Active Transfer Learning Using Knowledge of Anticipated Changes

Matthew O. Williams, Hala Mostafa
United Technologies Research Center, East Hartford, CT
william1,mostafh@utrc.utc.com

Abstract

Despite the furor around Big Data in recent years, data scarcity is still an issue in many industrial settings where the relevant data sets can be costly to obtain or only available on a set schedule. One approach to mitigate the data scarcity issue is transfer learning (TL) whose goal is to leverage knowledge from a data rich (source) task to improve learning performance on a different but related (target) task. TL approaches typically require large amounts of unlabeled target data which are often unavailable in industrial organizations. However, domain experts in these organizations usually have a qualitative understanding of the differences between the source and target tasks. We propose a TL approach that can leverage an expert's *anticipated changes (ACs)* between the source and target tasks to improve performance on the target task. We give examples of ACs and show how to incorporate them in two settings: the covariate shift setting where ACs bound the relative frequency of different types of observations, and the functional change setting where they provide bounds on differences between source and target prediction/labeling functions. To reduce the number of expert queries required, we also present an *active* TL approach that solicits ACs based on the current level of prediction uncertainty. We demonstrate the improved performance of our AC-enriched learners compared to "blind" uninformed TL on business-inspired datasets.

1 Introduction

In recent years, Big Data and data-mining techniques have become widely used in both academic and commercial settings. However, the data needed for these techniques can often be expensive or time-consuming to obtain. Furthermore, many problems of interest have an inherent temporal component, so the data and/or models generated at one time may become less relevant at later times. In the example we will present, our objective to predict the maintenance cost of units in a new system given historical data about the cost for units from an older but related system. Although the two systems are likely to share components, the costs can differ due to changes in the nature or pricing of maintenance procedures, and as a result, the accuracy of models trained solely on the historical data will diminish. While updated models can be generated, the cost and time associated with obtaining a new data set reduces the benefit the model can provide.

The ubiquity of this problem led to the development of *transfer learning* (see, e.g., [Pan and Yang, 2010] and the references therein) and concept drift (see [Gama *et al.*, 2014] and the references therein) techniques. In transfer learning (TL), the goal is to leverage relevant data from the original (source) task to boost performance in the new (target) setting. A key assumption in TL is that $P_S(x, y) \neq P_T(x, y)$, that is, the joint distribution of features and labels in the source domain (P_S) is different from the distribution in the target domain (P_T), so some form of adaptation is necessary. There is a taxonomy of problems within TL based on the differences between P_S and P_T. In this paper, we address *covariate shift* where the distribution over the covariates has changed ($P_S(x) \neq P_T(x)$), but the conditional distribution has not, and *functional changes* where the conditional distribution is different ($P_S(y|x) \neq P_T(y|x)$) but the distribution of covariates is not.

In many commercial settings, the data are not the only sources of information. For example, we often have access to subject matter experts who understand the details of how the old and new learning tasks relate. Machine learning approaches vary in the kinds of expert knowledge they can incorporate and when they seek to elicit that knowledge. For example, explanation-Augmented SVMs [Sun and DeJong, 2005] use domain knowledge encoded in generalized or explained examples to identify the "important" features in the explained instance that are allowed to contribute to the kernel inner product evaluation. This knowledge is incorporated as hard and soft constraints in the case of accurate and inaccurate knowledge, respectively. There have also been several efforts to leverage expert knowledge in learning graphical models. For example, expert judgments in the form of linear inequality and approximate equality constraints on Bayesian network parameters was shown to improve learning accuracy [Zhou *et al.*, 2014]. Other forms of expert knowledge are used to impose network structure (e.g. existence or absence of edges between nodes) [de Campos and Castellano, 2007], or guide the process of finding the maximum likelihood parameters [Altendorf *et al.*, 2005],

While the aforementioned methods target different types of expert information, the knowledge collection process is performed "offline," and not updated in light of the improved model predictions. In contrast, *active learning* in classification and regression tasks judiciously solicits knowledge in the form of labels based on metrics that quantify the benefits of querying the expert with different unlabeled instances [Schohn and Cohn, 2000; Cohn *et al.*, 1996a; Roy and McCallum, 2001]. Engaging the expert is interleaved with learning, with the benefit of a given potential query being re-assessed based on data provided by the expert so far. There has even been work on combining active and transfer learning into *active transfer learning (ATL)* that addresses the shortage of labels in the target task using both the source task and labels obtained from an expert by an active learning scheme. Early efforts performed the two kinds of learning independently and in sequence [Rai *et al.*, 2010]. Later work simultaneously does active and transfer learning for regression using Gaussian processes [Wang *et al.*, 2014].

