Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Hypothesis Transfer Learning Based on Structural Model Similarity

Kelwin Fernandes · Jaime S. Cardoso

Received: date / Accepted: date

Abstract Transfer learning focuses on building better predictive models by exploiting knowledge gained in previous related tasks, being able to soften the traditional supervised learning assumption of having identical train-test distributions. Most efforts on transfer learning consider revisiting the data from the source tasks or rely on transferring knowledge for specific models. In this paper, a general framework is proposed for transferring knowledge by including a regularization factor based on the structural model similarity between related tasks. The proposed approach is instantiated to different models for regression, classification, ranking and recommender systems, obtaining competitive results in all of them. Also, we explore high-level concepts in transfer learning like sparse transfer, partially-observable transfer and cross-model transfer.

Keywords Transfer Learning · Knowledge Transfer · Regularization · Supervised Learning

1 Introduction

Traditionally, supervised learning focuses on building models able to generalize from labeled training instances to test instances drawn from the same distribution [38]. However, since we are living in a data-driven world that is constantly changing, domain distributions change quickly in real applications,

Kelwin Fernandes
Universidade do Porto, Portugal
INESC TEC, Porto, Portugal
kafc@inesctec.pt

Jaime S. Cardoso
Universidade do Porto, Portugal
INESC TEC, Porto, Portugal
jaime.cardoso@inesctec.pt

and concepts that were valid in training time may not longer hold. Moreover, requirements, understood as the predictive task, may have changed. Thereby, classical approaches require to collect and to annotate new data, and to build new models from scratch. Since the repetitive data collection and model fitting process may become rapidly intractable in real world applications [38], it would be advantageous to transfer knowledge obtained from related problems to our target problem.

Transfer learning (TL) aims to extract knowledge from at least one source task and use it when learning a predictive model for a target task [38]. The intuition behind this idea is that learning a new task from related tasks should be easier (faster or with better solutions) than learning the target task in isolation. In this work, we focus on inductive TL, where both domains are represented by the same feature space and where the source and target tasks are different but related [38].

Pan and Yang [38] categorized previous efforts on inductive transfer into four groups depending on what is being transferred: instances, feature-representation, model parameters and relational knowledge [13]. Instance transfer consists on using data from the source task when learning the target problem [11,44,4,21], usually by means of assigning different weights to the observations. Feature-representation transfer concerns on finding a shared low-dimensional feature representation that is suitable for learning the target task [1,15,41,34]. We group these two approaches under the umbrella of data-driven transfer, where source data is re-used to train the target task. Although these approaches may seem appealing, the vast amount of training data in the source task turns the process prohibitively expensive. Analogously to Nearest Neighbors techniques in traditional machine learning, deferring the entire learning to the target-learning stage may be understood as lazy learning, i.e. deferring the actual learning until the query (target task) is made to the system. From a human-inspired point of view, this would be analogous to revisiting basic arithmetic problems when learning differential calculus or re-learning to walk when learning to run.

Thereby, transfer learning techniques (and its community) should be focused on adapting knowledge instead of data. This idea is handled by parameter transfer approaches, which rely on the idea that individual models for related tasks should share some structure (parameters or hyperparameters) [38]. In this sense, the knowledge generated from a source task is understood as the parameters that define a given model: the coefficients of a regression, the weights of a neural network, the feature hierarchy of a decision tree. A few methods have been proposed on this line [16,2,6,30,25], most of them for transferring parameters for specific models: Gaussian Processes [6], SVMs [16,30], Neural Networks [49] and ensembles [26]. Also, initialization-based models [40,27] can be included in this group, which use the source model as an initialization for the target task optimization process. This is frequently done in Neural Networks to promote convergence to local optima near the source model [27,43]. This behavior can also be achieved by applying a small number of iterations in the optimization process [40,43] or by fixing certain parameters from the source model [37,27]. However, this scheme does not guarantee that knowledge is preserved during optimization.

Hypothesis Transfer Learning (HTL) is a generalization of parameter transfer that has gained traction in the last few years [16,50,25,14,28,5,45,39,31, 29]. HTL assumes that knowledge is transferred directly from the source hypotheses. Experimental assessment [45,3] as well as theoretical properties regarding the stability of these models have been addressed by several authors in the past [5,28,39]. However, these works assumed that transfer was done between generalized linear models by regularizing the difference between source and target coefficients. In a more recent work [29], the problem of transferring knowledge from multiple source hypotheses with fast convergence using Regularized Empirical Risk Minimization was addressed.

In this paper, we generalize the HTL framework to be able to include other learning models and types of transfer. Thereby, we propose a unified structure-transfer approach that aims to transfer knowledge by regularizing the structural distance between the target and the source model. In order to illustrate the potential and flexibility of the proposed framework, we instantiate the proposed framework to four learning tasks: regression, classification, learning to rank and recommender systems (Sect. 3). Also, we explore three high-level concepts in the transfer learning area: sparse, partially observable and cross-model transfer.

The motivation for sparse transfer relies on using almost equivalent decision processes for related tasks by sparsely updating minor details in the model. For example, when an *English Checkers* player tries to play *International Checkers*, most of the decision rules learned for playing the former version are still valid for the second version. Thereby, the effort devoted to transfer the knowledge from one game to the other is spent in learning the few new rules instead of learning slight variations of the entire set of rules. Further details about this type of transfer are presented in Sect. 3.1.

On the other hand, partial transfer can be understood as having limited observability of the source model. This partial observability can be defined as restricting the set of assumed parameters by the source model that are observable when fitting the target model or by limiting the source model properties that are accessible during transfer. Being able to reuse knowledge in this context allows transfer in environments where privacy and security are important. Also, by transferring high-level properties instead of low-level parameters we can cover a wider spectrum of related tasks. We illustrate this concept in Sect. 3.2 and 3.3.

Finally, we explore in Sect. 3.3.2 an additional capability of the proposed framework, which relies on transferring knowledge between different types of models (e.g. Logistic Regression and Decision Trees, SVM and AdaBoost, etc.).

2 Transfer Learning using Structural Model Similarity

We consider the following scenario in this work. We have two learning tasks denoted by *source* and *target*. Without loss of generality, we assume that both tasks share the same feature space $X \subset \mathbb{R}^d$ and output type $Y \subset \mathbb{T}$ (e.g. regression, classification). Although this notation is an oversimplification that can be extended to other specific tasks like ranking and recommender systems, we adopt this simplistic scenario to present the method. For a given task $T \in \{source, target\}$, we have the training data $D^T \subseteq X^T \times Y^T$. Thus, the learning objective, Eq. (1), is to find the best model M^* given D^T

$$M^* = \arg\max_M \left(P(M|D^T) \right) \tag{1}$$

, where M is an instance belonging to the space of models. Applying the Bayes theorem and a monotonous logarithmic transformation, Eq. (1) can be transformed to Eq. (2) with the same solution.

$$M^* = \arg\max_M \left(log(P(D^T|M)) + log(P(M)) \right) \tag{2}$$

In this sense, Eq. (2) can be understood as finding the model that maximizes the (weighted) tradeoff between fitting the data (*dataFitness*) and having a desired structure (*modelFitness*).

$$M^* = \arg\max_M \left(dataFitness(M, X^T) + \lambda \ modelFitness(M) \right) , \ \lambda \geq 0 \tag{3}$$

In a transfer learning context, *dataFitness* is only associated to the model performance on the target data. While in classical learning settings the *modelFitness* term gives priority to simple models, we propose to prioritize models with high similarity with the model obtained using the source data only:

$$M^* = \arg\max_M \left(dataFitness(M, X^{target}) + \lambda \ similarity(M, M^{source}) \right) , \ \lambda \geq 0 \tag{4}$$

Eq. (4) presents a unified framework for hypothesis-transfer that can be instantiated to several predictive models given:

- A function that defines the similarity between the knowledge synthesized in the target model and the one in the source model.
- An optimization framework that allows introducing the regularization term using the structural similarity function.

