Interactive, Adaptive Transfer Learning for Biomedical Imaging

Róger Bermúdez-Chacón
CVLab – I&C, EPFL

***Abstract*—Although Machine Learning (ML) has proved useful for a wide range of disciplines, the assumption of same distribution that most ML methods impose on the data is, for many applications, an important limitation that needs to be overcome. This is especially relevant when the acquisition of annotated data is difficult or costly, but annotated data for similar or related problems are readily available.**

Transfer Learning (TL) provides the ideas and methods to make use of heterogeneous data in such a way that predictions for problems with few or no annotated data, are improved by borrowing knowledge from preexisting data.

Interactive approaches to Transfer Learning take this idea one step further, by adding an extra source of knowledge —the expert user— into the mix.

In the following proposal, we compile and present the main motivations and formulations that have been explored in the field of TL, describe two specific methods that exploit different approaches for regression and classification, and suggest new lines of research on the topic, and its applications in biomedical computer vision.

***Index Terms*—Transfer Learning, Biomedical Imaging, Computer Vision**

I. Introduction

MACHINE LEARNING (ML) has become a standard tool to deal with large amounts of data in a vast number of areas, ranging from economics and natural language processing, to biomedical research. ML algorithms provide rules to predict some response for new data based on preexisting knowledge. One big limitation of such algorithms, however, is that they make the assumption that all the data on which they operate is drawn from the same distribution.

The above assumption is often violated. When experiments of the same nature, but under different experimental conditions, are carried out, the distribution of the different sets of data obtained is most likely to differ (Fig. 1). Furthermore, one can be interested in reusing knowledge from related but different tasks, to help solve a new predictive task where the data is completely different in nature.

Transfer Learning (TL) is the idea that, upon proper manipulation of the data, knowledge acquired for a related, albeit different predictive task, can be reused —or *transferred*— to a new one. In general, TL methods make use of knowledge obtained from one or more reference, or *source* datasets, to improve the prediction on a given *target* dataset. It is also assumed that source and target tasks are related in some way.

Transfer Learning has been successfully applied to problems as diverse as Web-document classification [1], WiFi localization [2], sentiment classification for product reviews [3], and synapse and mithochondria segmentation [4], to name a few.

Different TL approaches have been suggested, according to what kind of data are available, and what part of the predictive task can be transferred. Recently, interactive strategies that

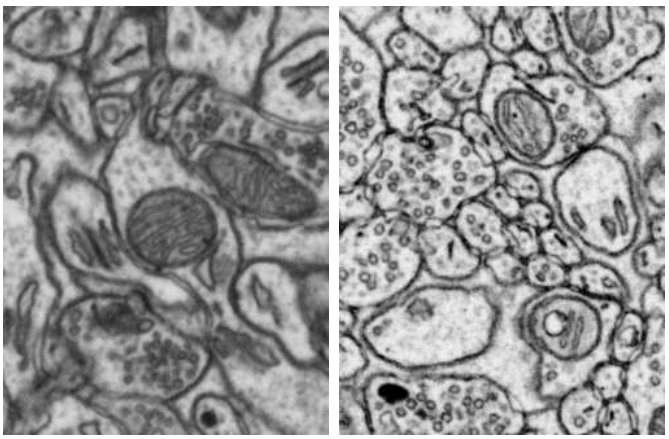


Fig. 1: Two electron microscopy images of rat brain tissue, acquired under different experimental conditions. Qualitative differences between both images make it difficult to use knowledge extracted from one source for prediction onto the other directly.

Proposal submitted to committee: June 15th, 2015; Candidacy exam date: June 22nd, 2015; Candidacy exam committee: Prof. Sabine Süsstrunk, Prof. Pascal Fua, Dr. Graham Knott.

This research plan has been approved:

Date: _____

Doctoral candidate: _____
(name and signature)

Thesis director: _____
(name and signature)

Doct. prog. director: _____
(B. Falsafi) (signature)

incorporate user feedback to guide the transfer process have been proposed [5].

The rest of this document is organized as follows: first we introduce the formal framework and notation, and provide an overview of the different ideas and techniques that have been proposed in the literature for TL. Then we provide two examples of methods that have been successfully applied to transfer knowledge across domains, one for regression and the other for classification. The former makes use of importance sampling to reweight the source domain, whereas the latter combines transfer and active learning. We conclude by proposing some research directions that we will further investigate.