A common limitation of existing ATL approaches is the level of detail at which knowledge must be elicited from the expert. In our running example, an expert is unlikely to be able to accurately price a unit with a given configuration (hence the need for a data-driven model). At best, the expert has knowledge of high-level **anticipated changes (AC)** between the source and target tasks. For example, the expert can tell us how they anticipate the target covariate distribution to change from the source distribution by indicating if an instance (or a group of instances) are more likely to occur under the source or target distributions. For the functional change setting, again instead of an expert providing a y value for an instance, it is easier to describe how the labeling function is expected to change in the form of upper/lower bounds on the difference between the source and target labeling functions.

We present an approach that combines transfer and active learning where our interactions with experts are limited to soliciting qualitative knowledge in the form of anticipated changes rather than explicit target labels. In the covariate shift setting, we integrate this knowledge as constraints on the learning procedure. In the functional change setting, we build on previous work on ATL with Gaussian processes [Wang *et al.*, 2014], extending it using constrained Gaussian processes [Da Veiga and Marrel, 2012] to incorporate this qualitative knowledge into their framework. The objective in both cases is to demonstrate that the accuracy of the resulting model can be improved by taking both historical data and qualitative expert knowledge into account.

The remainder of the manuscript is outlined as follows: in Sections 2 and 3, we examine the covariate shift and functional change settings, respectively. For each setting, we review existing TL approaches, give examples of expert knowledge encoding anticipated changes and show how we incorporate them into the TL approaches. We demonstrate the improved performance of TL with anticipated changes compared to baseline "uninformed" TL on a synthetic dataset and a dataset inspired by the running example from one of our business partners. We conclude and discuss potential future work in Section 4.

2 Covariate Shift Settings

Learning can generally be formulated as an optimization problem to find the prediction function f^* that minimizes the expected value of a loss function L over the joint distribution of covariates and labels $P(x, y)$, where

$$\begin{aligned} f^* &= \arg\min_{f\in\mathcal{F}} \mathbb{E}_T[L(f(x), y)], \\ &= \arg\min_{f\in\mathcal{F}} \int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} L(f(x), y)\frac{P_T(x, y)}{P_S(x, y)}P_S(x, y)\; dxdy, \\ &= \arg\min_{f\in\mathcal{F}} \mathbb{E}_S\left[\frac{P_T(x, y)}{P_S(x, y)}L(f(x), y)\right]. \end{aligned}$$

In settings with covariate shift, we assume $P_S(Y|X) = P_T(Y|X)$, which simplifies the above to:

$$\begin{aligned} f^* &= \arg\min_{f\in\mathcal{F}} \mathbb{E}_S\left[\frac{P_T(x)}{P_S(x)}L(f(x), y)\right] \\ &= \arg\min_{f\in\mathcal{F}} \mathbb{E}_S\left[w(x)L(f(x), y)\right] \end{aligned}$$

The weight w is the ratio of the target and source covariate distributions, known as the *density ratio*. From the above, we can see that if we have access to the weights $w(x)$, the prediction function that minimizes loss in expectation over P_T also minimizes loss in expectation over P_S when we weigh each instance by its density ratio. As such, the goal in many covariate shift approaches is to estimate w [Sugiyama and Storkey, 2006; Sugiyama *et al.*, 2010; 2012; 2013; Pan and Yang, 2010; Smola *et al.*, 2007].

2.1 Kernel mean matching

The following optimization problem finds w that minimize the difference between the first moments of $P_T(x)$ and its approximation $P_S(x)w(X)$:

$$\min_w \left\| \int xw(x)P_S(x)dx - \int xP_T(x)dx \right\|^2$$

However, finding the weights that result in an approximate distribution matching a finite number of moments of the target distribution does not necessarily give the true density ratio, even with infinitely many samples [Huang *et al.*, 2006; Gretton *et al.*, 2009]. Kernel mean matching (KMM) uses the kernel trick to do infinite-order moment matching, solving the following problem (see KMM references for details):

$$\min_\beta \left\| \frac{1}{N_S}\sum_{i=1}^{N_S} \beta_i\Phi(x_i) - \frac{1}{N_T}\sum_{i=1}^{N_T} \Phi(x_i) \right\|^2 \qquad (1)$$

where β_i and $\Phi(x_i)$ are x_i's weight and feature vector.