The analogous minimization problem can be defined using a data-driven loss function and a model-driven dissimilarity function.

As defined in Eq. (5), this framework can be extended to support transfer from multiple sources $S = \{s_1, s_2, \ldots, s_n\}$ in a straightforward manner, where λ_j denotes the regularization level associated to the source task j.

$$M^* = \arg\max_{M} \ \mathit{dataFitness}(M, X^{target}) + \sum_{j=1}^{n} \lambda_j \ \mathit{similarity}\,(M, M^{s_j}) \qquad (5)$$
$$\text{where } \lambda_j \geq 0, \ \forall j \in \{1, \ldots, n\}$$

Thereby, instead of transferring data from the source task as done by previous methods in the literature, knowledge is transferred through the model structure. Since a predictive model is a succinct representation of the data, the proposed approach is an efficient way to introduce knowledge obtained from the source task without resorting to the source data. Therefore, the proposed approach is also useful in scenarios where source data is unavailable at transfer time and in online learning settings.

3 Instantiations and Experimental Evaluation

In this section, several instantiations of the proposed framework to different models are presented. These models explore general learning tasks usually studied in the literature: regression, classification, learning to rank and recommender systems. Moreover, we validate high-level transfer concepts in each one of them in order to prove the flexibility of the proposed framework. For instance, concepts like sparsity, partial observability of the source model and cross-model transfer are analyzed.

For readability, the experimental evaluation is presented along with the model instantiation. Also, the following baselines [12,30] are used for comparison purposes:

- **Target-only:** the target model is learned using the target data only. This baseline is analogous to ignoring the source task and building a target model from scratch.
- **Weighting (W):** the target model is learned using a weighted combination of source and target data. The weight associated to each class is trained using nested cross-validation.
- **Extended (Ext):** the target model is learned using the target data extended with the prediction obtained by the source model. For classification tasks, the estimated probability is considered instead of the final class.

In order to avoid overfitting to the training data in these settings, all the baselines are regularized using their corresponding penalty terms (e.g. L_1, L_2).

In the experimental evaluation, data was split using a stratified training-test partition (80-20). Then, in order to validate the model performance on different stages of the data acquisition process, the training set was randomly subsampled in 10 nested subsets with several sizes $(10\%, 20\%, 30\%, \ldots, 100\%)$. Each experiment was repeated 30 times varying the test partition. For reproducibility purposes, source code, training-test partitions and the individual

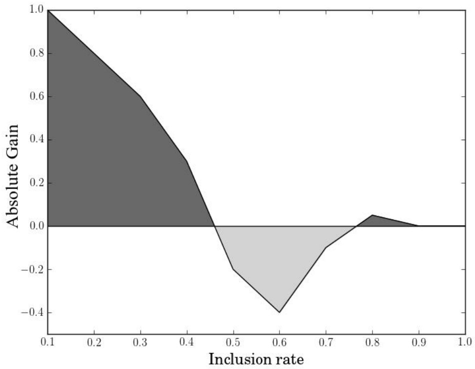

Fig. 1: The Signed Area under the Gain Curve (sAUC) is the sum of the area of all positive transfer regions (dark areas) minus the area of the negative transfer regions (light areas).

assessment per inclusion rate are made available[1]. The regularization factor and all the remaining intrinsic meta-parameters were learned using nested Stratified K-fold cross-validation (K = 3) over the training set. The same parameter fine-tuning scheme was conducted for all the baselines and proposed methods.

For each method, the absolute gain is measured when compared with the **Target-only**. Thus, positive gain reflects positive transfer and, analogously, negative gain reflects negative transfer. Figure 1 illustrates this concept, where dark regions represent positive transfer and light regions negative transfer. Many papers in the literature confine the results to a predefined training-test partition [11,16,26], restricting the comparison of the methods to specific stages of the data acquisition process. Other methods enumerate the performance when varying training set sizes [21,30,27]. In order to provide useful feedback about the actual performance of the method through the entire spectrum of data acquisition, the normalized Discounted Cumulative Gain (nDCG) was considered. nDCG is frequently used in *learning to rank* tasks to compare different rankers and, to the best of our knowledge, has not been used for assessing transfer learning. Its adequacy to TL stands as follows. If we consider a sequence of nested training sets, the main focus of TL is to increase the performance specially on the smallest sets [48], where data is scarce. Thereby, considering the aforementioned nested training subsets, the gain obtained by considering the i-th training subset is analogous to the relevance of the item ranked at position i in a ranking setting. Wang et al. [47] show that nDCG can decide in a consistent manner the best ranker in every pair of substantially different ranking functions. Eq. (7) defines a continuous version of the nDCG $\in (-\infty, 100]$ over the space of percentage inclusion of training data,

[1] URL available after decision

where $BE(x)$ and $ME(x)$ are the error of the baseline strategy and of the model of interest when considering $x\%$ of the data. ME^* is the zero constant representing the error of the best model assuming a noiseless training set. Given that it would be computationally intractable to build all possible training sets, we considered an approximation of Eq. (7) using the trapezoidal rule of the aforementioned partitions.

$$\mathrm{DCG}(BE, ME) = \int_0^{100} \frac{BE(x) - ME(x)}{log_2(x+1)+1} dx \tag{6}$$

$$\mathrm{nDCG}(BE, ME) = 100 \frac{DCG(BE, ME)}{DCG(BE, ME^*)} \tag{7}$$

In order to simplify the assessment of the proposed methodologies, we validate the performance of the proposed methologies with single source-target settings.

