From multi-label learning to cross-domain transfer: a model-agnostic approach

Jesse Read

Abstract
In multi-label learning it has been widely assumed in the literature that, to obtain best accuracy, the dependence among the labels should be explicitly modeled. This premise led to a proliferation of methods offering techniques to learn and predict labels together (joint modeling). Even though it is now acknowledged that in many contexts a model of dependence is not required for optimal performance, such models continue to outperform independent models in some of those very contexts, suggesting alternative explanations for their performance beyond label dependence. In this article we turn the original premise of multi-label learning on its head, and approach the problem of joint-modeling specifically under the absence of any measurable dependence among task labels. The insights from this study allow us to design a method for cross-domain transfer learning which, unlike most contemporary methods of this type, is model-agnostic (any base model class can be considered) and does not require any access to source data. The results we obtain have important implications and we provide clear directions for future work, both in the areas of multi-label and transfer learning.

Keywords Multi-label learning · Multi-task learning · Transfer learning

1 Introduction
Consideration of multiple tasks in a single machine learning framework is becoming commonplace, and is found in different forms, and applied to a wide variety of problems. Specific cases of multi-task learning include multi-label classification, where tasks are binary classification problems, each sharing a common input domain; and transfer learning, where only one or a subset of tasks will be evaluated and the other tasks are leveraged to obtain better performance (predictive performance, computational performance, or both) under that evaluation.

The raison d'être of these areas of the literature stems from the same promise: by formulating tasks together, we can obtain greater predictive performance and/or reduced computational requirements, than if we approach each task separately. To a significant degree, the promise has been met. The multi-label literature, for example, has flourished over the previous decades with a plethora of successful learning methods that out-compete independent models.

The success of transfer learning has been equally important, but less widespread, as the source task does not form an integral part of the target task, which implies manual and computational search time (the savings of which was a primary motivator for considering transfer learning in the first place) for such a task. For this reason it mainly been successful in domains involving natural language and images, where data is large, and deep learning models are notoriously successful yet also extremely expensive to train from scratch the current common practice is essentially to import and repurpose layers from ultra large networks.

However, very many machine learning engineers and data scientists around the world use algorithms that do not fall into the category of ‘deep learning’ (e.g., recent estimates [24] of up to 600,000 monthly users of scikit-learn – a framework which contains only rudimentary neural network learning algorithms). Most models cannot be easily dissected into weight matrices (as in popular deep-learning frameworks), and even when such an expression is feasible, sharing such layers is not yet widely practiced (at least; not beyond the image and natural-language domains).
In this article we approach the problem of model-based model-agnostic transfer learning (i.e., model-driven/no source data available, and no possibility to inspect the models). The key component of our approach is to remove the assumption of task similarity, i.e., the source and target tasks may come from unrelated domains. More precisely: we do not require any more similarity between target and source domains, than between any randomly selected domains in the scope of any task having somewhere some useful application. This is an extremely challenging context for learning; because both the data and the model representation of the source task is out of reach, we face an acute information bottleneck.

We achieve our goal by building from aspects and observations that have arisen from the multi-label literature. Notably, in this domain, 1) model-agnostic approaches are commonplace (where they are often known as problem transformation or data transformation methods); 2) recent results indicate that connectivity (learning labels together) may be successful even in the absence of label dependence (similarity); and 3) a generalization to multi-label multi-task transfer learning allows us to overcome the information bottleneck through the relative richness of multi-label output vectors; to be used as input features for the target task.

In other words this paper looks to translate some important results from multi-label learning into the particular case of model-agnostic cross-domain transfer learning; and then study and develop them further in this context. The main objective is not to forward a number of empirical outcomes on benchmark data sets supporting a claim to a new state of the art method, but rather to study and demonstrate underlying mechanisms, and discuss their implications. We develop a deeper understanding on how task similarity/correlation is involved in multi-task learning, from the perspective of multi-label classification. Alongside empirical testing, we discuss the implications on different types of transfer learning, including, for example, adaptation to concept shift in data-streams.

The article is outlined as follows: In Section 2 we provide notational preliminaries and formally define the problem setting. In Section 3 we review transfer learning, multi-task and multi-label learning. In Section 4 we formulate hypotheses regarding our goal of model-agnostic cross-domain transfer learning. In Section 5 we develop a novel approach to transfer learning under the numerous constraints and motivations outlined above; named Transfer Chains. We pool together outcomes and provide discussion in Section 6, leading to number of observations and recommendations that have important implications in the general area of transfer learning. We conclude in Section 7, outlining some promising future directions.

## 2 Notation and problem setting

A task is defined as seeking to build model $h : x \mapsto y$, able to map arbitrary instances $x \in X$ (where $X = \mathbb{R}^d$) to labels $y \in \mathcal{Y}$ (generally, $\mathcal{Y} = \mathbb{R}^m$), minimizing loss $\ell(h(x), y)$. In a multi-label classification tasks, $\mathcal{Y} = \{0, 1\}^m$.

In a multi-task setting, we can explicitly denote a number of tasks $1, 2, \ldots$, e.g., $X_1, \mathcal{Y}_1$ and $X_2, \mathcal{Y}_2$ in a two-task problem. The multi-label setting can be defined either as a multi-task setting $(X_1 = X_2 = X, \mathcal{Y}_1, \ldots, \mathcal{Y}_m | \mathcal{Y}_j = \mathbb{R})$; or a single-task multi-label setting $(X, \mathcal{Y} = \mathbb{R}^m)$ or even a combination of these two arbitrarily allocating the labels among the two (or more) tasks.

Transfer learning is the specific multi-task where only one, or a subset of, tasks are considered the target task (evaluated), and other source tasks are only used in view of improving performance of the target task. For notational simplicity we denote a single source task ($S$) and single target task ($T$), but without loss of generality; indeed in experiments we consider several source tasks. We approach the task of model-based model-agnostic transfer, implying the construction of target model $h_T$, leveraging a source model $h_S$, under the unique combination of constraints that we:

1. do not have access to the source data set(s)
2. cannot inspect the source model(s) $h_S$; and
3. the source domain ($X_S, \mathcal{Y}_S$) is not related to the target domain ($X_T, \mathcal{Y}_T$).

Note that this is different from, and much more difficult than, the huge majority of transfer learning approaches which seek to find and inspect source tasks, measure their similarity to the target task, then transfer relevant parts of either into the source model, prior to (possible) fine tuning.

We are supplied with

1. A source model $h_S : X_S \rightarrow \mathcal{Y}_S$ (or, generally, models) already built to tackle some source task
2. A target data set $D_T$ of training pairs $(x, y) \in (X_T \times \mathcal{Y}_T)$

and the goal is to build a model for the target task $h_T : X_T \rightarrow \mathcal{Y}_T$ such that predictive performance

1. is initially higher (after initialization)
2. increases more quickly (during training); and
3. plateaus at higher level (when training is finished)

as per the standard transfer learning goals (see, e.g., [48]).

We investigate if such an advantage can hold after removing the common assumption that source task and target task are in some way related. This is the far less-common
far the most common form of transfer today, owing to the context of related tasks. Recall, we are specifically interested in the case where knowledge is stored in the form of existing models (rather than data sets). Data-driven approaches are enormous current popularity and renown effectiveness of deep learning. It is a straightforward, well-documented, and largely successful recipe for transfer learning, usually cast as

\[ P(Y_T | x_T) = \frac{P(x_T | Y_T)P(Y_T)}{P(x_T)} \propto P(x_T | Y_T)P(Y_T) \]

where we then use \( P(Y_T) \approx P(Y_S | x_T) \) from the posterior of the source task. And more specifically, in Bayesian methods we may place a prior over model parameters [6, 29].

Concept drift and – particularly in this context – the adaptation to concept drift [20, 58] is a central concept to a large volume of literature on learning from data streams. Even if not typically discussed explicitly by authors, adaptation to cross-domain transfer learning, but in the true sense where we do not expect any more similarity between the target task and source task than would be present between any two tasks drawn uniformly at random from all existing real-world datasets. These strict assumptions make it infeasible to compete with modern state of the art transfer-learning methods (which explicitly assume and aim to leverage similarity). But they raise interesting considerations of if/when, and how, it can be beneficial to carry out transfer learning in this setting. We investigate these considerations.