II. Transfer Learning: A definition [6]

We introduce the following definitions and notation, to be used throughout the rest of this document:

- A **domain** $\mathcal{D}$ is a pair $\{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ is a *feature space*, X is a set of observations $\{x_1, \dots, x_n\}, x_i \in \mathcal{X}$, and $P(X)$ is the marginal probability distribution of X. In general, two domains are different if they differ either in their feature space or in their marginal probability distribution.
- Given a domain, a **task** $\mathcal{T}$ is a pair $\{\mathcal{Y}, f(x)\}$, where $\mathcal{Y}$ is a label space, and $f(x)$ is a function that predicts the label associated to instances $x \in X$. Probabilistically, $f(x)$ can be regarded as $P(y|x)$. In general, $f(x)$ is not observed, but can be learned from examples $\{x_i, y_i\}$, $x_i \in X$, and $y_i \in \mathcal{Y}$ (also known as the *training data*).
- From a source domain $\mathcal{D}_S$ and related task $\mathcal{T}_S$, the **source domain data** is the set of pairs $D_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_{n_S}}, y_{S_{n_S}})\}$, with $x_{S_i} \in \mathcal{X}_S$ and $y_{S_i} \in \mathcal{Y}_S$. Likewise, the **target domain data** for a target domain $\mathcal{D}_T$ and related task $\mathcal{T}_T$ is denoted as $D_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_{n_T}}, y_{T_{n_T}})\}$, with $x_{T_i} \in \mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$.

Definition 1 (**Transfer Learning).** Given a source domain $\mathcal{D}_S$ with related task $\mathcal{T}_S$, and a target domain $\mathcal{D}_T$ with related task $\mathcal{T}_T$, *Transfer Learning* aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$, by using knowledge from the source domain $\mathcal{D}_S$ and related task $\mathcal{T}_T$, where, in general, $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ [6].

Under the above general definition, most TL methods aim to solve the specifics on *what*, *how* and *when* to transfer knowledge between domains. "What to transfer" refers to what is shared or common between domains —and hence useful for transfer— and what is specific to each one. "How to transfer" involves the actual algorithms that encode and pass knowledge between domains. "When to transfer" refers to the phenomenon of *negative transfer*, in which using knowledge from other tasks into a new one causes more harm than good.

The ultimate goal of TL is to boost performance on target domains where labeled data is scarce or nonexistent. According to the availability of *labeled* examples in the target domain, it is possible to distinguish between the following three scenarios:

1) Some labeled examples are available in the target domain. In this case, such labeled data can be used in conjunction with the source domain to *induce* a predictive function. This scenario is known as **Inductive Transfer Learning**.
2) No labeled data are available in the target domain. The TL methods can only make use of the labeled data from the source domain, and the unlabeled data from the target domain. This is known as **Transductive Transfer Learning**.
3) **Unsupervised Transfer Learning** comprises tasks such as clustering, dimensionality reduction, or density estimation, where no labeled data are available in either domain.

III. What, how and when to transfer

We briefly describe the different aspects of the source and target data —the "what to transfer"— that most TL methods exploit, and provide, for each scenario, an example of the formulations —the "how to transfer"— proposed to relate the knowledge between the source and target domains. We discuss the shortcomings of using TL indiscriminately on arbitrary source and target domains —the "when to transfer"—.

A. Instance transfer

Instance-based transfer learning corresponds to the idea that instances or examples from the source domain can be incorporated into the target domain for prediction, after applying some transformation on their features. The idea is that, upon transformation, the distribution of the instances from the source domain matches the target domain distribution, which makes it possible to train models by using both the labeled target domain and the transformed version of the source domain.

Many instance-based transfer methods achieve this by using *importance sampling*. Importance sampling works by defining a reweighting function $w(\cdot, \cdot)$ to apply on the source domain, as:

$$P_T(x, y) = \frac{P_T(x, y)}{P_S(x, y)} P_S(x, y) = w(x, y) P_S(x, y), \quad (1)$$

where $P_S(x, y)$ and $P_T(x, y)$ are the joint probabilities of the instances and their labels, for the source and target domain respectively. Different methods to estimate $w(x, y)$ have been proposed ([7], [8], [9], [10]). In Section IV we study one such approach in more detail.