Let K be a kernel matrix $K_{ij} = k(x_i, x_j)$ and $\kappa_i = \frac{N_S}{N_T}\sum_{j=1}^{N_T} k(x_i, x_j)$ encode the similarity between each source instance and the target instances. The objective function can now be rewritten as

$$\min_\beta \frac{1}{N_S^2}\beta' K\beta - \frac{2}{N_S^2}\kappa'\beta + \text{const.}$$

Imposing a soft constraint on the weights β guarantees we produce a valid probability, and yields the quadratic program:

$$\min_\beta \frac{1}{2}\beta' K\beta - \kappa'\beta \quad \text{s.t.} \qquad \left|\sum_{i=1}^{N_S} \beta_i - N_S\right| \le \epsilon N_S$$

2.2 Incorporating ACs: Point-wise constraints

In our maintenance cost prediction example, a company may not have a large sample of configurations of a new system to estimate $P_T(x)$ or provide a good sample average approximations for Eq 1. However, we found that in many business settings domain experts can tell us something about the relative likelihood of (groups of) instances. If the maintenance cost depends on the condition of system components $x_{1..k}$, the engineers who built the system may know that the manufacturing process of component x_j has greatly improved in the new system, resulting in far fewer expected units having x_j in poor condition. Therefore, all else being equal, an instance $x^{(i)}$ having x_j=poor has a lower probability under P_T than under P_S, leaving it with $w(x) < 1$.

In KMM, if the AC is known for specific (groups of) instances, we impose *point-wise constraints (PWCs)* on their weights (β) . The AC can be in the form of exact values of the ratio, or an upper/lower bound. Despite their simplicity, introducing PWCs can easily lead to infeasible problems where the legality constraint and the PWCs are in conflict[1]. The new constraints do not capture the fact that the AC concerns the marginal distribution over x_j, ignoring the effect of other variables on the density ratio.

We ran KMM with AC on datasets inspired by our business case where we predict maintenance costs of units based on conditions of 8 components $x_{1..8}$ and $\frac{P_T(x_4=a)}{P_S(x_4=a)} \approx 2.7$. For each PWC, the expert must provide a lower or upper bound on w evaluated at the relevant point, and in practice, an expert may over/underestimate the true value of 2.7. Figure 1(a) (resp. 1(b)) shows how the difference between w provided in the expert ACs and the true w affects the R^2 value of our predictions when $P_S(x_{i\neq 4}) = P_T(x_{i\neq 4})$ (resp. $P_S(x_{i\neq 4}) \neq P_T(x_{i\neq 4})$). For small values, the expert is overly conservative in their predictions, and their knowledge of the AC does not provide any meaningful benefit. As the expert imposed ratio increases, so does the R^2 value up to around 2.7 where performance begins to degrade because the expert is over-estimating the values of w associated with a given value of x_4. While this flawed information still provides some benefit, the risk of an overly "aggressive" expert is captured by the drop around 3.5, where the inequality constraints provided by the expert create an infeasible optimization problem. The values reported in Fig. 1(a) beyond this point are the baseline performance of of training on (unweighted) source data.

2.3 Incorporating ACs: Set-wise constraints

We now present SWCs that (unlike PWCs) exactly capture expert knowledge on how the *marginal* distribution over one or more covariates changed at a given point. $x = [x_1; x_2]$ where x_1 is the covariate mentioned in the AC and x_2 is "everything else". Assume the expert AC concerns the density ratio of instances where the condition $\phi(x_1)$ holds.

[1]We originally tried incorporating ACs into KLIEP [Sugiyama *et al.*, 2008], another density ratio estimation approach. However, KLIEP is more susceptible to infeasibilities than KMM, since KMM is non-parametric, with more latitude in changing its instance weights β while KLIEP uses a parametric form to represent w.

