorentz model and Poincare disk model for hyperbolic geometry can be used produce the retraction formula used to project d-dimensional euclidean vectors to d-dimensional Poincaré ball $\mathcal{B}^d = \left\{x \in \mathbb{R}^d \mid \|x\| < 1\right\}$. Here the projection operator $\Pi : \mathbb{R}^d \to \mathcal{B}^d$ and the distance $d : \mathcal{B}^d \times \mathcal{B}^d \to \mathbb{R}_+$ in the hyperbolic space are given as:

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

Specifically, 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

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

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

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

6 Evaluation and Results

In order to assess whether the neural network encoder can encode any kind of input well into box space, we evaluate the performance of MBM using 7 real-world multi-label classification datasets spanning across three domains: text, images (Dimitrovski et al., 2011), and functional genomics (Clare, 2003).[1] The characteristics of these datasets are summarized in 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.

Table 1: Summary of the datasets used in experiments. The feature based multi-label datasets span across 3 domains: functional genomics, image and text.

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

The Mean Average Precision (MAP), that is the mean of the average precision values across instances in the test set is used as the evaluation metric. 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 better than that of MVM and MHM on 5 out of the 7 datasets.

6.1 Consistency analysis

MAP is a good metric for predictive performance, however it does not take into account inconsistencies in the predicted scores w.r.t. the 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 perform further analysis to check the consistency of the predicted scores. This is done by considering two opposite perspectives towards consistency evaluation. The first one is a punitive approach, where we count the number of instances that are given inconsistent scores w.r.t the taxonomy. In the second one, a more constructive approach, we try to check the amenability of the model, by applying post-hoc corrections to the predicted scores to enforce consistency. The results of these two approaches, are presented by using Constraint Violation

[1]These datasets do not require a licence and are available for public usage, and links to the sources for all the datasets are provided in the Appendix.

Table 2: Performance comparison of MBM models with the baselines for the 7 feature based multi-label classification datasets. The left section compares the models that do not require explicit taxonomy, i.e., MVM, MHM and MBM. In the left section, the models with the best performance w.r.t. MAP are highlighted. The right section shows the performance when we include taxonomy information in training through L_G (MHM-T and MBM-T), where the highlighted cells indicate an improvement in performance (MAP and CV) w.r.t. the respective non-T model. All metrics reported are averaged across five runs with different seeds and include the standard error interval.