### 3 Related work

The task we approach is essentially a heavily constrained type of transfer learning. In this section we look at links to existing work on transfer learning in the literature as relevant to this setting. Table 1 places model-based transfer learning in the context of related tasks. Recall, we are specifically interested in the case where knowledge is stored in the form of existing models (rather than data sets). Data-driven approaches are enormously important and have advanced the applicability of machine learning in many areas, but do not coincide with our particular constraints. Meanwhile, Fig. 1 exemplifies the differences (and similarities) among different models.

#### 3.1 Types of transfer learning and methods

The canonical task of transfer learning in machine learning, from which many diverse problem settings arise, can be phrased as transferring knowledge from a source task to a target task. The transfer can be in the form of data (data-based transfer), a model representation (model-based transfer), or even hyper-parameters (often called in this form meta learning). A general survey and introduction is given by [35] and [48].

Deep transfer learning (DTL) (a survey by [47]) is by far the most common form of transfer today, owing to the

| Name | Target Task | Source Task | Comment |
|------|-------------|-------------|---------|
| [Semi—]Supervised learning | \( x \rightarrow y' \) | | Single task |
| Multi-label learning | \( x \rightarrow \{y'_1, y'_2\} \) | \( x \rightarrow y'_S \) | Several tasks, \( x \) in common |
| Multi-task learning | \( (x_1, x_2) \rightarrow \{y'_1, y'_2\} \) | \( x_S \rightarrow y'_S \) | Several tasks |
| Transfer learning | \( x_T \rightarrow y'_T \) | \( x_S \rightarrow y'_S \) | Source and target tasks |
| Concept-drift (input-only) | \( x_S \rightarrow y'_S \) | \( x_S \rightarrow y'_S \) | \( x \) in common, models available |
| Concept-drift (output-only) | \( x_T \rightarrow y'_T \) | \( x_S \rightarrow y'_S \) | \( y'_S \) in common, models available |
| Self-taught learning | \( x_T \rightarrow y'_T \) | \( x_S \rightarrow y'_S \) | Source task unknown |

In self-taught learning, only data from input-domain of the source task is available. In transfer learning, input data and/or source models can be available; in our case we consider only source models and no data.

Image vectors [45] and word embeddings [39, 54] have had an enormous impact on the community, especially in the areas of computer vision and natural language processing, respectively, where these representations (or ‘pre-trained’ models) are considered as a standard, or even necessary component for producing state of the art models. Pre-trained models are expanding rapidly in other areas too [25, 55].

In the probabilistic sense, transfer learning can be carried out by using the posterior distribution of the source model as a prior for another. By Bayes’ theorem,

1. Identify a suitable deep neural network source model \( h_S : X_S \rightarrow Y'_S \); where source domain \( X_S \times Y'_S \) is related in some way to the target domain \( X_T \times Y'_T \);
2. transplant one or several hidden layers (depicted as \( z_S \rightarrow z_T \) in Fig. 1(c)) from the source architecture into the target architecture; and
3. train \( x_T \rightarrow y'_T \) and – optionally – fine tune the full target network \( x_T \leftrightarrow y'_T \).

Throughout, we use the notation \( P(Y | X) \) as shorthand for \( P(Y | X = x) \); the conditional distribution – conditioned on observation \( x \); a realization of the random variable \( X \).
Fig. 1 Simplified architectures (with hidden representation, \( z \)) for (a) multi-label learning, (b) multi-task learning, and (c) transfer learning. In (c) we see two intermediate \( z \)-layers, but only one is used per task. In fact, the multi-label task is a generalization, having multi-dimensional labels; \( y_1 \in \mathbb{R}^{m_1}; y_2 \in \mathbb{R}^{m_2} \).

Drift is indeed a form of transfer learning. Namely, the target task (i.e., target concept) is a degradation of the source task wrt that task. This form of transfer learning is interesting to us, because sudden and complete concept shift (considered frequently in the literature [20]), meaning that source and target domains are unrelated, is equivalent to the scenario we are aiming to tackle. The majority of the data-stream literature suggests the best course of action is to scrap completely the earlier (source) models. Our results specifically challenge this view (discussion later in Section 6.2).

This area of the literature is also interesting to our objectives because methods for drift-detection and adaptation are often treated as black-box and model-agnostic; typically a dynamic ensemble of models is maintained, without the need to explicitly specify the base model class [22] and members are reset (simply purged and reinitialized) over time. Maintaining models in a data stream means that, almost by definition (since streams are potentially infinite, whereas resources are usually not), source data cannot be kept for the source task; thus driving model-driven adaptation.

The assumption that source data is not available has also been tackled in, e.g., the context of natural language processing by [8], an approach both multi-source and model-based (i.e., not requiring parallel data). In [12] authors consider different distributions and different domains, finding structural similarities in the form of Markov logic. Meanwhile, [7, 27] perform cross-domain adaptation focusing on similarity metrics rather than directly on domain similarity. The importance of task diversity (as opposed to the importance of task similarity which is usually of focus) is approached by [49], but they still rely on the common assumption that the main purpose is to learn a feature representation shared across different tasks (i.e., multi-task, rather than transfer learning).

Meta-learning is a kind of transfer learning that transfers from the meta source task in terms of hyper-parametrization, including particular architecture search. In other words, knowledge from the source domain itself is not directly transferred, but rather, knowledge of how to approach tasks. There are a number of modern examples (e.g., [1, 2, 9, 19, 59]). However, despite often carrying the ‘model-agnostic’ label, these methods are still gradient-descent/neural-network-centric and almost invariably assume access to source datasets (i.e., they are predominantly data-driven, rather than model-driven).

3.2 Modular learning and random projection methods

The approach we take, with no assumptions on task similarity, brings us close to that of modular random projections. Random projections and untrained layers have been used in machine learning for decades, including textbook examples of polynomial regression [26], random-basis function neural networks [17], the eccentrically-named ‘extreme learning machines’ [28] and random kitchen sinks [40], and – if we involve recurrent layers – reservoir computing [32]. Such techniques have been frequently criticized or, worse, largely ignored by the deep-learning community where gradient-descent and back-propagation are the dominant techniques yet, nevertheless, these other diverse methods continue to persist and are successful in many tasks.

In our work the source model is not a random projection in the traditional sense that the model parameters are drawn randomly, rather in the sense of the model being trained on a source task which was drawn randomly (rather than guided by a similarity metric). Nevertheless, we face potentially the same challenge of information bottlenecks: even if a large amount of data is moved easily in memory the amount of information (relevant to solving the target task) may be poor.

The model-agnostic setting of not being able to inspect source models or view the data they were trained on (the context we consider) lends itself naturally to a comparison to modular learning; building a web of structure across black-box models. Such methods are well known in the multi-label literature (e.g., by [21, 42], and references therein), where
modules are off-the-shelf classifier or regression models (a frozen neural network layer would be a particular case, as by [23]) also known as parameter-isolation. Indeed, this purely model agnostic and modular view comprises the launching pad for our study, from which we proceed in the following section.

4 Lessons from multi-label learning via chaining

Multi-label learning is a specific case of multi-task learning, and often can be approached as a specific case of transfer learning. This is particularly the case with chaining models. The multi-label literature has generally overlooked the potentially close connection to transfer learning. In this section we specifically study the area of overlap between these areas and extract insights to apply to the special case of model-agnostic transfer learning without task similarity, that we have detailed above.

As a reminder (from Section 2): In a multi-label task we learn from data \( \{(x^{(i)}, y^{(i)})\}_{i=1}^n \) where \( y^{(i)} \in \mathcal{Y} \), including the common case where \( \mathcal{Y} = \{0, 1\}^m \) (\( m \) binary labels) and, more generally, \( \mathcal{Y} = \mathbb{R}^m \) (often called multi-output regression). We seek model \( h \) to provide outputs/predictions, \( \hat{y} = h(x) \) for test examples \( x \), where target labels \( \hat{y} = [\hat{y}_1, \ldots, \hat{y}_m] \) refer to a unique \( x \)-in-common.