We express marginal distributions in terms of integrals of the joint distribution, and approximate using the data.

$$P_S(\phi(x_1)) = \int_{x_1 \times x_2} I[\phi(x_1)] P_S(x_1, \vec{x}_2)\, dx_1 dx_2,$$
$$\approx \frac{1}{N_S} \sum_{i=1}^{N_S} I[\phi(x_1^{(i)})],$$

where δ and I are the Dirac delta and indicator functions.

$$P_T(\phi(x_1)) = \int_{x_1 \times x_2} I[\phi(x_1)] P_T(x_1, x_2)\, dx_1 dx_2,$$
$$= \int_{x_1 \times x_2} I[\phi(x_1)] w(x_1, \vec{x}_2) P_S(x_1, x_2)\, dx_1 dx_2,$$
$$\approx \frac{1}{N_S} \sum_{i=1}^{N_S} I[\phi(x_1^{(i)})] w_i.$$

For an AC that instances satisfying $\phi(x)$ are at least M times more likely, we get the SWC:

$$\frac{\sum_{i=1}^{N_S} I[x_1^{(i)} = a] w_i}{\sum_{i=1}^{N_S} I[x_1^{(i)} = a]} \geq M,$$

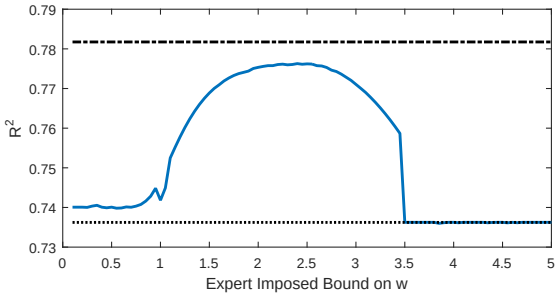
which simplifies to

$$\frac{\sum_{i=1}^{N_S} I[x_1^{(i)} = a] w_i}{N_{S_\phi}} \geq M,$$

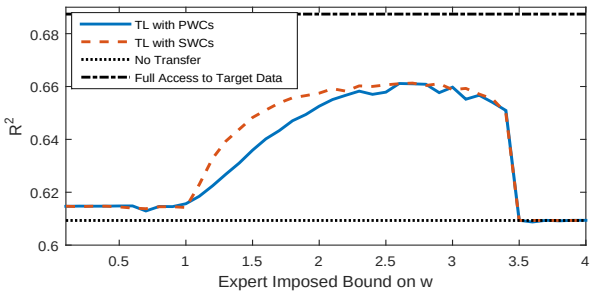
where N_{S_ϕ} is the number of instances satisfying ϕ. The SWC therefore constrains the *average weight* of these instances (the LHS) while PWCs impose a constraint on *the weight of each instance* in that set. In settings where the expert provides ACs for only some of the covariates whose distributions have changed, SWCs better capture the change in the marginal distributions, as illustrated in Figure 1(b).

Because the underlying distribution of covariates is more complex than in the previous section (the distributions of multiple covariate have changed), the R^2 values of all learners in Fig. 1(b) are lower than in Fig. 1(a). However, the injection of expert knowledge is still able to produce a meaningful improvement in the quality of the prediction. The red and blue curves in Fig. 1(b) show the R^2 value as a function of the expert-imposed ratio incorporated as SWC and (approximate) PWCs, respectively. Near the optimal value of 2.7, both methods have equivalent $R^2 \approx 0.66$. The benefit of SWCs is that they produce larger R^2 values even when expert knowledge is inaccurate. This additional robustness is important in practical settings where the true ratio is not known, and even an expert's best set of constraints will over- or underestimate the true value. Similar to PWCs, SWCs can result in infeasible optimization problems if the expert is aggressively over- or under-estimating the ratio..

In this section, we introduced an approach to the covariate shift problem in TL that incorporated expert knowledge in the form of inequality constraints. Intuitively, these constraints bound the relative likelihood of either individual instances (PWCs) or sets of instances (SWCs) in the source and target data sets. Using KMM as our base TL, we demonstrated the



(a) AC encoded in PWCs for $P_S(x_{i\neq 4}) = P_T(x_{i\neq 4})$.



(b) AC encoded in PWC vs SWC for $P_S(x_{i\neq 4}) \neq P_T(x_{i\neq 4})$.