Dataset	Metric	MVM	MHM	MBM	MHM-T	MBM-T	CHMCNN
Expr	MAP	49.11 ± 0.08	39.44 ± 0.02	48.35 ± 0.08	38.23 ± 0.21	47.88 ± 0.06	46.29 ± 0.01
	CMAP	46.47 ± 0.20	37.54 ± 0.06	48.60 ± 0.08	38.47 ± 0.22	48.13 ± 0.06	46.29 ± 0.01
	CV↓	2.75 ± 0.10	0.72 ± 0.09	2.21 ± 0.07	1.61 ± 0.17	1.52 ± 0.04	0 0
Cellcycle	MAP	42.80 ± 0.24	38.68 ± 0.05	44.77 ± 0.24	38.54 ± 0.05	44.91 ± 0.09	42.8 0.04
	CMAP	42.55 ± 0.21	36.94 ± 0.05	44.94 ± 0.23	36.78 ± 0.07	45.07 ± 0.01	42.8 0.04
	CV↓	1.73 ± 0.08	0.79 ± 0.05	1.57 ± 0.04	0.96 ± 0.05	0.4 ± 0.01	0 0
Derisi	MAP	40.09 ± 0.26	37.96 ± 0.18	40.81 ± 0.03	37.44 ± 0.05	40.67 ± 0.07	40.02 0.04
	CMAP	39.50 ± 0.43	37.25 ± 0.13	40.93 ± 0.02	37.43 ± 0.03	40.72 ± 0.06	40.02 0.04
	CV↓	2.01 ± 0.15	0.77 ± 0.05	1.34 ± 0.01	1.23 ± 0.17	0.05 ± 0.01	0 0
Spo	MAP	40.12 ± 0.52	38.21 ± 0.15	41.35 ± 0.05	37.24 ± 0.05	41.37 ± 0.08	33.60 0.03
	CMAP	40.07 ± 0.46	37.23 ± 0.18	41.51 ± 0.04	37.31 ± 0.04	41.53 ± 0.08	33.60 0.03
	CV↓	1.42 ± 0.06	0.86 ± 0.05	1.53 ± 0.07	2.68 ± 0.51	1.21 ± 0.02	0 0
Enron	MAP	83.76 ± 0.16	79.61 ± 0.02	82.08 ± 0.11	79.60 ± 0.01	81.32 ± 0.15	82.29 0.02
	CMAP	75.65 ± 0.57	79.63 ± 0.02	82.12 ± 0.09	79.63 ± 0.04	81.34 ± 0.11	82.29 0.02
	CV↓	0.79 ± 0.14	0.35 ± 0.01	0.12 ± 0.04	0.48 ± 0.01	0.04 ± 0.02	0 0
Diatoms	MAP	78.40 ± 0.35	47.63 ± 0.03	83.47 ± 0.15	47.71 ± 0.03	82.75 ± 0.24	62.1 0.10
	CMAP	71.21 ± 0.54	47.65 ± 0.03	83.55 ± 0.15	47.73 ± 0.03	82.66 ± 0.27	62.1 0.10
	CV↓	7.23 ± 0.21	2.03 ± 0.06	3.48 ± 0.10	1.93 ± 0.17	0.19 ± 0.02	0 ± 0
Imclef07a	MAP	73.97 ± 0.48	65.16 ± 0.16	74.3 ± 1.32	65.65 ± 0.20	74.94 ± 1.66	74.64 ± 0.22
	CMAP	73.87 ± 0.38	64.78 ± 0.38	74.52 ± 1.3	64.56 ± 0.28	75.14 ± 1.63	74.64 ± 0.22
	CV↓	2.98 ± 0.3	2.27 ± 0.17	2.92 ± 0.25	2.49 ± 0.22	2.83 ± 0.25	0 ± 0

(CV) and Mean Average Precision post Consistency correction (CMAP), which are described below. Note that we do not recommend the use of either CV or CMAP as the sole metrics to evaluate the model, we present these as measures to analyse the consistency of the model predictions.

Constraint violation is a punitive measure that quantifies 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

$$\text{CV} = \frac{1}{|D||\mathcal{T}|} \sum_{k=1}^{|D|} \sum_{(l_i,l_j)\in\mathcal{T}} \mathbb{1}\left(s_i^{(k)} - s_j^{(k)} < 0\right). \tag{6}$$

Mean Average Precision post Consistency correction (CMAP): Given a complete or partial taxonomy G for the labels, consistency can be imposed post-hoc by applying a modification function $\delta : \mathbb{R}^L \to \mathbb{R}^L$ to the label-scores produced by the model. One such approach is presented by Giunchiglia & Lukasiewicz (2020), where the score for each label is modified to be the maximum (δ_G^M) of the scores of any of its descendants in the taxonomy G. However, an equally valid choice would be to modify the score to be the min of the descendants (δ_G^m). Concretely, given the scores $s \in \mathbb{R}^L$ produced by a model, for an input, the two modification functions are given as:

$$\delta_G^m(s)_i = \min_{l_j \in \text{Anc}_G(l_i)\cup\{l_i\}} s_j, \qquad \delta_G^M(s)_i = \max_{l_j \in \text{Des}_G(l_i)\cup\{l_i\}} s_j, \tag{7}$$