4.1 Multi-label learning

Following the seminal paper by [50], the multi-label literature flourished, and over the years that followed many dozens of methods were presented. A recent empirical study [3] extensively tested a range of such methods. The results can be summarized as thus: the most successful algorithmic approaches are neural network architectures and architectures of multiple decision trees. Both of these can be specifically adapted to multi-label learning, yet a number of successful approaches fall into the category of problem transformation, i.e., model agnostic; of generic multi-class (single-label) approaches which are combined together. Notable approaches include classifier chains (where each label is a distinct binary classifier, connected across a chain), and random \( k \)-label labesets (RAkEL; combinations of labels are treated by multi-class [single-label] classifiers). These trends and successes are mostly replicated in terms of multi-output regression [4, 34, 51].

A generalization of both classifier chains are RAkEL together can provided as a framework of ‘modular chaining’; see Fig. 2(b). The final contribution of this paper (for transfer learning) is based upon such a framework. Other successful methods discussed in [3] provide (in addition to the contributions of such a framework) provide either ensemble versions, or other specific adaptations (e.g., hierarchy, pruning) for computational efficiency; or, decision-tree based methods which we do not consider (because they are not model agnostic).

4.2 Chaining methods for multi-label learning

For a multi-label method to be adaptable to transfer learning, it must be the case that some labels (that may be considered as ‘source’ tasks) are learned before others (that may be considered as ‘target’ tasks). Such is the case with chaining methods (a recent survey in [42]), as well as deep architectures developed contemporaneously such as ADIOS [11], and [57] that follow a chain-like approach. Consider the illustrations in Fig. 2.

In its simplest rendition of training in chaining for the multi-label learning case, the method first learns

\[
\begin{align*}
\text{h}_1 : X &\rightarrow \mathcal{Y}_1 \\
\text{h}_2 : X \times \mathcal{Y}_1 &\rightarrow \mathcal{Y}_2
\end{align*}
\]
in that order (an order which can be inverted in the multi-label case, but not for transfer). Predictions are later obtained from the model as

\[
\hat{y}_1 = h_1(x) \tag{1}
\]

\[
\hat{y}_2 = h_2(\hat{y}_1, x) \tag{2}
\]

\[
h_2(h_1(x), x) \tag{3}
\]

We remark that under this chaining mechanism, training of \(h_2\) does not require access to \(h_1\) or any kind of joint training, because labels \(y_1 \in \mathcal{Y}_1\) are already available in the training data. Further, at test time, \(h_2\) only requires access to \(h_1\)’s predictions (not the data it was trained on, or any knowledge of its model class). Therefore, this is essentially a model-agnostic transfer from \(y_1\) to \(y_2\). There is only one reason why it does not already meet requirements for transfer learning: the fact that the source \(x \in X\) is identical to both \(y_1\) and \(y_2\); whereas we seek performance in the case that \(X_1 \neq X_2\) (inputs from both tasks come from different distributions). Our question for the following is thus: is it possible to extend the chaining approach successfully to the (task-)transfer learning scenario without any particular assumption of conditional dependence among \(y_1\) and \(y_2\).

### 4.3 The importance of modeling label dependence

Novel methods proposed in the multi-label learning literature have very frequently been motivated primarily by the goal of modeling labels together, under the assumption that this was necessary due to the fact that labels were related to each other. This assumption was largely based on intuition and encouraged by empirical results, since indeed, these methods (of joint modelling of labels) widely improved over the use of independent models.

This intuition of label relatedness was formalized in the setting of probabilistic dependence by [13] (and [15]), in particular distinguishing between global label dependence,

\[
P(Y_2|Y_1) \neq P(Y_2) \tag{4}
\]

(for any two given labels \(Y_1\) and \(Y_2\)); and conditional label dependence, after observing an input vector,

\[
P(Y_2|Y_1, x) \neq P(Y_2|x). \tag{5}
\]

But what was particularly interesting was their analysis wrt loss metrics, which revealed an apparent disconnect between the larger intuitions of the community and its empirical studies, and theoretical results. Namely, it was shown that some common multi-label metrics, such as Hamming loss, could be minimized without any consideration of label dependence, even though dozens of published results (including classifier chains) proclaimed and demonstrated advantages in practice from doing exactly that (as opposed to modeling problems individually).

Hamming loss (let us denote \(\ell_H\)) is essentially an average of label-wise losses, thus due to the summation in the average, each label can in theory be tackled individually. On the other hand, the 0/1-loss (let us denote \(\ell_{0/1}\)) compares elements such that they must match exactly\(^{2}\) to obtain 0 loss, and else a loss of 1; i.e., a label-vector prediction is treated as a single label; and thus this encourages a model of label dependence, such that accurate combinations can be chosen even at the cost of poorer marginal predictions. The relationship between these two loss metrics is given as:

\[
\ell_H(y, \hat{y}) = \sum_{j=1}^{m} [y_j \neq \hat{y}_j] \tag{6}
\]

\[
\ell_{0/1}(y, \hat{y}) = \left[\left(\sum_{j=1}^{m} [y_j \neq \hat{y}_j]\right) > 0\right] = \left[\ell_H(y, \hat{y}) > 0\right] \tag{7}
\]

(where \(\|A\| = 1\) if condition \(A\) holds). Many other metrics are possible, but these two are a good illustration on the two extremes, and many methods perform well on both.

A model that performs well on 0/1-loss should generally perform well also on Hamming loss [14, 36, 56]. But, if a dependence-structured model performs better than independent models under Hamming loss, it suggests that there must be other effects at play.

### 4.4 The effect of extra capacity

For classifier chains in particular it was hypothesised that superior performance (wrt independent models) is partly due to the additional architecture of linking labels together [16], much like the inner layers of a neural network (where labels in the chain ‘replace’ a latent/hidden-layer representation) [41]. This idea has been recently further developed by [53] (more generally for multi-output prediction) and by [42] (more specifically to classifier chain models).

This can be illustrated by the simplistic problem in Fig. 3 (adapted from [41]; and further developed in the following) and most particularly the model depicted in Fig. 3(a), which is closely related to the classic solution of neural networks for the logical \(\text{xor}\) function. However, unlike the classic example, this is done in a chain not by the use of a hidden node, rather by another label from another task – the \(\text{AND}\) function, in this case. We highlight: there is no conditional label dependence at all between these two task labels wrt the true function, i.e., \(P(y_{\text{xor}}|y_{\text{AND}}, x) = P(y_{\text{xor}}|x)\).

\(^2\) Note that the inverse of Hamming loss and 0/1 subset loss are known as Hamming score and \(\text{exact}\) match (or subset accuracy), respectively.
Fig. 3 A toy example of two labels chained together in different ways. One label represents a linearly-separable problem (AND label) and one is not (label XOR). It is not a question of label dependence only, since with a linear base learner (e.g., logistic regression), the order of connectivity matters (only model (b) is accurate) yet statistical dependence/similarity metrics are symmetric. Success owes, rather, to the additional network capacity of the AND, and its additional trainable parameter, for modeling XOR.

Essentially, in view of neural networks: each label acts as an extra hidden node, and all arrows represent additional trainable parameters, and each predictive model (yj-node) is an activation function. The difference is that, rather than employing back propagation (which is not possible due to the model-agnostic assumption; the ‘activation functions’ are classifiers though which we cannot back-propagate a gradient), we simply transfer predictive capacity from an existing trained function. Therefore, this is a kind of ‘deep’ neural network (admittedly in this example, in the most minimalistic definition possible), without the deep learning.

4.5 The effect of regularization

Even several decades ago, the James-Stein estimator showed benefit from modeling target variables together even if those variables are intrinsically independent from each other. This idea was taken up with regard to neural networks by [43] but has only recently been revived in the analysis of multi-target learning by [53] (including multi-label and multi-task classification), mentioned specifically with regard to classifier chains by [42], and also in the context of multi-task learning [18].

As [53] explain, from the machine learning viewpoint the issue is that of regularization. Modeling multiple problems together, instead of independently, means that parameters are shared, which in turn discourages overspecialization and overfitting.