B. Feature transfer

Standard feature transfer approaches find a new representation of the source and target domains that is shared across their tasks. One way to learn such representation [11] can be expressed as an optimization problem of the form:

$$\arg\min_{A,U} \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} L(y_{t_i}, \langle a_t, U^\mathsf{T} x_{t_i} \rangle) + \lambda ||A||^2, \quad (2)$$

where L is a loss function, λ is a regularization coefficient, S and T denote the source and target domains, $A = [a_S \ a_T] \in \mathbb{R}^{d\times 2}$ is a matrix of parameters, and $U \in \mathbb{R}^{d\times d}$ is an orthogonal matrix that maps the domains into the shared lower-dimensional representation, as $U^\intercal X_S$ and $U^\intercal X_T$.

Other approaches, such as Structural Correspondence Learning (SCL [12]), augment the target domain with new features that encode the correspondence between domains. SCL makes use of (domain specific) binary *pivot features* taken from both source and target domains. The pivot features are removed from the data and used, one by one, as labels for classification problems:

$$f_l(x) = \text{sgn}(w_l^\intercal x),\ l = 1, \ldots, m, \tag{3}$$

applied on the *unlabeled* data of both source and target domains.

The w_l parameters learned from the m pivot features encode the relationship between the two domains. The matrix of parameters $W = [w_1 \ w_2 \ \ldots \ w_m]$ is factorized into $UDV^\intercal$ by singular value decomposition. $\theta = U_{[1:h,:]}$, i.e., the first h rows of U (the main singular vectors), are selected. A model is trained on the source domain, augmented by the new features: $X_{S_{scl}} = [X_S \ \theta X_S]$. Prediction is done on the augmented target domain $X_{T_{scl}} = [X_T \ \theta X_T]$.

C. Parameter transfer

Parameter transfer assumes that there is shared knowledge encoded into the model parameters or in the prior distribution of the hyperparameters. One example of such approaches, an adaptation of SVM [13], creates models to predict on both the source and target tasks. The method assumes source and target versions of the parameter w that define the support vectors, as:

$$w_S = w_0 + v_S, \tag{4}$$

$$w_T = w_0 + v_T. \tag{5}$$

In this setup, the parameter models w_S and w_T (for source and target tasks, respectively) are decomposed into a shared parameter w_0, and parameters v_S and v_T specific to each task.

The method learns the shared and specific parts of the parameters, by optimizing an objective function of the form:

$$\min_{w_0, v_t, \xi_t} \sum_{t\in\{S,T\}} \sum_{i=1}^{n_t} \xi_{t_i} + \frac{\lambda_1}{2} \sum_{t\in\{S,T\}} ||v_t||^2 + \lambda_2 ||w_0||^2, \tag{6}$$

such that $y_{t_i}(w_0 + v_t) \cdot x_{t_i} \geq 1 - \xi_t, \xi_t \geq 0$, analogous to the original SVM formulation.

D. Relationship transfer

When relationships (dependencies) between the data exist, statistical relational learning techniques can be used. One such approach is the use of Markov Logic Networks, which rely on first-order logic to perform uncertain inference.

One example of such methods is TAMAR [14], whose main assumption is that entities and their relationships from one domain (e.g. $Professors \leftrightarrow Students$) can be mapped or connected to entities and relationships from a different, related domain (e.g. $Managers \leftrightarrow Workers$).

E. Negative transfer

One of the main challenges in TL research is how to avoid *negative transfer*, whereby transferring knowledge actually hinders the performance of the predictive task. Such situation happens when the assumption that the source and target domains are related does not hold.

Even when the domains are semantically related, suboptimal transfer can happen when the source domain is poorly chosen, or by inappropriate data preprocessing on either domain.

In order to avoid negative transfer, suitable transferability measurement criteria must be designed. Limited effort has been made on this topic.

IV. An application of Transfer Learning to Regression [10]

Under the inductive transfer learning setting, Garcke and Vanck [10] have recently proposed two instance transfer methods based on importance sampling. They explore the general case where the source and target domains differ in the joint probability of the features and the labels, i.e. $P_S(x, y) \neq P_T(x, y)$.

Importance sampling relies on the assumption that, for some parts of the source and target domains, their joint distributions are similar, i.e. $P_S(\tilde{x}, \tilde{y}) \approx P_T(\tilde{x}, \tilde{y})$ for some $(\tilde{x}, \tilde{y})$. The aim is to find an importance function $w(x, y)$ to reweight the source data points, such that instances that contribute to the prediction have high weight and instances with negative influence are mostly neglected. As seen in equation (1), the expression for the appropriate weight requires knowledge of both P_S and P_T, which in practice is not available.