where $\text{Anc}_G(l)$ and $\text{Des}_G(l)$ is the set of ancestors and descendants, respectively, of l in the graph G. In practice, one would select whichever post-hoc modification performed better, and we refer to the maximum MAP obtained after applying these modification functions δ_G^m, δ_G^M to the scores as CMAP.[2] Intuitively, a CMAP value close to MAP value implies that the model is more perceptive of the latent hierarchy in the label space. Note that, while Giunchiglia & Lukasiewicz (2020) also modify the their learning objective to work well with the post-hoc correction, but here we use the post-hoc correction as a way to analyse the consistency of the predicted scores. In particular, the learning algorithm for MBM, MVM and MHM does not change, and although we include the CHMCNN model described in Giunchiglia & Lukasiewicz (2020) for comparison, it has to be noted that is not directly comparable with MVM, MHM and MBM as it requires the complete taxonomy during training as well as prediction time, without which it reduces to the MVM.

When consistency of the predicted scores is considered along with the predictive performance, MBM consistently outperforms MVM: it has significantly lower CV for 5 out of the 7 datasets and the difference $(\text{MAP} - \text{CMAP})$ for MBM is smaller for all 7 datasets indicating that MBM is much more perceptive of the latent taxonomy compared to MVM. To conclude, while on one extreme there is MVM that exhibits good predictive performance but fails to maintain consistency w.r.t the taxonomy, 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. When compared to CHMCNN which uses explicit taxonomy at training as well as prediction time, we see that MBM, that does not use the taxonomy explicitly, has significantly better MAP on 5 out of the 7 datasets, suggesting again that MBM provides a good trade-off between consistency and predictive performance.

6.2 Computational complexity

The computational complexity of the MBM can be divided into two parts–the computation of the instance box $\text{Box}_\theta(X)$, and the computation of probability score $P(Y \mid X)$ given the encoding of the instance and label box parameters. Computing $\text{Box}_\theta(X)$ amounts to splitting the output of a feed-forward network $\phi(x)$ into min and max parameters $\phi(x)^-, \phi(x)^+$, and thus has the same complexity as that of the MVM. Assuming that the operations $\sum, \prod, \log$ and $\exp$ have unit complexity, computing $P(Y \mid X)$ also has equivalent complexity. The computation of conditional probability for a single label y_k using the MVM model involves computing $\sigma(\phi(x) \cdot y_k)$, an $\mathcal{O}(dL)$ operation. Now, with same assumptions for the MBM, computing $\lambda(\text{Box}_\theta(x) \cap \text{Box}_\psi(y_k))$ involves d invocations of $\log(\exp(\cdot) + \exp(\cdot))$ followed by d subtractions and d invocations of $\log(1 + \exp(.))$, resulting in $\mathcal{O}(d)$ operations. The computation of $\lambda(\text{Box}_\theta(x))$ omits the intersection calculation, but otherwise is the same, and thus is also $\mathcal{O}(d)$. Hence, calculating $\frac{\lambda(\text{Box}(x) \cap \text{Box}(y_k))}{\lambda(\text{Box}(x))}$ is $\mathcal{O}(d)$, resulting in overall complexity of $\mathcal{O}(dL)$, which is equal to the MVM. We note that the complexity of our model is not dependent on the depth of the hierarchy, as in CHMCNN, and thus can reasonably scale to arbitrarily deep hierarchies as might be present in extreme multilabel classification. In addition to equivalent theoretical complexity, we also provide average runtimes presented in Appendix B.

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

[2] In our analysis, we find that as a pure post-hoc correction, δ_G^M produces lesser change in the scores predicted by the model, and hence is a better choice of this analysis (see Appendix F for further discussion on this)