Consider the task to estimate \( \mu = E[Y|x] \) (which will minimize either a squared-error metric for multi-target regression, or can be used to minimize Hamming loss in multi-label classification, where \( \hat{y}_j = [\mu_j \geq 0.5] \); cf. (6)). Suppose a single training example \( \{(x, y)\} \), which we sup-

Fig. 4 We see a (a) 1-dimensional view and (b) 2-dimensional view of a 50-dimensional problem (m = 50 outputs). We wish to estimate \( \mu = [\mu_1, \mu_2, \ldots, \mu_m] \) but we are only provided with \( \hat{y} \sim P(Y|x) \). The James Stein estimator works, in this example, by pulling the estimate \( \hat{y} \) towards \( \rightarrow 0 \), to produce \( \hat{y}_{JS} \). Even though \( \rightarrow 0 \) is not the solution, a regularized solution close to it results in better performance under squared error (E).
pose has been generated as follows:

\[ y \sim P(Y|x) \iff y = f(x) + \epsilon(x) \]

where \( f(x) \) is the deterministic element, and stochastic (Gaussian distributed) error \( \epsilon(x) \). A maximum likelihood approach will suppose that \( \hat{f} : x \mapsto y \). Hence, when \( x \) is encountered at test time:

\[ y = \hat{\mu} = \hat{f}(x) \]

Figure 4 illustrates where \( \mu = [0.5, 0.5, \ldots, 0.5] \) (for simplicity of illustration), where the James Stein estimate as

\[ \hat{y}_{JS} = \frac{1 - (m - 2) \hat{\sigma}^2}{\| y \|^2} \cdot y \]

vs the maximum likelihood estimate of \( \bar{y} = y \).

More generally, we have multiple estimates \( \{y^{(t)}\}_{t=1}^n \), because there are multiple training instances and we can bootstrap. A maximum likelihood estimate gives us unbiased estimate

\[ \bar{y} = \frac{1}{n} \sum_{t=1}^n y^{(t)} \]  

(8)

Fig. 5  Showing the regularization effect of James Stein; (a) analytically; and experimentally, as (b) performance wrt number of training examples \( n \), and (c) wrt number of labels \( m \) (empirical; synthetic data). Data is synthetic, with label concepts generated completely independently of each other. \( E_A \) is mean-squared error of estimator \( A \), which is James Stein (JS) or Least Squares (LS; i.e., maximum likelihood) estimators; meaning that values of \( (E_{LS} - E_{JS}) > 0 \) indicate that JS improves over LS (higher numbers in the vertical axes)

i.e., \( \bar{y} \approx \mu \). And (differently) the James-Stein estimator:

\[ \hat{y}_{JS} = \frac{1 - (m - 2) \hat{\sigma}^2}{\| \bar{y} \|^2} \cdot \bar{y} \]

(9)

\[ = \lambda((\bar{y}^{(t)})) \cdot \bar{y} \]

(10)

where we have explicitly denoted \( \lambda(\cdot) \) as a mechanism to regularize \( \bar{y} \), pulling it towards 0 whenever \( 1 - (m - 2) \hat{\sigma}^2 / n < \| \bar{y} \| \); shown graphically in Fig. 5(a). Note that from \( \{y^{(t)}\}_{t=1}^n \) we have \( n, \hat{\sigma}, m, \bar{y} \) (enough to fully express (9)). Shrinking estimates makes particular sense in the multi-label classification case because label vectors are relatively sparse. In other words, for any given label \( j \), the expected label \( \mathbb{E}[Y_j] \) is indeed more likely to be closer to 0 than to 1.

Figures 5(b) and 5(c) confirm the potential for empirical gains by modeling unrelated labels together (via the James-Stein estimator) even though they are generated independently from each other. However, we also observe that these gains are very slight, except for large \( m \) and, more particularly, very small \( n \). Which confirms earlier speculation (e.g., by [53]).

4.6 The ensemble effect

Many multi-label methods are developed and presented in the context of an ensemble, then compared successfully to (obtaining higher accuracy than) independent models.
Authors frequently attribute this success to the design of the novel methodology (that often includes some scheme of modeling label dependence) when in fact it may be attributed to ensemble methods. Indeed, ensembles of independent models (illustrated in Fig. 7(b)) have been recognized as unexpectedly competitive [33].

Adding more models (to an ensemble) provides the opportunity to include more predictive power (discussed in Section 4.4), as well as take into account more label dependence (discussed in Section 4.3), and it may also serve as a means of regularization. To unravel the overlap and specifically isolate the ‘ensemble effect’ we design an experiment, described as follows. We generate $m$ fully independent regression tasks (similarly as for Fig. 5), and apply independent models vs model chains, both as standalone and ensemble models, with linear-regression base models. In such a setting, extra capacity cannot be provided by the chains/joint modeling [4] and similarly, we can easily see this is also the case in the multi-output ensemble:

$$\hat{y} = \frac{1}{n} \left\{ x w^{(1)} + \cdots + x w^{(t)} + \cdots + x w^{(n)} \right\}$$
$$= \frac{1}{n} x (w^{(1)} + \cdots + w^{(t)} + \cdots + w^{(n)})$$
$$= w x$$

where, let $w/n = w^{(1)} + \cdots + w^{(n)}$, each $w^{(t)}$ representing the parameters of the $t$-th model. Note that the result is equivalent to a linear regression estimator.

Therefore, any improvement of ensembles in such a context must be due purely to the the ensemble effect. This is indeed what we observe in the results; Fig. 6. Despite that none of the four methods have any theoretical advantage among each other, there is a consistent and non-negligible difference in accuracy; with ensembles beating their standalone counterpart; and chains improving over chain-less (independent) models.

With regard to the James Stein estimator, this ‘ensemble effect’ concerns only the non-bias estimate of (8) (which appears as a component in (10)). It is known to perform a reduction of the variance component of the error. This effect is not specific to the context of a multi-output problem [30], unlike the James-Stein estimator, which will add bias to an estimate via $\lambda(\cdot)$ by considering the existence of other labels. Improvement can only be achieved by reducing instability. Chained models produce such instability simply with random chain orders [42]. This advantage is apparent even the case when there is no dependence for such a chain to model (cf. Fig. 6).

### 4.7 The effect of error correction

In Fig. 6 we not only observed that ensemble methods improved over individual models (when both have equal capacity) yet also that dependence (chained) models (RC) also provide improvement wrt independent regressors (RC) – despite no ensemble, equal capacity and the absence of any of label dependence.

Whereas a vanilla ensemble expands horizontally across multiple models and then averages their results per label; in chained models, predictions are ‘adjusted’ wrt one another in a related way to the general concept of stacking [26]. Both approaches are contrasted in Fig. 7 in the multi-target context.

This mechanism of layering predictions (such as that of Fig. 7(b)) has been credited for improvement over independent models several times in the multi-label literature, e.g., employed in [10, 31], recently discussed by [53] in a general multi-output context, and specifically in the context of chaining by [44]. However, although stacking is known generally in machine learning, the presence of multiple labels/tasks creates a different dynamic where label predictions are corrected based on earlier predictions from different labels. In some contexts it is called error correction (or, equivalently claimed to reduce error propagation) but the mechanism being referred to is the same.

In reflection of the earlier development of (10); whereas JS biases/pulls an otherwise non-biased estimate $\hat{y}$ according to term $\lambda(\cdot)$, bagging focusses only on the $\hat{y}$ term, and stacking focusses only $\lambda(\cdot)$. The relationship between the different methodologies can be expressed as follows (as earlier, $m$
Interestingly, this is exactly the opposite effect that stacking (i.e., more poor predictions for different labels) cascade along the chain and negatively influence later predictions. This refers to the idea that a poor prediction will have been evoked numerous times, e.g., [44] (and references therein); therefore we do not study this mechanism of ‘error correction’ further.

### 4.8 Combined effects: A study on effect interaction

Having so far discussed the hypothetical benefits of modeling labels (i.e., tasks) together even when they bear no measurable statistical dependence among each other, the question we ask now is – to what extent these effects can be translated into gains on real-world and benchmark multi-label datasets? (in view of translating those effects into cross-domain transfer learning).

In order to investigate, we repeat the following sequence of experiments on several standard multi-label datasets; results displayed in Fig. 8 (experimental details such as framework and the datasets are described in Section 5.4). The effects that we expect to capture are denoted in bold.

**Exp 1. Models of label dependence, non-linearity, regularization:** We compare independent models of vanilla logistic regression vs an ensemble of classifier chains with logistic regression as base model, evaluated under the 0/1 loss metric (7).

**Exp 2. Non-linearity, regularization:** As above, except we change the loss metric to a decomposable one: Hamming loss (6); for which no model of label dependence is required for optimum performance. Therefore any difference in performance must be explained by other mechanisms, namely extra capacity/non-linearity (given in the form of the classifier chain\(^3\) and regularization thereof; with variance reduction possible via both ensembles, and bias-reduction possibly via the chain structure among models. In other words: models of label-dependence should play no role here.