Two methods (one supervised and the other unsupervised) are proposed to adequately reweight instances from a source domain and incorporate them into the target task.

A. Supervised (direct) method

In general, the optimization problem on the target domain is given by:

$$\min ||Y_T - \hat{Y}_T||^2, \tag{7}$$

where Y_T are the true labels and $\hat{Y}_T$ are the predictions.

Predictions on the target domain, under the importance sampling formulation, are given by:

$$\begin{aligned} \hat{y}_T &= \arg\max_y P_T(y|x_T) P_T(x_T) \\ &= \arg\max_y \frac{P_T(y|x_T) P_T(x_T)}{P_S(x_T, y)} P_S(y|x_T) P_S(x_T) \\ &= \arg\max_y w(x_T, y) P_S(y|x_T) P_S(x_T). \end{aligned} \tag{8}$$

Therefore, we can incorporate the source domain data into the target task, as:

$$\min_{\hat{w}} \sum_{i=1}^{n_T} \left(y_{T_i} - \arg\max_y \hat{w}(x_{T_i}, y) P_S(y|x_{T_i}) \right)^2. \quad (9)$$

The proposed method uses weighted Kernelized Ridge Regression KRR to model the predictive task. In general, predictions for new instances under KRR are given by:

$$\hat{y}^* = \arg\max_y p(y|x^*) = a^\mathsf{T} \mathbf{k}(x^*), \quad (10)$$

for some vector a, where $\mathbf{k}(\cdot)$ is a kernel function applied on the new data x^* and the training data.

Likewise, a weighted prediction model derived from KRR can be expressed as:

$$\arg\max_y (w(x^*, y) p(y|x^*)) = a^\mathsf{T} W(x^*, y) \mathbf{k}(x^*) \quad (11)$$

where W is a diagonal matrix of weights.

Under this result, we can express the target prediction task as:

$$\min_{\hat{w}} \sum_{i=1}^{n_T} \left(y_{T_i} - a^\mathsf{T} W(x_{T_i}, y_{T_i}) \mathbf{k}(x_{T_i}) \right)^2. \quad (12)$$

Importance function: In order to estimate the importance function $w(x, y)$, the common approach of linear combination of Gaussian kernels is used:

$$\hat{w}_\alpha(x, y) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{||(x, y) - (x_{S_i}, y_{S_i})||^2}{2\eta^2} \right). \quad (13)$$

The problem then reduces to find appropppriate α coefficients for each instance in the source domain.

Solving equation (12) for W requires knowledge of the parameter a. For this, since the labels in the source domain are known, one can estimate a directly by solving (10) for a.

Once the value for a is available, the α coefficients for (13) can be calculated from W. The importance function is then fully defined.

Loss function: For prediction on the target domain, the following KRR loss function, that considers the data from both domains, is proposed:

$$\min_\theta \frac{1}{2} \left(S_{\theta,\mathbf{1}}(x_T, y_T) + S_{\theta,\mathbf{w}}(x_S, y_S) \right) + \frac{\lambda}{2} ||\theta||^2,$$
$$\text{with } S_{\theta,\mathbf{w}}(x_t, y_t) = \sum_{i=1}^{n_t} w_i \left(\theta^\mathsf{T} \phi(x_{t_i}) - y_{t_i} \right)^2. \quad (14)$$

Notice that the instances in the source domain are weighted by $\mathbf{w} = \{w(x_{S_1}, y_{S_1}), \ldots, w(x_{S_{n_S}}, y_{S_{n_S}})\}$.

The hyperparameters λ and η are determined by cross-validation.