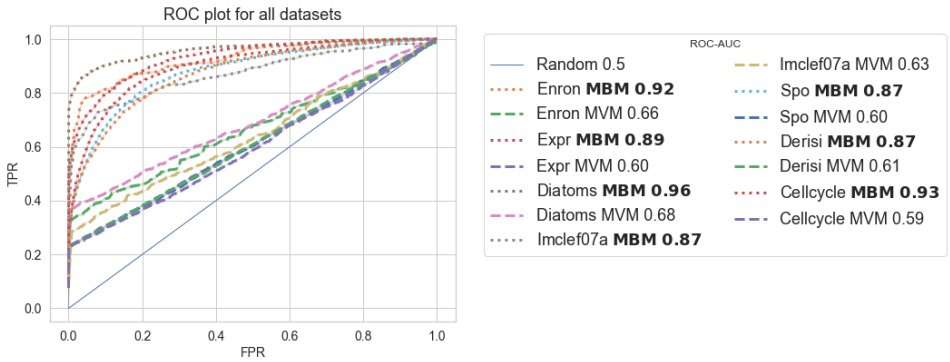


Figure 2: Above figure shows the ROC curves for the ancestor-descendant relationship classification in the label space for the models MBM and MVM across the datasets. The plot shows that the MBM outperforms MVM in capturing the label space taxonomy.

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.

Model	Expr	Cellcycle	Derisi	Spo	Enron	Diatoms	Imclef07a
MHM	-0.37	0.38	0.40	0.38	0.19	0.31	0.32
MVM	-0.06	-0.11	-0.01	0.06	-0.11	0.04	-0.02
MBM	0.47	0.49	0.50	0.48	0.47	0.23	0.43

label 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 MVM and MBM have different geometrical interpretations, we use different scoring for each. Specifically, $\beta_{\text{MVM}}(i, j) = \frac{\psi_i \cdot \psi_j}{\|\psi_j\|}$, $\beta_{\text{MBM}}(i, j) = s_{\text{MBM}}(l_i, l_j)$. 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 ($AUC \geq 0.87$) for all datasets.

For hyperberbolic space, it is suggested that the magnitude of embeddings relate to the level of generality in taxonomy (Nickel & Kiela, 2017). We show that the same observation holds for box embeddings, with the vector embedding magnitude replaced by box embedding volume.[3] 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 effectively capture taxonomic relations present between labels in the multi-label classification setting without the explicit access to the taxonomy. The multi-label box model achieves a fine balance between predictive performance and consistency. Furthermore, we find that the graph relationships between labels can be effectively injected via extra supervision during training, resulting in an increase in the consistency of the model predictions. Since the model has the same computation complexity as a simple vector neural network model (MVM), and unlike CHMCNN it does not require the label taxonomy, we believe that the proposed model can be scaled to work with extremely large label spaces. Moreover, due to the flexibility of using any neural network encoder this model can be easily extended to different input modalities like raw text, images, etc. We wish to pursue these two directions in our future work.

[3]Refer to Appendix I for scatter plots of box embedding volume vs. the number of children in the label taxonomy.

REFERENCES

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.

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

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.

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*, pp. 2829–2841, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.247.

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

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

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*, 2020b. URL https://proceedings.neurips.cc/paper/2020/file/01c9d2c5b3ff5cbba349ec39a570b5e3-Paper.pdf.

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

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

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

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.

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

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.

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*, pp. 1051–1060. ACM, 2019. ISBN 9781450369763. doi: 10.1145/3357384.3357885. URL https://doi.org/10.1145/3357384.3357885.

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

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.

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

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.

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.

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, pp. 97–109, 2018. ISBN 9781948087322. doi: 10.18653/v1/p18-1010. URL https://www.aclweb.org/anthology/P18-1010.

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.

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

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 (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

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.

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

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

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.

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

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

A IMPLEMENTATION DETAILS

In this section, we describe the implementation details: training procedure and hyper-parameter search.

Frameworks used: We implement all the models described in this work using PyTorch (Paszke et al., 2019). We also make use of NLP specific abstractions over PyTorch provided by AllenNLP (Gardner et al., 2017).

Data pre-processing: The datasets were pre-processed to remove noisy characters, fix encoding issues, standardize the continuous variables 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 (Kingma & Ba, 2017) optimizer was used during training, along with ReduceLROnPlateau learning rate scheduler with early stopping. 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 4).

Hyper-parameter search: All the results are reported using the best hyper-parameters found using grid search. Table 4 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 4: 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

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.