**Exp 3. Regularization:** As above, except now replacing logistic regression as a base classifier with multi-layer perceptrons (MLP) in both cases. MLPs provide suffi-
Fig. 8 The gain of modeling labels together (across a cascade/chain) over independent models on well-known benchmark multi-label data sets (listed in Table 2); indicating how many times better the predictive performance is:

\[ \text{Gain} = \frac{1 - \ell(\text{indep.})}{1 - \ell(\text{chains})} \]

where \( \ell(h) \in [0, 1] \) is the loss obtained by employing model \( h \).

Therefore a gain of 1.2 indicates that a chain-based model obtains 20% better predictive performance that independent models applied on the same problem. Each point is obtained as an average over 5-fold cross validation, on experiments 1.–4, as listed above.

Exp 4. **Spurious effects only**: As above but we now add regularization (in the form of a typical L2 weight penalty) to all MLP instantiations. According to our hypotheses, any gain of chaining found here should be treated as spurious; not likely to be replicated in general.

Results in Fig. 8 confirm the widely-held belief in the literature that modeling label dependence is a core issue in multi-label learning (depending on which loss metric is under consideration), and support the idea that studies and models of such dependence can lead to strong gains over a benchmark method that treats each label independently. However, in addition, as we hypothesized, other techniques can have a non-negligible influence. Consider the Music dataset (Fig. 8(d)), one of the most widely-used benchmarks in hundreds of multi-label papers: effects other than label dependence (e.g., additional architecture, and regularization) have produced on average a model around 1.2 times more performant in terms of accuracy than a typical baseline of independent models. Even gains of 1.05 times the benchmark are very interesting results when there is no reason otherwise to expect any benefit at all from label dependence. The ‘spurious effects’ of Experiment 4 under Bird can be put down to the small but apparently-not-insignificant difference between the two model classes, rather than any advantage in label-dependence modelling.

In the remainder of this paper, we look to translate these results to the domain of transfer learning.
5 Transfer learning without task similarity via transfer chains

We infer from the results of the previous section that task similarity is only one of several aspects to consider when modeling multiple tasks together. In this section we develop the setting of model-driven model-agnostic (black-box) cross-domain transfer by removing precisely the assumption on task similarity, and nevertheless specifically leveraging the chaining mechanism. In this section we outline, justify, and explain our approach of ‘transfer chains’, with some empirical experiments to examine its performance.

5.1 An outline of transfer chains

Figure 9(a) depicts a simplified representation of our proposal; the pipeline is denoted as follows:

\[ \tilde{x}_S = f(x_T) \]  \hspace{1cm} (11)

\[ \hat{y}_S = h_S(\tilde{x}_S) = \arg\max_{\hat{y}_S} \hat{P}(\hat{y}|\tilde{x}_S) \]  \hspace{1cm} (12)

\[ \hat{y}_T = h_T(\tilde{y}_S, x_T) = \arg\max_{y_T} \hat{P}(y_T|x_T, \tilde{y}_S) \]  \hspace{1cm} (13)

where \( h_S \) and \( h_T \) are the source and target task models, respectively; and

\[ f : X_T \rightarrow X_S \]

is a function mapping from the source input to target input space.

5.2 Theoretical insights: manufactured dependence

Consider Fig. 10(a) which illustrates as a probabilistic graphical model transfer-learning under the assumption that tasks \( Y_S \) and \( Y_T \) are intrinsically independent of each other (our assumption). Evidently, the decision for \( Y_T \) does not require or benefit from joint-modeling with \( Y_S \).

The key observation is that at prediction time we are not dealing with true distributions \( P \), but (as indicated in (12)) approximations \( \hat{P} \) (where, usually, \( \hat{P} \neq P \)), and \( \tilde{y}_S = f(x_T) \) is simply a transformation of observation \( x_T \), leading to the potential appearance of dependence \( \tilde{y}_S \rightarrow \hat{y}_T \); i.e.,

\[ \hat{P}(Y_T|x_T) \neq \hat{P}(Y_T|x_T, \tilde{y}_S) \]  \hspace{1cm} (14)
(shown in Fig. 10(b)).

This dependence has been manufactured artificially, when learning \( h_T \). Figure 10(c) provides an example of this mechanism in the transfer learning context (extending Fig. 10(c) from earlier, which was only a multi-label problem; this time \( P(X_S) \) and \( P(X_T) \) may have different distributions; predicting the XOR label becomes conditionally dependent on the prediction of the AND label, if minimal linear modeling is used.

However, this ‘artificial’ dependence only arises when \( h_T \) has insufficient modelling capacity or otherwise is not sufficiently effective. Otherwise, may have no influence on the decision of \( h_T \), for example if \( h_S \) performs no useful function (e.g., outputting a constant or random noise), or otherwise fails to add any predictive power above what \( h_T \) already offers. This may even occur in tasks that are intrinsically dependent, such as in low-noise scenarios when the target task can be solved easily and efficiently by \( h_T \) alone.

Hence the main assumption: that \( h_S \) provides some useful predictive capacity for some (any) task. And thus the challenge: find \( f \) such that we make use of this predictive capacity via additional feature vector \( \tilde{y}_S = h_S(f(x_T)) \) in addition to \( x_T \).

### 5.3 Considerations for the mapping-function

In Transfer Chains the mapping function \( f \) is a crucial element, since if we cannot make use of \( h_S \) in any way, there can be no justification for our approach.

The first obstacle is that both the distribution and dimensions of this source problem are different from those of the target problem. We might try to ‘disguise’ our target input instance \( x_T \) as a source instance, suggesting the use of a transport map [52] between the target and source distributions, \( P(X_T) \mapsto P(X_S) \). Alas, we have neither those distributions nor the data with which we can form an approximation (at least, not \( P(X_S) \)). And even if we did, transport maps are an expensive undertaking that do not directly address our focus point which is the predictive performance of target model \( h_T \).

Therefore, we directly seek out the functional part of the \( X_S \)-space wrt decision-making, which is near the decision boundary in classification (in the case of regression models, and more generally – the decision surface).

Fig. 11 illustrates: points \( x_T^{(i)} \) should be mapped from the right (target) space into the left (source) space, in a way that \( \tilde{y}_S = h_S(f(x_T)) \) are useful features for predicting \( y_T \). We can cast this as a generic optimization problem

\[
\hat{f}^* = \arg\max_{f \in \mathcal{F}} J(f) \tag{15}
\]

with performance measure (expected payoff/inverse loss)

\[
J(f) = J(\tilde{y}_S, y_T | x_T) \propto \mathbb{E}[\ell^{-1}(Y_T)]; \tag{16}
\]

in other words, the importance of features \( \tilde{y}_S \) generated via map \( f \) and source model \( h_S \), wrt predicting the values \( y_T \) (alongside original inputs \( x_T \)).

Since \( \tilde{y}_S \) can be treated as a vector of features for the target model \( h_T \), we can consider existing tools for assessing feature importance, e.g., filter metrics like mutual information as well as more computationally-intensive wrapper methods (involving the (re)-training of \( h_T \)). In our experiments we indeed experiment with such measures: namely mutual information, denoted \( I_{1}(f) = I(\tilde{y}_S, y_T) \); and internal cross-validation returning an average loss according to loss function \( \ell \) on predictions \( \tilde{y}_T \) vs \( y_T \), denoted as \( I_{1}(f) \) (in experiments, we use 0/1-loss (7) but this choice is problem-dependent). Both are fairly standard filter- and wrapper-approaches (respectively) for feature selection. Essentially we are trying to maximize and measure how

![Figure 11](image-url)

Fig. 11 Both the (a) Source and (b) Target data sets have been generated independently of each other (as according to Fig. 10(a) above). Plots in (a) is faded to emphasise that, from the point of view of Transfer Chains, there is no view of neither the data nor the decision boundary; only access to evaluate \( h_S(\cdot) \). The goal is to map \( x_T \) into \( X_S \) via mapping \( f : X_T \mapsto X_S \) such that using \( \tilde{y}_S = h_S(f(x_T)) \) as a feature leads to increased predictive performance for a target model \( h_T : X_T \times Y_S \mapsto Y_T \).
much ‘artificial’ dependence has been manufactured during the training process.