3.1 Regression

In this section we instantiate the proposed framework to the Linear Regression model. In Eq. (8) we adopt the well known Elastic Net (EN) loss function, where ω^s and ω^t stands for the source and target coefficients respectively and $\| \|_p$ is the p-norm of the coefficients. In this case, the model similarity is instantiated as the distance between the target and source coefficients. In order to allow concept drift, the independent term is not regularized.

$$J_{X,y}(\theta) = \sum_{i \in N} \left(y_i - X_i^\top \cdot \omega^t\right)^2 + \lambda \left(\alpha \parallel \omega^t - \omega^s \parallel_1^1 + (1-\alpha) \parallel \omega^t - \omega^s \parallel_2^2\right) \tag{8}$$

The target model ω^t can be defined in terms of the source model as $\omega^t = \omega^s + \Delta$ and, considering the residuals of the source model on the target task, $\epsilon_i = y_i - X_i^\top \cdot w^s$, the optimization objective defined in Eq. (8) can be rewritten as stated in Eq. (9). Thereby, the optimization objective is equivalent to fitting a classical regularized linear regression to the residuals.

$$J'_{X,\epsilon}(\Delta) = \sum_{i \in N} \left(\epsilon_i - X_i^\top \cdot \Delta\right)^2 + \lambda \left(\alpha \parallel \Delta \parallel_1^1 + (1-\alpha) \parallel \Delta \parallel_2^2\right) \tag{9}$$

Sparse transfer is an interesting concept that can be achieved using this framework and the proper regularizer. The intuition behind this idea is that an intelligent agent should be able to reuse a decision strategy obtained from a related source task by changing a small number of details instead of updating the entire model. In this specific instantiation, such property can be obtained by using an L_0 or L_1 regularizer. Since Eq. (9) is agnostic about the source coefficients distribution, encouraging sparsity in the transfer stage induces sparse

Table 1: Comparison of Regression models using different transfer strategies: Ridge (L_2), Lasso (L_1) and ElasticNet (EN). Performance is measured using Mean Absolute Error (MAE).

Dataset[32]	**W**	**Ext**	**Proposed**		
			$\mathbf{L_2}$	$\mathbf{L_1}$	**EN**
Automobile Gas/Diesel	6.75	3.07	**22.01**	**19.31**	**17.74**
Solar Flare M/C	**11.83**	0.14	1.67	-0.29	0.47
Parkinson Men/Women[46]	5.32	-0.42	**5.74**	-7.60	-7.48
Students P1/M1 [10]	0.30	0.29	**2.06**	0.09	0.25
Students P1/P2 [10]	**4.08**	1.00	3.83	**4.30**	**4.23**
Wine Red/White [9]	-0.60	-0.06	**0.15**	-2.50	-0.92
Wine White/Red [9]	**0.42**	-0.16	**0.52**	-0.88	-0.21

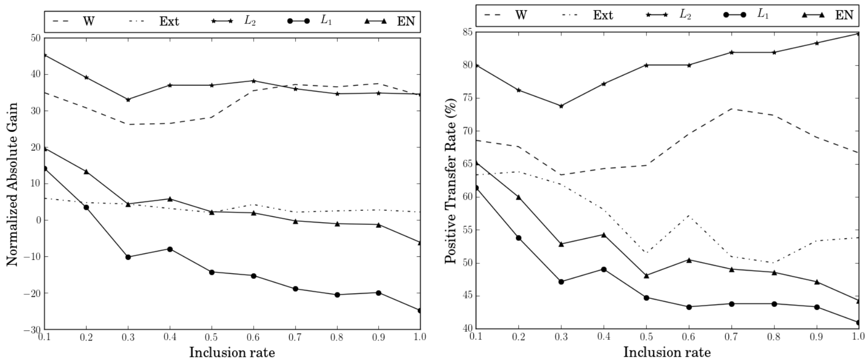

Fig. 2: Average gains (left) and positive transfer rates (right) with nested training sets on regression tasks

differences between the source and target model instead of sparse coefficients per se.

In the experimental assessment, three regularizers for the transfer step were used: Ridge ($\alpha = 0$), Lasso ($\alpha = 1$) and the general Elastic Net ($0 \leq \alpha \leq 1$).

Table 1 shows the results obtained in several datasets. Gain was measured in terms of decrease in the Mean Absolute Error (MAE). Hereafter, the best scores are presented in bold, as well as all statistically identical scores, using a paired difference Student's t-test with a 90% confidence level. The best results were obtained by at least one of the proposed regularization schemes on most datasets (see Table 1). As can be seen in Fig. 2, the proposed strategy using L_2 normalization dominates the other curves, specially in the smallest partitions where the larger gains are achieved. As expected, while all the models achieved positive transfer on the first partitions, as we move towards the full inclusion of the training data, the gains become negative.

It is well known that, when evaluated using only prediction quality, Ridge tends to be superior to Lasso (and Elastic Net). Thus, results are aligned with this. An interesting behavior can be observed by studying the results obtained in the *Students Performance* [10] dataset, where we explored predicting the students grades on maths (M) and Portuguese (P). In the case that knowledge was transferred between different courses in the same academic period (P1/M1) the L_2 regularizer achieved the best results. On the other hand, when knowledge was transferred between the same course but using different periods (P1/P2), a sparse transfer strategy obtained the best results. These examples validated the motivation behind sparse transfer, which focuses on changing a small subset (sparse) of properties of the model when the tasks are strongly related.

3.2 Classification

The proposed transfer learning framework is instantiated to Linear Support Vector Machines and to the AdaBoost classifier in this section. Although other classifiers can be adapted to this framework, these models are suitable to explore the idea concisely. For example, Artificial Neural Networks may be regularized using the coefficients difference and Decision Trees by considering the edit distance between the source and target trees.

Also, we explore in this section the concept of partial transfer, allowing to selectively transfer knowledge from the source model. Partial transfer can be understood as improving the model performance on the target task by using a partially observable source model. This can be done by considering regularization schemes that explore high-level properties of the model instead of its actual state (i.e. assumed values). This capability is specially important in some scenarios, where unlimited access to the model parameters is not possible due to privacy and security concerns (e.g. health and biometrics applications). In these cases just high-level properties of the model are available. Also, regularizing high-level properties of the models allows transfer between less similar tasks. Thereby, even when the source model is fully observable, it could be interesting to study partial transfer mechanisms.

3.2.1 Support Vector Machines

Similarly to the Linear Regression, the proposed framework can be instantiated to linear Support Vector Machines (SVM) considering the difference between the source and target coefficients. This idea was previously explored for Structural SVMs by Lee and Jang [30] and in a multitask learning setting by Evgeniou and Pontil [15]. In both cases, the dual formulation is used. Instead, we use the soft-margin primal formulation with hinge loss (cf. Eq. 10) using stochastic subgradient descent [42]. Also, some authors explored this problem from a theoretical point of view to show its stability [28, 39].

$$\arg\min_{\omega^t} \ \frac{1}{N}\sum_{i=1}^{N} \max\left(0, 1 - y_i X_i^{\top} \cdot \omega^t\right) + \lambda \parallel \omega^t - \omega^s \parallel_2^2 \tag{10}$$

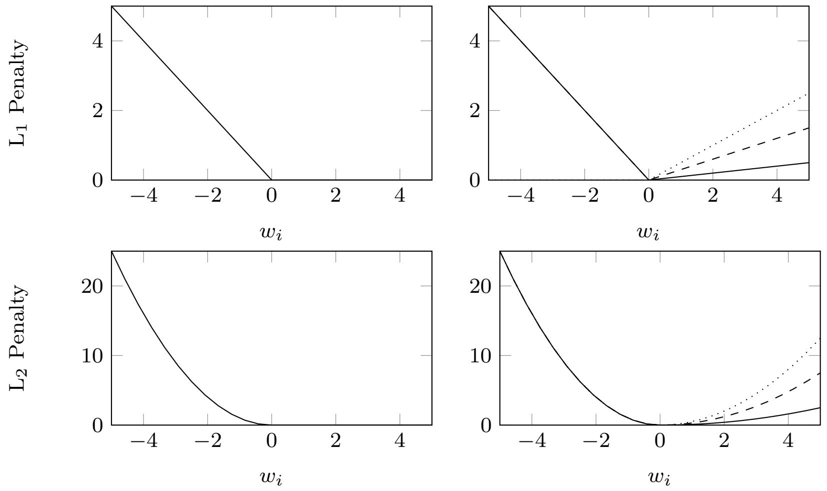

Fig. 3: Sign regularization factors assuming $w_i^s > 0$. First row illustrates the penalization using L_1 regularizers ($p = 1$) with same-sign uncontrolled penalty on the left and with different α values on the right (0.9 - solid, 0.7 - dashed, 0.5 - dotted). Second row is analogous to the first row but using L_2 penalty ($p = 2$).