B COMPUTATIONAL COMPLEXITY

As discussed in Section 6.2, the computational complexity of MVM $\approx$ MBM $<$ MBM-T and Table 5 presents average epoch duration in seconds for MVM, MBM, MBM-T, and CHMCNN with the same hidden size.

Table 5: Average epoch duration in seconds for MVM, MBM, MBM-T and CHMCNN with the same hidden size

Dataset		Expr	Cellcycle	Spo	Derisi
Avg. epoch duration(s)	MVM	17.2	15.4	15.8	15.5
	MBM	18.4	16.9	16.3	15.2
	MBM-T	20.1	26.7	24.1	27.1
	CHMCNN	17.8	17.0	15.9	15.4

C CODE

Completely anonymized and executable code with detailed instructions is provided using Anonymous Github at `anonymous.4open.science/r/modeling-label-space-`

`interactions-in-multi-label-classification-using-box-embeddings`. 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 datasets, and reproducing the graphs reported in the analysis section.

D 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/`

E COMPARISON TO CHMCNN AND HMCN

In our experiments with the official code of CHMCNN(Giunchiglia & Lukasiewicz, 2020) as well as our re-implementation of the same, we found that CHMCNN consistently gets lower MAP compared to a well tuned MVM (note that the base MVM results are not reported in the CHMCNN or HMCN paper). In Giunchiglia & Lukasiewicz (2020), CHMCNN was shown to outperform HMCN(Wehrmann et al., 2018b). Based on these observations, we considered a well-tuned MVM to be the stronger baseline and only included the results of CHMCNN (our implementation) in table 2. Table 7 compares our models to the CHMCNN and HMCN models. Here, the columns with * are taken from their respective papers and these models use the union of train and validation set to train their final model. However, in all of our experiments, we only train on the train set to obtain the final test performance. As seen from the table, except for Imclef07a, MBM performs better than HMCN and CHMCNN (even with lesser training data).

Table 7: The table presents the results comparing MVM and MBM to CHMCNN(Giunchiglia et. al.) and HMCN-F (Wehrmann et. al.). Here, the columns with * are taken from their respective papers.

Dataset	**Metric**	**CHMCNN* (Giunchiglia et. al.)**	**CHMCNN (Giunchiglia et. al.)**	**HMCN-F* (Wehrmann et. al.)**	**CHMCNN (our implementation)**	**MVM**	**MBM**
Expr	AU(PRC)	30.2	27.66	30.1	37.58	41.08	38.88
	MAP	-	-	-	46.29	49.11	48.35
Cellcycle	AU(PRC)	25.5	24.02	25.2	35.06	35.2	36
	MAP	-	-	-	42.8	42.8	44.77
Derisi	AU(PRC)	19.5	18.91	19.3	30.48	31.95	32.6
	MAP	-	-	-	40.02	40.09	40.81
Spo	AU(PRC)	21.5	20.25	21.1	25.85	32.53	33.45
	MAP	-	-	-	33.6	40.12	41.35
Enron	AU(PRC)	75.6	75.29	72.4	79.2	80.5	78.38
	MAP	-	-	-	82.29	83.76	82.08
Diatoms	AU(PRC)	75.8	66.36	53.0	57.97	74.69	82.8
	MAP	-	-	-	62.1	78.4	83.47
Imclef07a	AU(PRC)	95.6	94.85	95.0	73.17	72.27	74.01
	MAP	-	-	-	74.64	73.97	74.3

F CMAP AND MAX OF DESCENDANTS FOR MVM

In MCL problems, where predictions consistent with a taxonomy are necessary, the model designer might be ready to sacrifice some absolute predictive performance (MAP) for increased consistency (CV). The motivation behind introducing CMAP was to capture this trade-off using a single metric. CMAP aims to measure the MAP post score corrections. If a human were to make consistent score corrections based on the complete hierarchy, they would either pick the score of a label to be the minimum of its ancestors (like we do in CMAP) or maximum of its descendants. Once picked, the same approach has to be applied to all the labels. We argue that in terms of correlation with human judgment, both these methods are equivalent. We resorted to empirical performance of the post-hoc

correction algorithm, we observed that the "min of ancestors" approach produced higher MAP with both MVM and MBM models compared to the "max of descendants" correction, as can be seen in table 8. We believe that "min of ancestors" works better because it promotes sparsity in predictions, which is essential for the task of MLC.