Since the main purpose of Transfer Chains is to leverage existing non-linearities in \( h_S \), we restrict mapping function \( f \in \mathcal{F} \) to a simple linear transformation:

\[
\tilde{x}_S = f(x_T) = U^T x_T + u
\]  

Even with this simplification, finding the best search mechanism for \( f \in \mathcal{F} \) is an extensive problem beyond the scope of this paper. In experiments we use vanilla hill-climbing methodologies; with proposal function adding Gaussian noise to \( U \) and a given budget of iterations (specified in the experiment section below).

A high-level overview of the framework is given in Algorithm 1. Fig. 12 provides a demonstration and insight on a toy example.

**Algorithm 1** Transfer Chain: A high-level overview

1: \textbf{procedure} \textsc{Train}(\( h_S, (\tilde{x}_S^{(1)}, y_T^{(1)}))_{i=1}^n \))
2: \( f \leftarrow \text{argmax}_{f \in \mathcal{F}} J(f) \) \hspace{1em} \( \triangleright \) Search; (15)
3: \( \tilde{x}_S^{(i)} = f(x_T^{(i)}) \hspace{1em} \forall i = 1, \ldots, n \) \hspace{1em} \( \triangleright \) Mapping; (17), and (11)
4: \( \tilde{y}_S^{(i)} = h_S(\tilde{x}_S^{(i)}) \hspace{1em} \forall i = 1, \ldots, n \) \hspace{1em} \( \triangleright \) Feature creation; (12)
5: Fit \( h_T: X_T \times \mathcal{Y}_S \rightarrow \mathcal{Y}_T \) using dataset \( \{(\tilde{x}_S^{(i)}, y_T^{(i)}))_{i=1}^n \)
6: \textbf{return} \( f, h_T \)

7: \textbf{procedure} \textsc{Predict}(\( \tilde{x}_T, h_S, f, h_T \))
8: \( \tilde{x}_S = f(\tilde{x}_T) \)
9: \( \tilde{y}_S = h_S(\tilde{x}_S) \)
10: \textbf{return} \( \tilde{y}_T = h_T(\tilde{x}_T, \tilde{y}_S) \) \hspace{1em} \( \triangleright \) (13)

5.4 Experimentation

We conduct experiments to investigate if Transfer Chains can offer promise in practice for model-agnostic cross-domain transfer learning. Target and source problems are given in Table 2. It is important to recall that only black-box source models are used (not the source data) for the target problems. To that end, we train multi-label random-forest classifiers on source problems and, without loss of generality, pool them together as a single source model \( h_S \). For example, the source model we have available when using Music as a target data set, is as

\[
\tilde{y}_S = h_S(f(x_T)) = [h_{\text{Scene}}, h_{\text{East}}](\tilde{x}_S)
= [\tilde{y}_{1,1}, \ldots, \tilde{y}_{1,6}, \tilde{y}_{2,1}, \ldots, \tilde{y}_{2,14}]
\]

We do not use Birds or Music in the same experiment as each other’s source or target because their domains are related; namely both are audio\(^4\). Multi-dimensional source datasets are important to our approach, in order to broaden the information flow towards target models (more source outputs implies more chained features).

Transfer learning should offer some advantage vs standalone fully data-driven target models \( h_T \). In particular, we will look for: best predictive performance, best initial predictive performance, and fastest increase in predictive performance (mentioned by [48] and others as three desiderata for transfer learning). Target model classes \( h_T \) used in the experiments are described as follows where we use the concept of step to be able to compare at different points of training:

- SLP: Single–Layer Perceptron, one logistic regression model per output trained via SGD at each step: An additional 100 iterations of SGD for each base learner;
- ECC: Ensemble of Classifier Chains, each chain in a random order at each step: A new base model (random chain) is added;

\(^4\) Even though there is no other similarity beyond the audio domain, we wish to insist later on the true cross-domain nature of the experiments
From multi-label learning...

Table 2  Multi-label data sets (d attributes and m binary labels) and their corresponding source problems used in experiments (as black-box models \( h_S \))

| Target Dataset | \( n \) | \( d \) | \( m \) | Domain                  | Source Problem |
|---------------|--------|--------|--------|-------------------------|----------------|
| XOR           | 200    | 2      | 1      | Logical/Synthetic       | AND            |
| AND           | 100    | 2      | 1      | Logical/Synthetic       |                |
| Music         | 593    | 72     | 6      | Audio/Music             | Scene, Yeast   |
| Scene         | 2407   | 294    | 6      | Image/Scene             | Music, Yeast   |
| Birds         | 645    | 260    | 19     | Audio/Birdsong          | Scene, Yeast   |
| Yeast         | 2417   | 103    | 14     | Microbiology            | Music, Scene, Birds |

The synthetic data is described above; all other data sets used in the studies of this work are available from: https://www.uco.es/kdis/mllresources/

- RAK: **Random k–Labeled Sets**, of \( k = 3 \) labels At each step: A new base model (random k-label subset) is added;
- MLP\(_v\): **Multi–Layer Perceptron** (architectures \( v \) denoted below) At each step: An additional 100 iterations of SGD;
- RLP\(_{v}\): **Random Layer Projection**: Initialized as MLP\(_v\) but not trained At each step: trial new random weights; keep them if performing better (hill climbing on \( J \));
- TC\(_{v}\): **Transfer Chain**. Subscripts and layers as per MLP. 100 initial search iterations \( f \in F \) An additional 100 iterations of SGD for base learners
- ETC: **Ensemble of Transfer Chains**, Ensemble of random TC At each step: A new transfer chain (random \( f \in F \); then 100 iterations SGD) is added.

Where \( v \) determines the architecture, such that

- \( v = 2 \): 2 hidden layers of 100 units each,
- \( v = 1 \): 1 hidden layer of 400 units, and
- \( v = 0 \): 0 hidden layers (thus SLP \( \equiv \) MLP\(_0\)).

As model-agnostic methods, we can consider any base model. We select stochastic gradient descent (SGD) since it provides a clear view of running time divided up into iterations, and thus provide a clearer picture of accuracy vs computational time invested. We use a weight decay penalty \( \lambda = 0.05 \) throughout. For all ensemble learners a new model (rather than additional iterations) is added at each step.

Experimental methodology: we run each method over 50 steps on the datasets (listed in Table 2; using a 60:40 train-test split), recording accuracy (exact match; inverse 0/1 loss) and computational expenditure at each step. Results are given in Fig. 13.

All methods are implemented in Python making use of the scikit-learn framework [37] for base models (SGD and classifier chains) where default parameters are as such unless otherwise specified above. All experiments carried out on a laptop with Intel 1.80GHz processors.

We do not compare to state-of-the-art methods in deep transfer learning which require model introspection and manipulation not compatible with our assumptions. The

![Fig. 13](https://example.com/fig13.png) The evolving performance of target models: log time (horizontal axis) wrt exact-match accuracy (vertical axis), on the test set, over 50 steps
main intention of these experiments is to provide a proof of concept supporting the discussion and development we put forward in this article, studying in particular the question of model-agnostic cross-domain transfer learning in a strict sense, and its link to multi-label learning.

Results are given in Fig. 13. We see that one of variety of Transfer Chains (either ECT, or TCₜ) performs best on all datasets (equal best in the case of Scene in terms of best performance). There are many effects to decode, from over-fitting to capacity to the influence of the underlying learning algorithm (SGD, in all cases) and the source model. The main message here is consistent: taking a model-based approach, even in a model-agnostic setting, rather than a purely-data driven approach, holds promise. Further detailed discussion is given in the following section.

6 Discussion, insights, and implications

Basic hill-climbing approaches are limited over a small number of iterations on a high number of dimensions. At the same time, although the effect is limited it can nevertheless be significant.

Recall that our main aim is not to provide a new state-of-the-art method for transfer learning, because such methods assume task similarity, available source training data, or transparency of the source model. Our approach assumes none of these. It is already clear that task similarity is an important ingredient to success of transfer learning, just as label dependence is a major factor in the success of many multi-label and multi-task methods. However we have sought (in both cases) to highlight and demonstrate that it is not the only ingredient and to develop a study around the question of other aspects.