B. Unsupervised (indirect) method

Another method to find an appropppriate w relies on the difference between target and reweighted source distributions, rather than on the predictions. The loss function to optimize is given in terms of their Kullback-Leibler divergence, as:

$$\begin{aligned} &\min_\alpha \mathrm{KL}\left(P_T(x, y) \;||\; \hat{w}_\alpha(x, y) P_S(x, y) \right) \\ &= \min_\alpha \left(\int P_T(x, y) \log \frac{P_T(x, y)}{\hat{w}_\alpha(x, y) P_S(x, y)} \; dxdy \right) \\ &= \min_\alpha \left(- \int P_T(x, y) \log \hat{w}_\alpha(x, y) \; dxdy \right) \\ &\approx \max_\alpha \sum_{i=1}^{n_T} \log \hat{w}_\alpha(x_{T_i}, y_{T_i}), \end{aligned} \quad (15)$$

under the normalization constraints:

$$\begin{aligned} \int \hat{w}_\alpha(x, y) P_S(x, y) \; dxdy = 1 \\ \Rightarrow \sum_{i=1}^{n_S} \hat{w}_\alpha(x_{S_i}, y_{S_i}) = n_S, \\ \alpha \geq 0. \end{aligned} \quad (16)$$

Optimization of (15) under the constraints given by (16) is performed by a standard solver for constrained problems. Once the α coefficients for the importance function have been found, the same loss function defined for the direct method can be used for prediction on the target domain.

Both the direct and the indirect methods have been successfully applied to problems such as earthquake prediction, where the source and target domains come from measurements taken from different geographical locations, and indoor location estimation, where the source and target data have been also acquired from measurements from different locations.

The main drawbacks of these methods are that they assume that corresponding regions from the source and target distributions are similar, they necessarily require labeled data from the target domain, and they do not make use of the unlabeled data from the target domain.

In the following section, we review a recent approach that considers the unlabeled target data, as well as expert user input in an interactive fashion.

V. Combining Transfer Learning and Active Learning [5]

Transfer Learning addresses the issue of insufficient labeled data in a target domain, by making use of auxiliary data from other related domains. The related field of Active Learning (AL) addresses the same problem by finding unlabeled instances in the target domain that, if labeled, would increase the predictive performance of a task the most. Efforts to combine both methodologies have been proposed in [15] and [16].

Chattopadhyay et. al. [5] introduce a framework that, unlike previous approaches, integrates both TL and AL into a single optimization problem.

A. Problem formulation

The central idea of the method is to split the instances from both domains into two groups. The first group will contain the labeled data from both the source and the target domain, as well as a small batch of unlabeled data from the target domain. The second group will contain the rest of the unlabeled instances from the target domain.

Since the source and target domain distributions differ, a reweight of the source domain instances is needed, analogous to the application of the importance function in Section IV.

The method aims to match the marginal probabilities of both groups described above. This is achieved by simultaneously finding the proper weights for the source domain data, and the subset of the unlabeled data that, after manual labeling, improves the matching the most.

We regard the transformed source domain as S_a, the labeled target domain data as L, the unlabeled target domain data as U, and the batch selected for labeling as Q. The aim of the method can be thus expressed as finding S_a and Q, such that $P_{S_a \cup L \cup Q}(X) \approx P_{U \setminus Q}(X)$.

The difference between the marginal distributions of the two sets is quantified by using Maximum Mean Discrepancy (MMD) [17], defined on two sets of instances A and B, with n_A and n_B instances each, as:

$$mmd(A,B) := \left\| \frac{1}{n_A} \sum_{i=1}^{n_A} \phi(x_{A_i}) - \frac{1}{n_B} \sum_{j=1}^{n_B} \phi(x_{B_j}) \right\|^2. \quad (17)$$

We denote the weight for each instance x_{S_i} from the source domain data as w_i. The selection of unlabeled instances for the batch is performed by using a binary vector α, where a value of 1 indicates that the corresponding unlabeled instance is selected. The MMD-like cost function to optimize then becomes:

$$\begin{aligned} \arg\min_{w,\alpha} \Bigg\| \frac{1}{n_s + n_l + b} \Bigg(\sum_{i \in S} w_i \phi(x_{S_i}) + \sum_{j \in L} \phi(x_{T_j}) \\ + \sum_{k \in U} \alpha_i \phi(x_{T_k}) \Bigg) - \frac{1}{n_u - b} \sum_{i \in U} (1 - \alpha_i) \phi(x_{T_i}) \Bigg\|^2, \end{aligned} \quad (18)$$

with n_s, n_l, and n_u the number of instances in the source domain, the labeled target domain, and the unlabeled target domain, respectively, and b the batch size.