Table 8: CMAP represents post-hoc correction of scores by taking the "minimum of ancestors", CMAP' represents post-hoc correction of scores by taking "maximum of descendants". Empirically, CMAP produces better results for both MBM and MVM compared to CMAP'.

Dataset	**CMAP**	**CMAP'**	**MAP**
Expr	46.47	26.53	49.11
Cellcycle	42.50	17.58	42.80
Derisi	39.50	17.23	40.09
Spo	40.07	16.75	40.12
Enron	75.65	66.07	83.76
Diatoms	71.21	67.81	78.40
Imclef07a	73.87	43.58	73.97

G PERFORMANCE ON LONG TAIL OF LABELS

Since MVM and MBM both are embedding based models, we expect the performance of the MBM model to wary with the label frequency in the same manner as MVM. This hypothesis is corroborated by the following table which shows the spearman rank correlation between the label frequency and the mean average precision (MAP) for that label. The high positive correlation for all the models suggests that the performance of both MVM and MBM often degrades as the frequency of the labels drop. However, as shown by the table 9, both the models perform reasonably well for extremely low frequency labels. Here, tail_MAP is the mean average precision for labels with frequency 0.001% or lower.

Table 9: Comparing the performance of long tails of labels for MBM, MBM-T and MVM. Here, correlation is the spearman rank correlation between label frequency and MAP of the label and tail_MAP is the MAP for labels with frequency 0.001% or lower.

Dataset	MBM		MBM-T		MVM	
	Correlation	tail_MAP	Correlation	tail_MAP	Correlation	tail_MAP
Expr	0.72	48.91	0.73	48.68	0.75	49.45
Cellcycle	0.74	46.11	0.75	46.17	0.82	44.02
Derisi	0.79	41.86	0.82	41.62	0.79	41.78
Spo	0.8	42.67	0.77	42.75	0.79	42.15
Enron	0.84	79.19	0.83	74.97	0.91	67.63
Diatoms	0.29	86.3	0.29	85.46	0.28	82.2
Imclef07a	0.73	59.59	0.74	60.2	0.82	58.78

H EFFECT OF EMBEDDING DIMENSION

To study the effect of embedding dimensions on the performance, we plot the MAP vs the dimension size for a subset of the datasets. As shown in figure 3, at extremely low dimensions, the MBM model consistently outperforms the MVM. This shows that the favorable inductive bias of the box space induces parsimonious representations.