Even if Transfer Chains does not constitute an immediately attractive option insofar as the state of the art, its performance – alongside the development and ablation studies earlier – nevertheless raises interesting points of discussion. Here we reflect upon some major implications.

6.1 The potential of model-agnostic cross-domain transfer chains

The results of Section 5.4 support the hypothesis derived from our analysis of Section 4: in the absence of task similarity, the gain for considering tasks together is arguably slight, yet non-negligible and even important in some cases.

It is true that careful fine tuning and hyper-parametrization can improve the performance of the existing data-driven approaches, but the fact remains that massive data-driven learning approaches are not tenable to those who do not have the data, and computationally expensive to those who do; unless pre-built models are gracefully supplied.

Multi-layer neural networks (represented by MLP, in our experiments) can provide enormous model capacity. However, chain transfer can reduce the need for (or at least depth of) back-propagation.

Random projections (RLP, in our study) is a cheap method to obtain extra capacity without back propagation, and are well known to the machine learning literature in many forms. However, they are not only wasteful in the computational sense, but exhibited limited success (e.g., in Fig. 13). Transfer Chains are qualitatively different, because the source models have been constructed via regularized maximum likelihood (rather than random initialization only), and are being reused at minimal cost.

This revives the question: what is similarity between two tasks? And, moreover – what is the most distant expected similarity among two models drawn randomly from the pool of all available trained models? In the least, two such randomly-selected [trained] models have greater similarity with each other than a randomly initialized model simply by the fact that they have been constructed by maximum likelihood (and, by duality, their parameters have smaller entropy).

6.2 Implications for data stream learning with abrupt concept shifts

Earlier (Section 3.1) we mentioned the case of learning from data streams; a well-known area of research where adapting to concept drift is an important challenge; often tackled by identifying and jettisoning knowledge which is assumed to be no longer relevant and learning the new concept as quickly as possible.

Our study indicates that rather than destroying old models, they should be incorporated as part of a kind of general knowledge. This is precisely the best opportunity for methods such as Transfer Chains: at the start of a new concept there is a small amount of target data and an already-established previous model in memory. Obviously, the benefit is magnified in the case of recurring concepts.

We might even suspect this mechanism is already at play in data streams methodologies. Since concept-drift detection is notoriously imprecise, models may be kept around much longer than they are directly relevant; yet still play a role of predictive capacity. Further analysis (outside the scope of this article) would be needed to further develop this hypothesis.

6.3 A fresh look at connections to human and biological transfer

There is no novelty in drawing connections to human transfer learning. But we can highlight that the kind of transfer learning in humans is, like the approach we investigate, also mainly model-driven learning. Indeed, it is now well known that the human brain is not a data storage device in the sense...
of storing original data for faithful recall later [38], at least not in the same sense as a database or csv file. On observation and interaction our brains develop neural pathways for processing information, and these pathways can be developed regardless of objective task similarity. For example, learning a foreign language may benefit achievement in mathematics [46], and the study of mathematics is commonly motivated beyond contexts where it can be directly applied. This reinforces the idea that transferability goes beyond questions of task similarity.

Chain transfer essentially repurposes a source model for which that model was not initially designed. We can find this throughout biology, and it is clearly observed in the Covid19 pandemic: global behavior was modified with planet-wide impact via a relatively simple mechanism (a virus). Admittedly, the modified behaviour was not a target model of the virus, and the reaction was even largely adversarial to its own reproduction, but it provides ample demonstration of the scale of leverage on/repurposing of a complex pre-existing structure by a comparatively simple vector.

6.4 A place for model-agnostic cross-domain transfer

With the development and study of transfer chains we pushed towards the development of more model-driven learning approaches, moving a way from purely data-driven learning. Such an approach provides a novel methodology towards model re-use and recycling, and away from single-use models involving memory-wasteful and computationally-expensive from-scratch techniques. It is worth restating the main benefit: transfer learning that is employable using off-the-shelf black-box modules, not necessarily of a particular model class or framework.

We have also acknowledged that explicitly denying aspects of task similarity comes at a cost in terms of expected accuracy obtained via transfer. It thus offers few practical improvements under a short-term viewpoint. However, under a broader and more long-term view, its potential is greater, and one could envision a large library of source models to select from. Due to the model-agnostic and model-driven nature, this could foster greater collaboration among practitioners without compromising privacy of data sources.

There are many avenues for improvement. We have so far only used relatively simplistic off-the-shelf base learners and hill-climbing methodologies. We can speculate that increasing the complexity of these methodologies as well as increasing the number of source tasks have improved the accuracy on the target task via a transfer chain, but further research would be needed to confirm this.

We have only looked at multi-label classification for the source and target tasks. Having multiple outputs is useful in overcoming an information bottleneck, but there is a vast array of task that could be considered: multi- and structured-output regression, time series, data streams, reinforcement learning, etc. In particular, we did not yet consider recurrent architectures, but such architectures are even more amenable learning regimes other than those based on gradient descent, including an area of research known as reservoir computing [32]. This would be a clear candidate for future investigation.

In transfer chains, not only does the source data not need to be similar to the target data, but the task definition does not need to be the same either.

7 Conclusion

In this article we studied, developed, and leveraged mechanisms in multi-label and transfer learning other than task similarity, which is usually seen as a fundamental motivating assumption in these areas; removing it was a central point of our study. Namely, we looked initially in the multi-label case (which involves a large corpus of literature), where task similarity takes the form of label dependence. By removing the assumption of label dependence, we brought to surface theoretical insights on other mechanisms, such as effects of regularization and additional capacity that arise naturally in connecting different tasks. And we provided an empirical study to isolate and examine these effects in practice in benchmark multi-label datasets. We showed that these mechanisms, which have been hitherto underappreciated or even disregarded completely, can significantly impact the performance in multiple task systems. We made use of these findings to develop an approach to the extremely challenging goal of model-agnostic (black-box) model-driven (no source data) cross-domain (no task similarity) transfer learning. Despite this difficult setting, with our methodology of transfer chains inspired from related concepts in the multi-label literature, that arose from our study, we were able to demonstrate attractive results on a number of standard data sets, in most cases out-competing the established methods.

We did not question the important positive association between task similarity and predictive accuracy in transfer learning. Rather, by removing an assumption of similarity, our goal was not to produce a new state-of-the-art in multi-label or transfer learning, but to provide greater understanding on the significance and potential of effects other than task similarity; we developed some theoretical insights behind these effects, and demonstrated and isolated them in practice. This is a relatively small step, but we argue it has important implications for future development. We discussed a few of these implications; in particular, we elaborated a discussion on adapting to concept drift in data streams, among other possibilities. And we identified many areas for promising future developments of model-agnostic transfer learning,
including using recurrent modular structures and reinforcement learning.

**Funding** The authors did not receive support from any organization for the submitted work. No funding was received to assist with the preparation of this manuscript. The authors have no relevant financial or non-financial interests to disclose.

**Declarations**

**Conflicts of interests** Competing interests

**Ethical standard** There are no ethical implications to discuss; the research in this paper did not involve human or animal participants. All datasets involved in the current study are listed in Table 2; the real-world benchmark data sets are available from the web link supplied in the table caption; the synthetic data/toy data sets for demonstration and illustration are described and displayed throughout the article, for example, Fig. 11.