Solving this optimization problem is equivalent to solving its dual version:

$$\begin{aligned} \min \frac{1}{2} \alpha^\mathsf{T} K_{u,u} \alpha + \frac{1}{2} w^\mathsf{T} K_{s,s} w + w^\mathsf{T} K_{s,u} \alpha \\ - k_{u,u}^\mathsf{T} \alpha - k_{s,u}^\mathsf{T} w + k_{u,l}^\mathsf{T} \alpha + k_{s,l}^\mathsf{T} w + \text{const.} \\ \text{s.t. } \alpha_i \in \{0,1\}, w_i \in [0,1], \alpha^\mathsf{T} \mathbf{1} = b, \end{aligned} \quad (19)$$

The following notation is introduced to describe the terms above. We define $G_{(n_s+n_u+n_l)\times(n_s+n_u+n_l)}$, the kernel matrix over S, U, and L, and $c = \frac{n_s+n_u+n_l}{n_u-b}$. Then:

$$\begin{aligned} K_{s,s} &= \frac{1}{c^2} G(1:n_s,\ 1:n_s), \\ K_{u,u} &= G(n_s+1:n_s+1+n_u,\ n_s+1:n_s+1+n_u), \\ K_{s,u} &= \frac{1}{c} G(1:n_s,\ n_s+1:n_s+1+n_u), \\ k_{u,u}(i) &= \frac{n_s+n_l+b}{c^2(n_u-b)} \sum_{j=1}^{n_u} K_{u,u}(i,j), \\ k_{s,u}(i) &= \frac{n_s+n_l+b}{c^2(n_u-b)} \sum_{j=1}^{n_u} K_{s,u}(i,j), \\ k_{s,l}(i) &= \frac{1}{c^2} \sum_{j=1}^{n_l} G(i,\ n_s+n_u+j), \\ k_{u,l}(i) &= \frac{1}{c} \sum_{j=1}^{n_l} G(n_s+i,\ n_s+n_u+j). \end{aligned}$$

Each term in (19) contributes to meet desireable properties such as representativeness, diversity and minimum redundancy in the selected data:

- The first term selects unlabeled instances with minimum similarity, hence avoiding *redundancy* in the selected batch.
- The second term minimizes the similarity within the reweighted source instances, hence avoiding *redundancy* in the source set.
- The third term minimizes the similarity between the selected query and the reweighted source data, hence avoiding *information overlap*.
- The fourth term maximizes the similarity between the unlabeled target data and the selected batch, hence enforcing *representativeness* of the batch.
- The fifth term maximizes the similarity between the reweighted source data and the unlabeled data, hence enforcing *representativeness* of the reweighted source data.
- The sixth term minimizes the similarity between the labeled data and the batch, hence enforcing *diversity* in the batch.
- The last term minimizes the similarity between the labeled target data and the reweighted source data, hence enforcing *diversity* in the reweighted set.

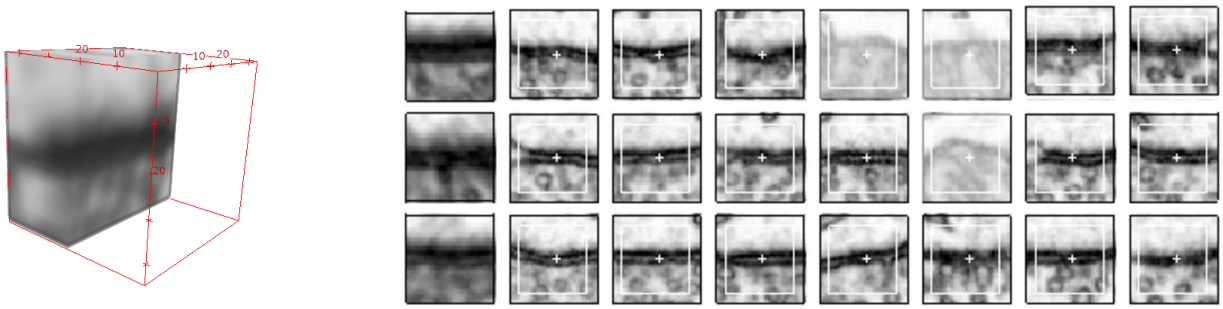
B. Quadratic Programming (QP) formulation