I EFFECT OF HIERARCHY ON THE MARGINAL SCORE

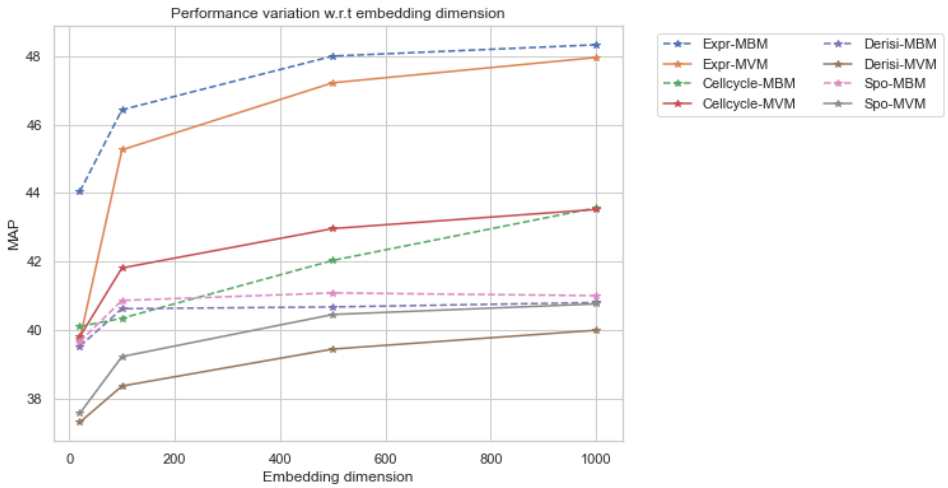
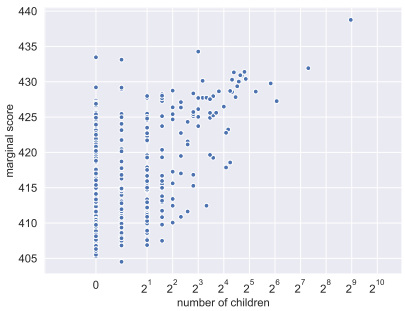
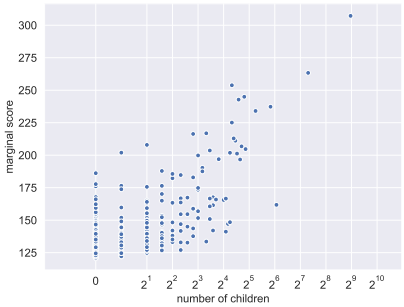


Figure 3: Above figure shows the variation of performance w.r.t the embedding dimensions for model MBM and MVM.

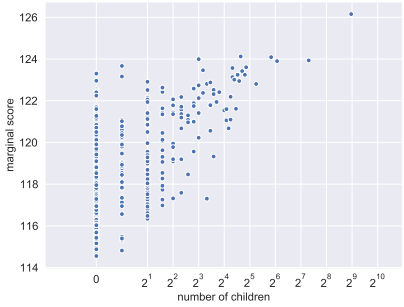
Figure 4: As discussed in Section 7, the scatter plots further show that the magnitude of box embeddings(box volume) relate to the level of generality in taxonomy.In the figure, the marginal score refers to box volume of a label and the number of children in the label space taxonomy refers to the level of generality



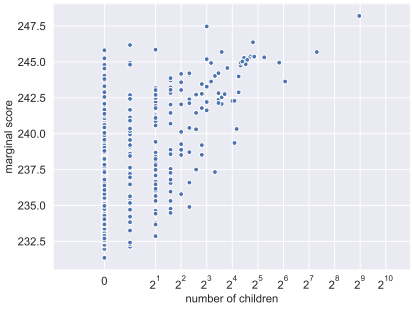
Expr



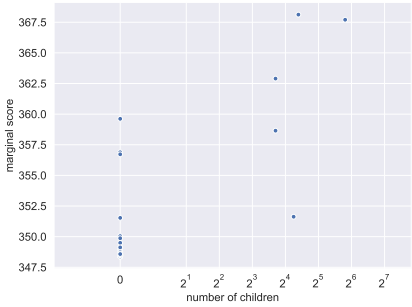
Cellcycle



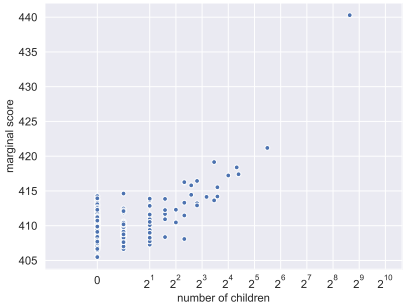
Derisi



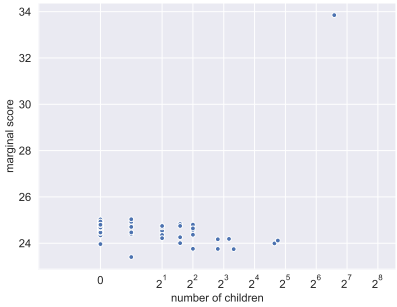
Spo



Enorn



Diatoms



Imclef07a