Figure 1: Effect of w provided in the AC of x_4 on KMM R^2 value. For reference "No Transfer" indicates the R^2 value obtained when all source points are given equal weight, and "Full Access to Target Data" is the performance achieved if we are given the full set of target labels. In this example, setting the expert imposed bound to 0 is equivalent to "blind" transfer learning.

benefit of both types of expert knowledge on a business inspired problem, and demonstrated increasing accuracy as the expert's ACs better captured the changes in the underlying distributions.

3 Functional Change Settings

In functional change problems, we assume that the conditional distributions in the source and target domains are different (i.e., $P_s(y|x) \neq P_t(y|x)$) but that the marginal distributions of the covariates remain unchanged (i.e., $P_s(x) = P_t(x)$). We build on and extend the "offset" approach presented in [Wang *et al.*, 2014] to incorporate qualitative expert knowledge about the differences between the source and target data sets. In what follows, this expert knowledge is in the form of an upper and lower bound on the difference between $P_s(y|x)$ and $P_t(y|x)$. As in the previous section, this could be point-wise knowledge such as "the output at this point will increase by between two and ten units" or set-wise knowledge such as "the average value of these points will decrease though certain individuals may have larger values."

In the remainder of the section, we briefly review constrained Gaussian processes (GP), and outline the offset approach to transfer learning based on constrained GPs. We then introduce our modifications to incorporate expert knowledge. Finally, we propose a simple iterative active learning scheme for selectively querying the expert rather than requiring they provide all their knowledge upfront, and demonstrate it on a industrially inspired problem of predicting maintenance costs based on damage levels.

3.1 Constrained Gaussian Processes

In the examples that follow, we choose our models of the source and target functions to be inequality constrained Gaussian processes (GPs) as defined in [Da Veiga and Marrel, 2012]. Although we refer the reader to that work for details, we will reproduce the salient features of the method here.

Suppose our objective is to approximate the function $y = f(x)$. Following the approach in [Da Veiga and Marrel, 2012], there are three types of inputs: (i) labeled data of the form $\{(x_i^l, y_i^l)\}_{i=1}^{N_l}$, (ii) inequality constrained data of the form $\{(x_i^c, a_i, b_i)\}_{i=1}^{N_c}$ where $a_i \leq f(x_i^c) \leq b_i$, and (iii) unlabeled data where we want to make predictions $\{x_i^*\}_{i=1}^{N_*}$. Neglecting constraints for the moment, the underlying assumption GPs make is that the vector

$$\begin{bmatrix} y^l \\ y^c \\ y^* \end{bmatrix} \sim \mathcal{N}(0, \Sigma),$$

where $y^l = \{y_1^l, y_2^l, \ldots\}$, $y^c = \{f(x_1^c), f(x_2^c), \ldots\}$, and $y^* = \{f(x_1^*), f(x_2^*), \ldots\}$ are vectors containing the predicted function values. The covariance matrix

$$\Sigma = \begin{bmatrix} \Sigma_{ll} & \Sigma_{lc} & \Sigma_{l*} \\ \Sigma_{cl} & \Sigma_{cc} & \Sigma_{c*} \\ \Sigma_{*l} & \Sigma_{*c} & \Sigma_{**} \end{bmatrix},$$

is computed by evaluating a kernel function K for all pairs of x_i^l, x_i^c, and x_i^*. The conditional probabilities of y^c and y^* given the labels are:

$$P\left(\begin{bmatrix} y^c \\ y^* \end{bmatrix} \middle| y^l \right) \sim \mathcal{N}(\mu_1, \Sigma_1), \tag{2a}$$

where

$$\mu_1 = \begin{bmatrix} \Sigma_{cl} \\ \Sigma_{*l} \end{bmatrix} \Sigma_{ll}^{-1} y^l, \tag{2b}$$

$$\Sigma_1 = \begin{bmatrix} \Sigma_{cc} - \Sigma_{cl}\Sigma_{ll}^{-1}\Sigma_{lc} & \Sigma_{c*} - \Sigma_{cl}\Sigma_{ll}^{-1}\Sigma_{l*} \\ \Sigma_{*c} - \Sigma_{*l}\Sigma_{ll}^{-1}\Sigma_{lc} & \Sigma_{**} - \Sigma_{*l}\Sigma_{ll}^{-1}\Sigma_{l*} \end{bmatrix}. \tag{2c}$$