The binary constraint on the elements of α makes the problem in (19) NP-hard. A relaxation of this constraint is proposed. The resulting QP formulation is given by:

$$\min_{X: X_i \in [0,1], X^\mathsf{T} B = b} 0.5 X^\mathsf{T} H X + f^\mathsf{T} X,$$

where

$$H = \begin{bmatrix} K_{s,s} & K_{s,u} \\ K_{s,u}^\mathsf{T} & K_{u,u} \end{bmatrix}, \qquad f = \begin{bmatrix} k_{s,l} - k_{s,u} \\ k_{u,l} - k_{u,u} \end{bmatrix},$$
$$X = \begin{bmatrix} w \\ \alpha \end{bmatrix}, \qquad B = \begin{bmatrix} \mathbf{0}_{n_s \times 1} \\ \mathbf{1}_{n_u \times 1} \end{bmatrix}. \quad (20)$$



(a) (b)

Fig. 2: Interdomain correspondences in image space, for two electron microscopy datasets. (a) Synapsis extracted from a reference source domain, centered about the synaptic cleft, and rotated to a canonical orientation. (b) Central slices of the interdomain correspondences found by using Normalized Cross Correlation as a metric of similarity on 3D patches; each row shows an instance from a reference source domain (leftmost column), and its correspondences on the target domain. False positives (shown shaded) are selected by a manual annotator.

Solving the QP can be done efficiently by using existing solvers. Solving for X will simultaneously give the weights w for the source domain instances, and a score $\alpha \in [0,1]$ for the unlabeled instances. The batch Q chosen for manual labeling consists of the b unlabeled instances with the highest values of α.

Each time a new batch Q has been labeled, the weights and sets of labeled and unlabeled data are updated. The steps for the joint optimization can be summarized as:

1) Compute H and f.
2) Solve (20) for w and α.
3) Choose the b unlabeled instances with highest α as Q.
4) Manually annotate Q. Update the sets of labeled and unlabeled instances, $L \leftarrow L \cup Q, U \leftarrow U \setminus Q$.
5) Update the source instance weights $w_{new} \leftarrow w_{new} + w$.
6) Repeat.

It is possible to use the above formulation for TL or AL only. The main difference is that, for TL, one would only optimize for $X = w$ while neglecting α. Likewise, for AL only, one finds the optimal batch Q by optimizing $X = \alpha$, and ignoring w and the source domain data.

As described above, this method is flexible and does not require target labeled data. However, it does not make use of label information, as it only tries to match the marginal distribution of the features. Another potential issue is that the metric to optimize compares distributions in terms of their means only, which might underrepresent the distributions.

This method has been successfully applied to sentiment analysis on product reviews from different categories, such as electronics and books. The interactive and adaptive nature of this method is an interesting topic of research that we want to further study.

VI. Discussion and Research Proposal

The same distribution relaxation that TL methods provide does not come without its drawbacks. As mentioned above, some methods require specific kinds of features, others are computationally expensive, and others make strong assumptions about the problem, which limits their range of applicability.

Methods based on importance sampling, for example, assume that there is a significant overlap between the source and target distributions that can be leveraged by reweighting. Systematic feature shifts, quite common in many problems, render this assumption invalid.

Negative transfer has been largely disregarded in the literature, mostly because it is difficult to detect beforehand. When sharing knowledge between domains, it seems natural to design a transfer strategy that is sensitive to the particular problem. While this might not be enough to prevent negative transfer, finding a semantic common ground between the two sets of instances is a necessary first step.

In the particular context of biomedical imaging, we have explored the incorporation of problem-specific knowledge into the TL formulation. The main idea is to employ a metric of similarity between source and target instances in image space to exploit context and image attributes to guide a domain adaptation task.

Given the metric of similarity, we find candidate correspondences from source domain instances to instances in the target domain (Fig. 2). The main intuition is that instances that are similar (under the chosen similarity metric) between the two datasets will induce correspondences also in feature space.

By using the correspondences between the source and target domain as anchors between the source and target distributions, a KRR model is used for learning a transformation that, when applied on the training data, will bring its distribution as close as possible to the source data distribution:

$$\min_{\theta} \frac{1}{2}\left|\left|X_S - \theta^{\mathsf{T}}\phi(X_T)\right|\right|^2 + \frac{\lambda}{2}||\theta||^2. \tag{21}$$

The transformation found by this method is then applied to all instances from the target domain. Models trained on the source domain can be readily applied on the transformed target instances for prediction.