In order to validate the concept of partial transfer, we explore the idea of transferring the contribution direction of each feature (i.e. coefficient sign) instead of its importance in the source task (i.e. coefficient magnitude) [18]. This type of transfer is not only pertinent in partially observable settings but also allows positive transfer between tasks that are only slightly related. Eq. (11) defines a way to regularize the coefficient sign.

$$\delta_p(\omega^t, \omega^s) = \sum_{i=1}^{d} \max(0, -\omega_i^t \cdot \text{sign}(\omega_i^s))^p \tag{11}$$

Although this regularizer is able to control the sign change between source and target task, it does not establish any type of control on models with large coefficients with the same sign. Thereby, we include the classical Tikhonov regularization (see Eq. (12)). Fig. 3 illustrates the behavior of two particular instances of the proposed regularizer with $p = 1$ and $p = 2$.

$$\Delta_{p,\alpha}(\omega_i^t, \omega_i^s) = \alpha\delta_p(\omega_i^t, \omega_i^s) + (1-\alpha) \parallel \omega^t \parallel_p^p,\ 0 \leq \alpha \leq 1 \tag{12}$$

The proposed regularizer is based on the Hinge loss traditionally used in the optimization of Support Vector Machines. In this sense, the particular case when $p = 2$ is a smooth version that allows gradient computation on its entire domain. Thereby, it does not introduce further complexity to the loss function defined in Eq. (10).

Table 2: Comparison of classifiers using different transfer strategies: SVM with Structural regularization (SVM), SVM with Structural Sign regularization (S-SVM), SVM with Structural Sign-mixed regularization (αS-SVM). Performance is measured using accuracy.

Dataset [32]	**W**	**Ext**	**Proposed**		
			SVM	**S-SVM**	**αS-SVM**
Echocardiogram (fluid)	0.16	-3.76	3.77	5.22	**9.21**
Glass (RI high)	**13.48**	0.37	**2.08**	4.07	**12.86**
Hepatitis (No Histology)	9.05	-1.89	**17.54**	8.84	9.62
Car Evaluation high/med	-1.00	-1.29	1.99	**2.40**	**3.51**
Pima Indian (old)	-9.91	**2.83**	**3.32**	**6.17**	**3.58**
Contraceptive (Working)	-7.94	0.78	5.75	**9.13**	**9.84**
Ionosphere Ft. 29 (High)	**15.96**	2.29	10.02	5.57	4.87
Wine (White/Red)	-10.96	1.21	-0.46	11.76	**18.74**

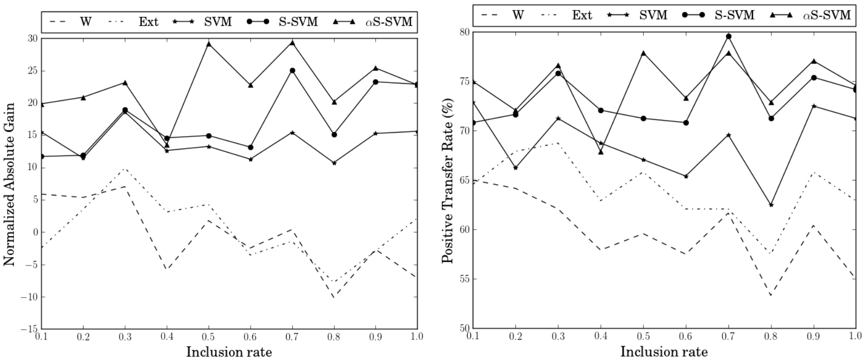

Fig. 4: Average gains (left) and positive transfer rates (right) with nested training sets on classification tasks using SVMs

On the other hand, when $p = 1$, the derivative at $\omega_i = 0$ is non-deterministic. However, the subgradient at $\omega_i = 0$ can be computed, inducing a subgradient descent optimization strategy. This type of regularization would also induce sparse transfer, a concept previously studied in Section 3.1. In this work, we only present results for the smooth version of the proposed regularizer.

Table 2 shows the results for SVMs. The proposed schemes achieved the best results in most datasets using the proposed structural similarity transfer. Moreover, the *sign* regularization scheme obtained better results than the difference-based in several datasets. In this sense, structural regularization TL offers a competitive and efficient framework for transferring knowledge from SVM. As was validated in the experimental evaluation, considering partial-transfer schemes improves the transfer gains between more dissimilar tasks. For example, comparing the gain obtained in the *Wine* dataset by the partial transfer strategy was higher than in the *Hepatitis* dataset. This suggest that predicting life expectancy of patients with and without histology is more related than predicting quality of different types of wine. Thereby, using a strong regularization with full observability achieves the best performance in the latter while using a more flexible regularization (partial observability) achieves the best performance in the former.

For this instantiation, the gain achieved by the models as we collect more data doesn't decrease. In general, we may observe that the gains achieved by the models with partial observability dominate the other curves (see Fig. 4). Thereby, it was validated the relevance of transferring partial knowledge instead of promoting low-level similarity between source and target models.

3.2.2 AdaBoost

In this case, we instantiate the proposed framework to the Discrete AdaBoost model [20]. As typical, we used unidimensional decision thresholds as weak learners. However, the concepts explored in this section can be easily extended to other types of estimators.

In the AdaBoost model two type of concepts can be transferred from a source model: the weak estimators and their associated importance. We regularized the weak estimators by encouraging similar decision thresholds, considering that the target model can probabilistically choose a learner from the pool of source weak learners or can create a new estimator from scratch. On the other hand, in order to regularize the relative importance of each estimator, we encourage closeness between the iteration at which each estimator was chosen in the source and target tasks. This type of regularization has the secondary advantage of promoting similar updates to the weight distribution associated to the training set in both, the source and the target task.