**References**

1. Alet F, Lozano-Perez T, Kaelbling LP (2018) Modular meta-learning. In Billard A, Dragan A, Peters J, Morimoto J, (eds) Proceedings of The 2nd Conference on Robot Learning, vol 87 of Proceedings of Machine Learning Research, pp 856–866. PMLR, 29–31
2. Andreas J, Rohrbach M, Darrell T, Klein D (2016) Neural module networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 39–48
3. Bogatinovski J, Todorovski L, Džeroski S, Kocev D (2022) Comprehensive comparative study of multi-label classification methods. Expert Syst Appl 203:117215
4. Borchani H, Varando G, Bielza C, Larrañaga P (2015) A survey on multi-output regression. Wiley Int Rev Data Min Knowl Disc 5(5):216–233
5. Breiman L (1996) Bagging predictors. Mach. Learning 24(2):123–140
6. Chandra R, Kapoor A (2020) Bayesian neural multi-source transfer learning. Neurocomputing 378:54–64
7. Chehboune MA, Kaddah R, Read J (2023) Transferable deep metric learning for clustering. In IDA 2023: Advances in Intelligent Data Analysis XXI, 21st International Symposium, pp 15–28
8. Chen X, Awadallah AH, Hassan H, Wang W, Cardie C (2019) Multi-source cross-lingual model transfer: Learning what to share. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 3098–3112 Florence, Italy, July. Association for Computational Linguistics
9. Chen Y, Friesen AL, Behbahani F, Doucet A, Budden D, Hoffman M, de Freitas N (2020) Modular meta-learning with shrinkage 33:2858–2869
10. Cheng W, Hüllermeier E (2009) Combining instance-based learning and logistic regression for multilabel classification. Mach Learn 76(2–3):211–225
11. Cisse M, Al-Shedivat M, Bengio S (2016) Adios: Architectures deep in output space. In Proceedings of The 33rd International Conference on Machine Learning, vol 48, pp 2770–2779, New York, USA, 20–22 Jun PMLR
12. Davis J, Domingos P (2009) Deep transfer via second-order markov logic. In Proceedings of the 26th annual international conference on machine learning, pp 217–224
13. Dembczyński K, Waegeman W, Cheng W, Hübnermeier E (2010) On label dependence in multi-label classification. In Workshop Proceedings of Learning from Multi-Label Data, pp 5–12, Haifa, Israel, June
14. Dembczyński K, Waegeman W, Cheng W, Hübnermeier E (2010) Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In Joint European conference on machine learning and knowledge discovery in databases, pp 280–295. Springer
15. Dembczyński K, Waegeman W, Cheng W, Hübnermeier E (2012) On label dependence and loss minimization in multi-label classification. Mach Learn 88(1–2):5–45
16. Dembczyński K, Waegeman W, Hübnermeier E (2012) An analysis of chaining in multi-label classification. In ECAL: European Conference of Artificial Intelligence, vol 242, pp 294–299. IOS Press
17. Du K-L, Swamy MNS (2013) Neural Networks and Statistical Learning. Springer Publishing Company, Incorporated
18. Feldman S, Gupta MR, Frigyik BA (2014) Revisiting stein’s paradox: multi-task averaging. J Mach Learn Res 15(1):3441–3482
19. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning, pp 1126–1135. PMLR
20. Gama J, Zliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4):44:1–44:37
21. Gasse M (2017) Probabilistic Graphical Model Structure Learning?: Application to Multi-Label Classification. Université de Lyon, January, Theses
22. Gomes HM, Barddal JP, Enembreck F, Bifet A (2017) A survey on ensemble learning for data stream classification. ACM Comput Surv 50(2):23:1–23:36
23. Goutam K, Balasubramanian S, Gera D, Sarma RR (2020) Layer-out: Freezing layers in deep neural networks. SN Comput Sci 1(5):1–9
24. Grisel O (2021) All about scikit-learn, with olivier grisel. 2021. Accessed April
25. Han X, Zhang Z, Ding N, Gu Y, Liu X, Hua Y, Qiu J, Yao Y, Zhang A, Zhang L et al (2021) Pre-trained models: Past, present and future. AI Open 2:225–250
26. Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA
27. Hsu Y-C, Zhaoang Lv, Kira Z (2018) Learning to cluster in order to transfer across domains and tasks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net
28. Huang G-B, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Machine Learn Cybernet 2(2):107–122
29. Karbalayghareh A, Qian X, Dougherty ER (2018) Optimal bayesian transfer learning. IEEE Trans Signal Process 66(14):3724–3739
30. LeJeune D, Javadi H, Baraniuk R (2020) The implicit regularization of ordinary least squares ensembles. In International Conference on Artificial Intelligence and Statistics, pp 3525–3535. PMLR
31. Loza Mencía E, Janssen F (2016) Learning rules for multi-label classification. A stacking and a separate-and-conquer approach. Mach Learn 105(1):77–126
32. Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Mach Learn 34(5):216–233
33. Moyano JM, Galindo Cios KJ, Ventura S (2018) Review of ensembles of multi-label classifiers: Models, experimental study and future. AI Open 2:225–250
34. Nakano FK, Pliakos K, Vens C (2022) Deep tree-ensembles for multi-output prediction. Pattern Recogn 121:108211
From multi-label learning...

35. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
36. Park LAF, Read J (2018) A blended metric for multi-label optimisation and evaluation. In ECML-PKDD 2018: 29th European Conference on Machine Learning, pp 719–734
37. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830
38. Pinker S et al. (1997) How the mind works, vol 524. New York Norton
39. Qiu X, Sun T, Xu Y, Shao Y, Dai N, Huang X (2020) Pre-trained models for natural language processing: A survey. Sci China Technol Sci 63(10):1872–1897
40. Rahimi A, Recht B, et al. (2007) Random features for large-scale kernel machines. In NIPS, vol 3, pp 5. Citeseer
41. Read J, Hollmén J (2014) A deep interpretation of classifier chains. In IDA 2014: Advances in Intelligent Data Analysis XIII, 13th International Symposium, pp 251–262
42. Read J, Pfahringer B, Holmes G, Frank E (2021) Classifier chains: A review and perspectives. J Artif Intell Res (JAIR) 70:683–718. https://jair.org/index.php/jair/article/view/12376/26658
43. Rojas R (1996) Modular Neural Networks, pp 411–425 Springer Berlin Heidelberg
44. Senge R, del Coz JJ, Hüllermeier E (2014) On the problem of error propagation in classifier chains for multi-label classification. In: Schmidt-Thieme L, Janning R (eds) Spiliopoulou M. Data Analysis, Machine Learning and Knowledge Discovery, Cham Springer International Publishing, pp 163–170
45. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition
46. Stewart JH (2005) Foreign language study in elementary schools: Benefits and implications for achievement in reading and math. Early Childhood Educ J 33(1):11–16
47. Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018) A survey on deep transfer learning. In International conference on artificial neural networks, p 270–279. Springer
48. Torrey L, Shavlik J (2010) Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pp 242–264. IGI global
49. Triputranein N, Jordan M, Jin C (2020) On the theory of transfer learning: The importance of task diversity. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H (eds) Advances in Neural Information Processing Systems, vol 33. Curran Associates Inc, pp 7852–7862
50. Tsoumakas G, Katakis I (2007) Multi label classification: An overview. Int J Data Warehouses Min 3(3):1–13
51. Tsoumakas G, Spyromitros-Xioufis E, Vrekou A, Vlahavas I (2014) Multi-target regression via random linear target combinations. In ECML PKDD 2014
52. Villani C (2009) Optimal transport: old and new, vol 338 Springer
53. Waegeman W, Demsbięski K, Hüllermeier E (2019) Multi-target prediction: a unifying view on problems and methods. Data Min Knowl Disc 33(2):293–324
54. Wang S, Zhou W, Jiang C (2020) A survey of word embeddings based on deep learning. Computing 102:717–740
55. Weimann K, Conrad TOF (2021) Transfer learning for ecg classification. Scientific Reports 11(1):1–12
56. Wu G, Zhu J (2020) Multi-label classification: do hamming loss and subset accuracy really conflict with each other? Advances in Neural Information Processing Systems 33:3130–3140
57. Yeh C-K, Wu W-C, Ko W-J, Wang Y-CF (2017) Learning deep latent space for multi-label classification. In Proceedings of the AAAI conference on artificial intelligence, vol 31
58. Zhao P, Cai L-W, Zhou Z-H (2020) Handling concept drift via model reuse. Mach Learn 109(3):533–568
59. Zoph B, Le QV (2017) Neural architecture search with reinforcement learning

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Jesse Read is a Professor in the Computer Science Laboratory (LIX) of École Polytechnique in France since 2019, after joining as Assistant Professor in 2016. He obtained his PhD from the University of Waikato (New Zealand) in 2010, followed by a number of years of postdoctoral research at the Carlos III University of Madrid, Aalto University in Helsinki, and Télécom Paris (France). His research interests involve machine learning, and particularly multi-label classification, multi-target and structured prediction, and models for sequential data and evolving data streams, including reinforcement learning. He enjoys cross-disciplinary research and has published in the signal processing, statistics, data mining, applied data science, and machine learning communities; as well as leading and contributing to numerous open-source software frameworks, such as MEKA (for multi-label classification) and Scikit MultiFlow/River.