We propose to explore the one-to-many relationship scenario, where each reference instance from the source domain can be related to multiple instances from the target domain. Since different correspondences have, in general, different similarity values, a straightforward way to extend our approach to multiple correspondences is to use a weighted version of

the optimization problem, of the form:

$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n_S} \sum_{i \in C(x_{S_j})} w_i \Big(x_{S_j} - \theta^{\mathsf{T}} \phi(x_{T_i}) \Big)^2 + \frac{\lambda}{2} ||\theta||^2, \quad (22)$$

where $C(x_{S_j})$ is the set of correspondences found in the target domain for a source domain instance x_{S_j}, and w_i quantifies the quality of the correspondence.

This formulation is given under the assumption that the different correspondences to a source instance in the target domain are close to each other in feature space. An assessment of this situation will reveal if different weighting strategies are necessary.

We also want to investigate the robustness of the similarity metric, by using the correspondences found on the target domain as a reference, and reverse the roles of source and target domain.

Here, we will find correspondences for target domain instances in the source domain. If such correspondences lie close to the original source data, we can in turn use those new correspondences (instances in the source domain) to obtain new correspondences in the target domain, in a recursive fashion. Our hypothesis is that this will help us detect regions on both feature spaces where we are highly confident of the quality of matches. One possible risk of this approach is that it might overrepresent some parts of the domains (sampling bias). Strategies to balance exploration of the feature space and exploitation of the most representative instance groups must be considered.

Semi-supervised methods can also take advantage of auxiliary domains. We propose the use of one or more source domains, each with relative contribution according to some measure of their similarity with the target domain, to induce initial labels on their corresponding instances from the target domain, and then apply label propagation or similar semi-supervised methods to annotate the unlabeled data.

Another open research topic that we aim to study is the use of interactive approaches to TL. The general idea is to design methods that propose TL hypotheses (e.g. different transformations over the data, different predictive tasks, or different choices or combinations of source domains), and have a user validate or rank them. This is in contrast with active learning, where the user interacts with the method by providing labels to instances directly.

A possible implementation of this idea could follow a motivation similar to the boosting-trick to learn and combine locally accurate hypotheses as a joint source domain and task.

VII. Conclusion

High-quality annotated data is a cornerstone in many data-intensive tasks, yet labeling is an inconvenient process. It is expensive, time-consuming, and error-prone. To make things worse, most annotated datasets are one-use-only, and as a result the labeling effort put on them is mostly underutilized.

The strategies provided by TL to reuse knowledge are especially relevant nowadays, as the amount of data obtained from high-throughput experiments vastly surpases what can be processed, let alone labeled.

Biomedical imaging is one area that can greatly profit from TL, since the datasets are obtained by following well-defined protocols, which translates into some consistency preserved over acquisitions. Furthermore, geometric and graphical properties of the data can be exploited alongside the features extracted for particular elements of the image.

Many of the current TL methods impose extra premises on the data to circumvent the same distribution assumption. The main assumption of TL is that the source and target domain are related, which in practice is often difficult to assess and quantify, and mostly relies on the intuition of the user and their knowledge of the problem.

The above suggests that fully-automated approaches to TL pose the risk to overlook important aspects of the problem. Hence, we propose to investigate on strategies that combine automation with expert user supervision.

We expect that the ideas hereby presented, and their eventual implementation, will improve the state-of-the-art on TL, and help make a smarter use of knowledge available from diverse sources.

References

[1] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu, "Text classification without negative examples revisit," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 1, pp. 6–20, 2006.

[2] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Proc. Workshop Transfer Learning for Complex Task of the 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, 2008.

[3] J. Blitzer, M. Dredze, F. Pereira *et al.*, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *ACL*, vol. 7, 2007, pp. 440–447.

[4] C. Becker, C. Christoudias, and P. Fua, "Domain adaptation for microscopy imaging," 2014.

[5] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Joint transfer and batch-mode active learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, 2010.

[7] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 193–200.

[8] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 505–512.

[9] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2006, pp. 601–608.

[10] J. Garcke and T. Vanck, "Importance weighted inductive transfer learning for regression," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 466–481.

[11] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[12] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of 2006 conference on empirical NLP methods*. Assoc. Computational Linguistics, 2006.

[13] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.

[14] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *AAAI*, vol. 7, 2007.

[15] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 27–32.

[16] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *Machine Learning & Knowledge Discovery Databases*. Springer, 2008.

[17] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, 2006.