In this sense, at each iteration of the AdaBoost training algorithm, we select the estimator (f, t, d, i) that minimizes the tradeoff between the exponential loss, traditionally used in AdaBoost, and the regularizer defined in Eq. (13), where f is the feature of interest, t is the threshold value, $d \in \{-1, +1\}$ is the estimator output when the thresholding condition is satisfied, i is the iteration where the estimator was included in the ensemble and N is the maximum number of estimators.

$$D(e, \text{pool}) = \underset{p \in \text{pool}}{\arg\min} \left(\alpha D_{thrs}(e, p) + (1-\alpha) D_{order}(e, p)\right) \tag{13}$$

$$D_{thrs}((f^t, t^t, d^t, i^t), (f^s, t^s, d^s, i^s)) = \begin{cases} |t^t - t^s| & , \text{ if } f^t = f^s \ \wedge \ d^t = d^s \\ 1 & , \text{ otherwise} \end{cases}$$

$$D_{order}((f^t, t^t, d^t, i^t), (f^s, t^s, d^s, i^s)) = \begin{cases} \frac{|i^t - i^s|}{\text{N}} & , \text{ if } f^t = f^s \ \wedge \ d^t = d^s \\ 1 & , \text{ otherwise} \end{cases}$$

Given that D_{thrs} denotes the similarity between the decision thresholds in the source and target hypotheses and D_{order} denotes the similarity between the feature relevances, the α parameter controls the model observability. By setting $\alpha = 1$ we will observe the decision thresholds and ignore the importance.

Table 3: Comparison of classifiers using different transfer strategies: AdaBoost with observable thresholds and order (Full), AdaBoost with observable thresholds (Thres) and Adaboost with observable order (Order). Performance is measured using accuracy.

Dataset [32]	**W**	**Ext**	**Proposed**		
			Full	**Thres**	**Order**
Echocardiogram (fluid)	**10.55**	-0.28	-0.07	2.38	-4.53
Glass (RI high)	**4.04**	**3.60**	**2.63**	**2.14**	**-5.56**
Hepatitis (No Histology)	-12.66	**-2.76**	**-4.36**	**-6.70**	-10.96
Car Evaluation high/med	-37.81	**17.83**	2.51	**19.61**	17.56
Pima Indian (old)	**0.32**	-4.03	**-0.06**	**-1.77**	-2.87
Contraceptive (Working)	**14.90**	0.63	1.00	4.74	2.72
Ionosphere Ft. 29 (High)	**28.66**	12.45	5.13	16.47	13.13
Wine (White/Red)	-0.38	-0.54	**0.30**	-0.74	-0.26

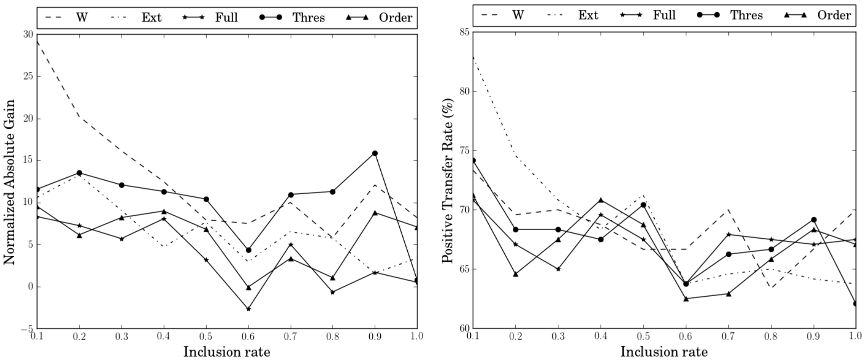

Fig. 5: Average gains (left) and positive transfer rates (right) with nested training sets on classification tasks using AdaBoost

Conversely, using $\alpha = 0$ will ignore the thresholds but will encourage the target model to choose the features in a similar order. In the experimental assessment we considered models with 50 estimators. Table 3 shows the results for the proposed regularizers. While the proposed strategy achieved positive transfer in most cases, the weighting strategy achieved the larger gains in the smallest partitions on average (see Fig. 5).

Partial observability of the decision thresholds obtained better performance than transferring the selection order of the weak estimators.

3.3 Learning to Rank

Learning to Rank in combinatorial domains has become a trendy topic in recent years due to the growing number of applications involving the prediction of structured preference data. Examples of applications where predicting rankings is crucial are found in information retrieval (e.g. search engines) and

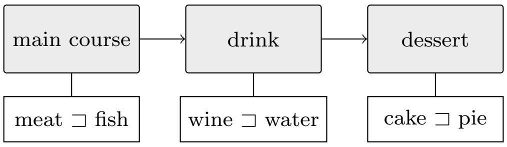

Fig. 6: Illustration of a unconditional Lexicographic Ranker with three attributes

recommender systems. Learning to rank strategies can be categorized according to their input type into pointwise, pairwise and listwise techniques. In this section we consider pairwise rankers, which rely on deciding which observation, if any, is better in a given pair.

3.3.1 Lexicographic Orders

Here, we instantiate the proposed TL framework to lexicographic orders [19], which compactly express the order between any pair of observations. Instantiating other ranking models like RankSVM [24] is very straightforward using the techniques explored in previous sections. In order to simplify the presentation of the structural similarity function between lexicographic orders, we limit the scope of this work to unconditional/linear lexicographic orders – LO – (e.g. LexRank [19]) with binary features. LO can be understood as a total order of the attributes and of their respective values. Thereby, given a ranking task with D binary attributes, a LO model M can be understood as a pair $M =< A, V >$ where $A : \mathbb{N}^{\leq D} \to \mathbb{N}^{\leq D}$ is a bijective function that indicates the relevance of each feature and $V : \mathbb{N}^{\leq D} \to \mathbb{B}$ is a function that defines the preferred value for a given feature.

Fig. 6 illustrates an instance of a linear Lexicographic Ranker with three features: main course, drink and dessert. The attribute domains are {meat, fish}, {wine, water} and {cake, pie} respectively. To predict the ordering of two options using such model, the two observations are compared through the model on a cascade manner (using the feature relevance), until they differ in a given feature. The order direction is dictated by the preferred value for that feature. For instance, using the model illustrated at Fig. 6, the following is a valid ordering of options:

$$(\text{meat}, \text{wine}, \text{cake}) \sqsupset (\text{meat}, \text{wine}, \text{pie}) \sqsupset (\text{meat}, \text{water}, \text{pie}) \sqsupset (\text{fish}, \text{wine}, \text{cake})$$

Linear LO are of interest due to their high interpretability. Despite the existence of lexicographic rankers with higher expressiveness, we limit the scope of this work to this type of ranker to simplify the regularizer definition. The ideas explored in this section can be extended to conditional LO [17].

We define the distance between two LO as the weighted sum of the normalized Kendall tau distance between the attribute ordering and the number of attributes with different preferred values in Eqs. (14)-(16). This distance can be extended to conditional Lexicographic Orders [7,17] by considering the edit distance between trees instead of the Kendall tau distance.

$$dist_{\alpha}(\langle A^s, V^s\rangle, \langle A^t, V^t\rangle) = \alpha K(A^s, A^t) + (1-\alpha)P(V^s, V^t) \tag{14}$$

$$K(A^s, A^t) = \binom{D}{2}^{-1} \sum_{1\leq i<j\leq D} [A^s(i) < A^s(j) \not\equiv A^t(i) < A^t(j)] \tag{15}$$

$$P(V^s, V^t) = \frac{1}{D}\sum_{i=1}^{D}[V^s(i) \not\equiv V^t(i)] \tag{16}$$

Given the discrete nature of LO, greedy algorithms have been used in the literature to obtain models fitted to data [19]. In our experimental evaluation the regularization term is introduced as part of the objective function in a local search strategy. The neighborhood is defined by all possible swaps of consecutive attribute pairs and by changing the preferred value of each feature. A first-best approach was conducted for choosing the next neighbor to be expanded. Table 4 shows the results obtained for this task. Since local search rapidly converges to local optima, two independent runs were executed starting from different initial solutions. These solutions were generated using the greedy LexRank algorithm proposed by Flach and Matsubara [19] on the source and target data separately. Besides the instantiation with full-knowledge transfer, which was denoted in Table 4 as Comb ($\alpha = 0.5$), two instances with partial observability of the model structure were considered: Priorities ($\alpha = 1$) and Preferences ($\alpha = 0$). Performance is measured in terms of correctness [8] (see Eq. (17)), which considers the balance between concordant (C) and discordant (D) predicted pairs. As was observed with the classification models, using partial transfer improved the model performance in most datasets. Morever, as can be seen in Fig. 7 the gains achieved by the proposed strategies are consistently higher than the ones achieved by the other methods in the literature, being able to achieve positive transfer in more than 80% of the cases.

$$CR(\sqsupset, \sqsupset_{*}) = \frac{|C| - |D|}{|C| + |D|} \tag{17}$$

3.3.2 Cross-model Transfer: from RankSVM to Lexicographic Orders

In this section we explore another capability of the proposed transfer framework: transferring knowledge between models with different nature. In order to do this, we can use a regularizer that relies on high-level structural properties of the model instead of model specific parameters. We explored some intuitions behind this idea in the sign regularization for the SVMs. In this section, we will transfer information from the RankSVM model [24] to LO. We can use linear SVMs in the context of rankings by transforming the decision function $f(a) < f(b)$ into $g(a - b) > 0$. In this sense, the final linear

Table 4: Comparison of Ranking models using different transfer strategies: Priorities (Prior), Preferences (Pref) and Combined (Comb). Performance is measured using correctness.

Dataset[32]	**W**	**Ext**	**Proposed**					
			LexRank-LexRank			**RankSVM-LexRank**		
			Prior	**Pref**	**Comb**	**Prior**	**Pref**	**Comb**
Lenses (Hyper./Myope)	**37.38**	33.87	**42.71**	33.71	32.24	11.37	12.39	-2.24
T.A Regular/Summer	11.77	10.81	**17.53**	12.68	7.40	11.73	8.30	9.01
Acute Infl Urin./Renal	28.19	12.32	28.08	28.08	**31.08**	28.08	28.08	**30.94**
Servo A/C	1.83	-4.96	13.97	11.39	**20.49**	7.67	-5.27	7.56
Mammographic (Old)	3.44	1.49	1.70	0.19	1.67	**9.34**	7.91	**9.18**
Contraceptive/Std. Liv	-7.50	-1.26	**-0.17**	-0.45	-0.48	-0.91	-1.02	-1.07

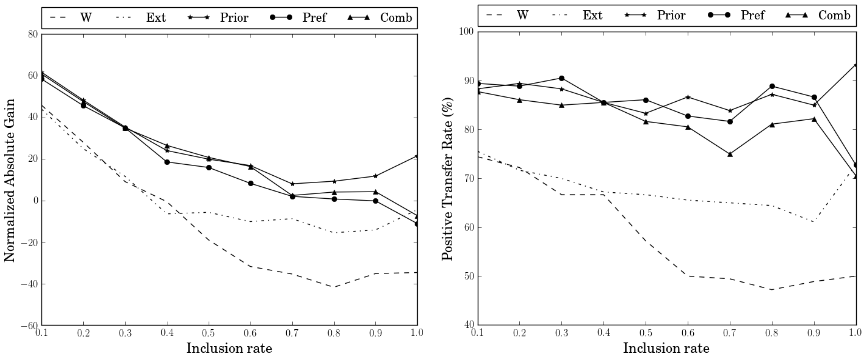

Fig. 7: Average gains (left) and positive transfer rates (right) with nested training sets on ranking tasks using LexRank

SVM model will induce a decision boundary defined by $\omega^{\top}(a-b)>0$. Then, for binary variables, the absolute-valued magnitude of each coefficient can be understood as the feature relevance and the coefficient sign as the preferred value of each feature in the lexicographic orders. Thereby, we can use the regularizer formalized in Eq. (20) to transfer knowledge from RankSVM to linear Lexicographic Rankers.

$$dist_{\alpha}(\omega^s, \langle A^t, V^t\rangle) = \alpha K(\omega^s, A^t) + (1-\alpha)P(\omega^s, V^t) \tag{18}$$

$$K(\omega^s, A^t) = \binom{D}{2}^{-1} \sum_{1\le i<j\le D} [|\omega_i^s| > |\omega_j^s| \;\not\equiv\; A^t(i) < A^t(j)] \tag{19}$$

$$P(\omega^s, V^t) = \frac{1}{D}\sum_{i=1}^{D}[(\text{sign}(\omega_i^s)) > 0) \not\equiv V^t(i)] \tag{20}$$

Table 4 shows the behavior of the cross-model transfer between these two models. As can be seen in the results, the proposed framework was able to achieve competitive results, obtaining correctness values similar to other traditional techniques and even better results in some datasets. Although the performance gain is not transversal to the entire set of problems used for validation, it was shown that using regularization on high-level structural properties of the models were able to transfer knowledge even between highly dissimilar learning paradigms.

This idea can also be explored in other predictive tasks (e.g. classification, regression) and between other models. For example, we can transfer the thresholds decided by a decision tree as the weak estimators used in AdaBoost, the features chosen by a sparse generalized linear model to the probabilities of including each feature in a Random Forest, etc.

3.4 Recommender Systems

Collaborative filtering is a frequent paradigm in Recommender Systems based on the idea of using preferences from many users to guide predictions about a given user's preferences, conversely, for items. Given N users and M items, Matrix Factorization is a type of collaborative filtering technique that approximates the preference matrix $R \in \mathbb{R}^{N \times M}$ by combining two matrices $U \in \mathbb{R}^{N \times D}$, $V \in \mathbb{R}^{M \times D}$, where D is a small number of unobserved factors that model user and items preferences, U and V respectively [36]. As typical, we consider the combination $R = U \cdot V^\top$. Salakhutdinov and Mnih [36] proposed Probabilistic Matrix Factorization, a method for fitting these latent factors by means of minimizing the regularized sum-of-squared-errors (see Eq. (21)), where $\| \cdot \|^2_{Fro}$ denotes the Frobenius norm and I_{ij} equals 1 if user i rated item j and equals 0 otherwise.

$$J(U,V) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} (R_{ij} - U_i V_j^\top)^2 + \frac{\lambda_U}{2} \sum_{i=1}^{N} \|U_i\|^2_{Fro} + \frac{\lambda_V}{2} \sum_{j=1}^{M} \|V_i\|^2_{Fro} \quad (21)$$

A local minimum of J can be found using gradient descent. A well known problem in Recommender Systems is the cold-start problem [33], which can be understood as the impossibility of producing accurate predictions for users (or items) with scarce information. This problem has been tackled in the past by introducing content information, using some priors when initializing the latent features of an entity, among others. In general, this problem can be understood as transferring knowledge from existing users to new users. In this work, we instantiate the proposed TL framework for solving the cold-start problem. Given k new users, the fitted unobserved factors for a given user U_i are regularized in order to be similar to its most similar previously fitted user U_i^* (see Eq. (22)).

$$J'(U) = \frac{1}{2} \sum_{i=N+1}^{N+k} \sum_{j=1}^{M} I_{ij} (R_{ij} - U_i V_j^\top)^2 + \frac{\lambda_U}{2} \sum_{i=N+1}^{N+k} \|U_i - U_i^*\|^2_{Fro} \quad (22)$$

Table 5: Comparison of Recommender Systems using different transfer strategies: Structural with a unique central user (Global) and Structural with a subset of candidate users (Subset). Performance is measured using Mean Absolute Error (MAE).

Dataset	**W**	**Ext**	**Proposed**	
			Global	**Subset**
Movielens100k [23]	9.08	4.52	12.06	**12.59**
Amazon Instant Video [35]	5.67	-0.11	6.08	**6.64**
Amazon Musical Instruments [35]	2.45	**6.30**	3.88	5.12
Amazon Videogames [35]	9.27	0.22	10.12	**11.05**
Jester2+ [22]	2.92	-1.72	12.46	**19.40**

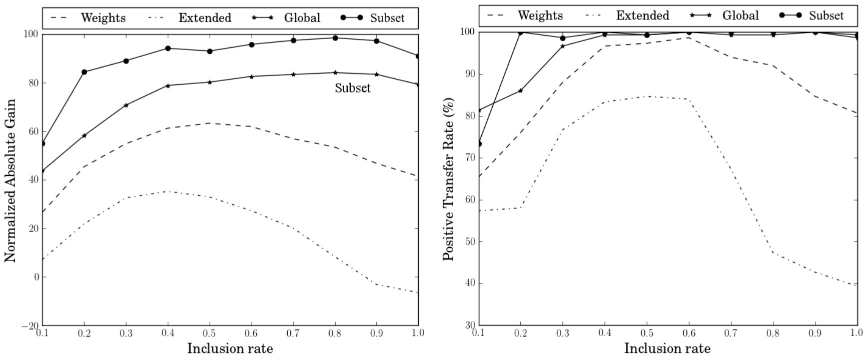

Fig. 8: Average gains (left) and positive transfer rates (right) with nested training sets on Recommender Systems

In order to simplify the computation of the most similar user, two variations of the proposed idea were considered: a subset of candidate users obtained using K-means (K=10) and a unique central user with the averaged latent features of the previously trained users. Applying the same ideas explored in Linear Regression (cf. Section (3.1)), the problem can be formulated in terms of the target residuals (Eqs. (23)).

$$J''(U) = \frac{1}{2} \sum_{i=N+1}^{N+k} \sum_{j=1}^{M} I_{ij} (\hat{R}_{ij} - \mathit{\Delta}_i V_j^\top)^2 + \frac{\lambda_U}{2} \sum_{i=N+1}^{N+k} \|\mathit{\Delta}_i\|_{Fro}^2 \tag{23}$$

where $\hat{R}_{ij} = R_{ij} - U_i^* V_j^\top$

In the experimental evaluation, the Extended baseline was modeled by interpolating the average ratings for the specified item and the predicted ratings. All experiments were executed using $D = 50$ latent factors and $\lambda_U = \lambda_V \in [10^{-3}, \ldots, 10^3]$. The users considered for transfer were the top 100 users with more votes in order to validate the widest spectrum of known ratings. As can be seen in Table 5, the proposed transfer schemes obtained the best results

Table 6: Overview of the performance of the proposed strategies. The table summarizes the number of datasets (%) where each proposed strategy achieved an average behavior better than the literature baselines. The cases where the proposed techniques performed better than the baselines are presented in bold.

Task	**Model**	**Type**	**W**	**Ext**
Regression	$\mathbf{L_2}$	Full	**71**	**100**
	$\mathbf{L_1}$	Full, Sparse	29	29
	EN	Full	29	43
Classification	**SVM**	Full	**75**	**88**
	S-SVM	Partial	**62**	**100**
	α**S-SVM**	Partial	**75**	**100**
	AdaBoost-Full	Full	38	**50**
	AdaBoost-Thres	Partial	25	**62**
	AdaBoost-Order	Partial	38	**50**
Ranking	**LexRank-LexRank-Prior**	Partial	**67**	**83**
	LexRank-LexRank-Pref	Partial	**50**	**67**
	LexRank-LexRank-Comb	Full	**50**	**67**
	RankSVM-LexRank-Prior	Cross-model, Partial	**50**	**83**
	RankSVM-LexRank-Pref	Cross-model, Partial	33	**50**
	RankSVM-LexRank-Comb	Cross-model, Full	**67**	**67**
RecSys	**Global**	Partial	**100**	**80**
	Subset	Partial	**100**	**80**

in most cases. An interesting property on the results that wasn't observed in previous cases is that gains achieved by our model increases through most of the spectrum of inclusion rates while the gains achieved by the other strategies saturate and decrease drastically after a given point (60% of inclusion rate). Moreover, the rate of cases with positive transfer using the proposed strategy is close to 100% (see Fig. 8).

3.5 Discussion

The proposed generic strategies achieved good performance when compared to traditional transfer strategies (see Table 6). For example, in more than 76% of the cases, the HTL techniques achieved better performance than their literature counterparts in at least half of the datasets. In general, at least one of the HTL-based strategies performed better than the alternative approaches from the literature. Moreover, the proposed methodologies tend to dominate the other approaches when data is scarce which is one of them main goals of transfer learning (see Figures 2, 4, 5, 7 and 8).

The optimal performance of the gain curves should be a monotonically decreasing curve, where the gains achieved by using transfer learning are high when data is scarce and tend to zero as we add data to the training set. However, given that we are measuring the performance of the model on a small subset of partitions (30 runs), it is expectable to observe an irregular behavior.

We focused on providing a general framework that may be instantiated to achieve good performance in a wide diversity of scenarios. As in traditional machine learning settings where the best model is unknown a priori, finding the best regularizer, its observability and the regularization strength (λ) are application-dependent problems which can be solved – in general – using cross-validation. Moreover, application knowledge can be used to conduct this selection.

4 Conclusions

In this work we presented a new transfer learning framework based on structural model regularization. In contrast to most transfer learning techniques, which either transfer data or are designed for specific models, the proposed framework addresses the problem of transferring knowledge in a general way. Namely, knowledge is transferred by including a regularization term that measures the structural similarity between source and target models. Thereby, the proposed method is able to reuse knowledge gained from the source task without revisiting source data, which might be prohibitively large or even unavailable at transfer time. In order to show its flexibility, the proposed framework was instantiated to several learning tasks: regression, classification, learning to rank and recommender systems. Positive results were obtained in most experiments, being competitive with other methods in the literature both, in terms of predictive performance and in terms of computational cost. Furthermore, key problems like sparse, partial and cross-model transfer were analyzed and assessed, showing their adequacy on several scenarios. The proposed method relies on defining a good relatedness measure between models, which may allow the integration of application-specific knowledge.

As future work, it is relevant to evaluate the performance of the proposed methodology with multiple source tasks and with multiple similarity functions, enabling the user to specify several alternatives to embed the desired knowledge in the learning process. While this could be done in a straightforward manner using weighted regularization terms, the empirical study of this problem is relevant.

Transfer learning research line should move towards a deep understanding on how models encode knowledge and how to transfer this knowledge in a general and unified way. Through this paper, we explored how this can be done efficiently. Emerging regularization schemes that favor this kind of transfer are feasible paths to explore, as well as similarity learning techniques able to infer the actual relatedness between models for a specific task.

Acknowledgements This work was funded by the Project "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics / NORTE-01-0145-FEDER-000016" financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF), and also by Fundação para a Ciência e a Tecnologia (FCT) within PhD grant number SFRH/BD/93012/2013.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning **73**(3), 243–272 (2008)
2. Argyriou, A., Pontil, M., Ying, Y., Micchelli, C.A.: A spectral regularization framework for multi-task structure learning. In: Advances in neural information processing systems, pp. 25–32 (2007)
3. Aytar, Y., Zisserman, A.: Tabula rasa: Model transfer for object category detection. In: 2011 International Conference on Computer Vision, pp. 2252–2259. IEEE (2011)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
5. Ben-David, S., Urner, R.: Domain adaptation as learning with auxiliary information. In: New Directions in Transfer and Multi-Task-Workshop@ NIPS (2013)
6. Bonilla, E.V., Chai, K.M., Williams, C.: Multi-task gaussian process prediction. In: Advances in neural information processing systems, pp. 153–160 (2007)
7. Booth, R., Chevaleyre, Y., Lang, J., Mengin, J., Sombattheera, C.: Learning conditionally lexicographic preference relations. In: ECAI, pp. 269–274 (2010)
8. Cheng, W., Rademaker, M., De Baets, B., Hüllermeier, E.: Predicting partial orders: ranking with abstention. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 215–230. Springer (2010)
9. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems **47**(4), 547–553 (2009)
10. Cortez, P., Silva, A.M.G.: Using data mining to predict secondary school student performance (2008)
11. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200. ACM (2007)
12. Daume III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research **26**, 101–126 (2006)
13. Davis, J., Domingos, P.: Deep transfer via second-order markov logic. In: Proceedings of the 26th annual International Conference on Machine Learning, pp. 217–224. ACM (2009)
14. Dredze, M., Kulesza, A., Crammer, K.: Multi-domain learning by confidence-weighted parameter combination. Machine Learning **79**(1-2), 123–149 (2010)
15. Evgeniou, A., Pontil, M.: Multi-task feature learning. Advances in Neural Information Processing Systems **19**, 41 (2007)
16. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM (2004)
17. Fernandes, K., Cardoso, J.S., Palacios, H.: Learning and ensembling lexicographic preference trees with multiple kernels. In: Proceedings of International Joint Conference on Neural Networks (IJCNN) (2016)
18. Fernandes, K., Cardoso, J.S., Fernandes, J.: Transfer Learning with Partial Observability Applied to Cervical Cancer Screening. Iberian Conference on Pattern Recognition and Image Analysis, 243–250 (2017)
19. Flach, P., Matsubara, E.T.: A simple lexicographic ranker and probability estimator. In: European Conference on Machine Learning (ECML), pp. 575–582. Springer (2007)
20. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory, pp. 23–37. Springer (1995)
21. Garcke, J., Vanck, T.: Importance weighted inductive transfer learning for regression. In: Machine Learning and Knowledge Discovery in Databases, pp. 466–481. Springer (2014)

22. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval **4**(2), 133–151 (2001)
23. Harper, F.M., Konstan, J.A.: The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) **5**(4), 19 (2015)
24. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470), vol. 1, pp. 97–102. IET (1999)
25. Jiang, J.: Multi-task transfer learning for weakly-supervised relation extraction. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pp. 1012–1020. Association for Computational Linguistics (2009)
26. Jiang, L., Zhang, J., Allen, G.: Transferred correlation learning: An incremental scheme for neural network ensembles. In: Neural Networks (IJCNN), The 2010 International Joint Conference on, pp. 1–8. IEEE (2010)
27. Kandaswamy, C., Silva, L.M., Cardoso, J.S.: Source-target-source classification using stacked denoising autoencoders. In: Pattern Recognition and Image Analysis, pp. 39–47. Springer (2015)
28. Kuzborskij, I., Orabona, F.: Stability and hypothesis transfer learning. In: ICML (3), pp. 942–950 (2013)
29. Kuzborskij, I., Orabona, F.: Fast rates by transferring from auxiliary hypotheses. Machine Learning **106**(2), 171–195 (2017)
30. Lee, C., Jang, M.G.: A prior model of structural SVMs for domain adaptation. ETRI Journal **33**(5), 712–719 (2011)
31. Li, X., Mao, W., Jiang, W.: Extreme learning machine based transfer learning for data classification. Neurocomputing **174**, 203–210 (2016)
32. Lichman, M.: UCI machine learning repository (2013). URL http://archive.ics.uci.edu/ml
33. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. Expert Systems with Applications **41**(4), 2065–2073 (2014)
34. Long, M., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. arXiv preprint arXiv:1605.06636 (2016)
35. McAuley, J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2015)
36. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: J.C. Platt, D. Koller, Y. Singer, S.T. Roweis (eds.) Advances in Neural Information Processing Systems 20, pp. 1257–1264. Curran Associates, Inc. (2008). URL http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf
37. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1717–1724 (2014)
38. Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on **22**(10), 1345–1359 (2010)
39. Perrot, M., Habrard, A.: A theoretical analysis of metric hypothesis transfer learning. In: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp. 1708–1717 (2015)
40. Povey, D., Chu, S.M., Varadarajan, B.: Universal background model based speech recognition. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 4561–4564. IEEE (2008)
41. Rückert, U., Kramer, S.: Kernel-based inductive transfer. In: Machine Learning and Knowledge Discovery in Databases, pp. 220–233. Springer (2008)
42. Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: Primal estimated subgradient solver for SVM. Mathematical programming **127**(1), 3–30 (2011)
43. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging **35**(5), 1285–1298 (2016)
44. Silver, D.L., Poirier, R., Currie, D.: Inductive transfer with context-sensitive neural networks. Machine Learning **73**(3), 313–336 (2008)

45. Tommasi, T., Orabona, F., Caputo, B.: Learning categories from few examples with multi model knowledge transfer. IEEE transactions on pattern analysis and machine intelligence **36**(5), 928–941 (2014)
46. Tsanas, A., Little, M.A., McSharry, P.E., Ramig, L.O.: Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. Biomedical Engineering, IEEE Transactions on **57**(4), 884–893 (2010)
47. Wang, Y., Wang, L., Li, Y., He, D., Chen, W., Liu, T.Y.: A theoretical analysis of NDCG ranking measures. In: Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013). Citeseer (2013)
48. Yang, L., Hanneke, S., Carbonell, J.: A theory of transfer learning with applications to active learning. Machine learning **90**(2), 161–189 (2013)
49. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
50. Zhang, J., Ghahramani, Z., Yang, Y.: Flexible latent variable models for multi-task learning. Machine Learning **73**(3), 221–242 (2008)