To implement constraints, we replace the normal distribution with a truncated multinormal distribution:

$$P\left(\begin{bmatrix} y^c \\ y^* \end{bmatrix} \middle| y^l \right) \sim \mathcal{TN}(\mu_1, \Sigma_1 | a_i \leq y_i^c \leq b_i \forall i = 1, \ldots, N_c).$$

To draw samples from this distribution, we used a Gibbs sampler such as the ones described in [Geweke, 1991] and [Rodriguez-Yam *et al.*, 2004]. To reduce the computational cost of the method and exploit the fact that many of the data points may be unconstrained, [Da Veiga and Marrel, 2012] noted that ν_*, the mean of the unconstrained points, and Γ_*, the covariance matrix of the unconstrained points, are:

$$\nu_* = \Sigma_{*c}\Sigma_{cc}^{-1}\nu^c + \Sigma_{*l}\Sigma_{ll}^{-1}y^l, \tag{3a}$$

$$\Gamma_* = \Sigma_{**} - \Sigma_{*c}\left(\Sigma_{cc}^{-1} - \Sigma_{cc}^{-1}\Gamma_{cc}\Sigma_{cc}^{-1}\right)\Sigma_{c*}, \tag{3b}$$

where ν^c and Γ_{cc} are the mean and covariance matrix of the subset of data points with the constraints imposed. We approximate these moments empirically via the Gibbs sampler, and use these expressions to draw samples from a N_c-dimensional truncated normal distribution rather than an $(N_c + N_*)$-dimensional distribution, which reduces the cost associated with the Gibbs sampling procedure.

3.2 The Offset Approach

In this subsection, we outline our modification of the offset approach for functional change, which was first presented by [Wang *et al.*, 2014]. First, we will review the "standard" offset approach from that manuscript, and then show how we incorporate expert knowledge of ACs.

The problem formulation for the offset approach assumes we are given a set of N_s labeled data points from the source domain, $\{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ where the relationship between the features and labels is modeled by the (unknown) function f^s. Furthermore, we are given N_t labeled data points from the target domain, $\{(x_i^t, y_i^t)\}_{i=1}^{N_t}$ with the assumption $N_t \ll N_s$. The features and labels in the target domain are related by the function f^t where we have assumed $f^t \neq f^s$, and both the source and target levels are affected by Gaussian noise with a standard deviation of σ. New to this manuscript are N_c data points on which an expert has qualitative knowledge about the differences between the target and source tasks. For simplicity of exposition, we assume these are constraints on the *offset* in the function values (i.e., $\{(x_i^c, a_i \leq f^t(x_i^c) - f^s(x_i^c) \leq b_i)\}_{i=1}^{N_c}$), but one could also impose constraints on derivatives or integrals. As before, our objective is to use these data sets to learn a model for the target task (i.e., approximate f^t) where the primary challenge is the scarcity of labeled target data.

Outline of the Method

The offset approach decomposes the target task f^t into

$$f^t(x) = f^s(x) + \Delta f(x), \tag{4}$$

where Δf is the *offset* between the source and target tasks. The accuracy of the offset approach depends on the complexity of Δf, where the main assumption is that Δf is smoother than f^t and can therefore be learned with fewer data points [Xuezhi Wang, 2015].

The offset approach consists of two main steps: (1) approximate the function f^s, and (2) approximate the offset Δf. To accomplish step (1), we model the source data using a GP. This will be a straightforward application of a GP, but the results will be important in step (2). The GP assumes that $P(y^s|x) = \mathcal{N}(0, K_s(x, x))$ where $K_s(x, x)$ is the $N \times N$ covariance matrix associated with the kernel function $K_s(\cdot, \cdot)$ and $y^s = [f^s(x_1), f^s(x_2), \ldots]$. Assuming the source data has noise that is independently and normally distributed with variance σ^2, the predicted values of f^s at the points in x^t are:

$$P(f^s(x^t)|x^t, x^s, y^s) = \mathcal{N}(\mu_s, \Sigma_s), \tag{5}$$

where $\mu_s = K_s(x^t, x^s)(K_s(x^s, x^s) + \sigma^2 I)^{-1} y^s$ and $\Sigma_s = K_s(x^t, x^t) - K_s(x^t, x^s)(K_s(x^s, x^s) + \sigma^2 I)^{-1} K_s(x^s, x^t)$.

In step (2), we approximate the offset function. This task was originally accomplished using a GP, but we will use a constrained GP (Sec. 3.1) so that expert-provided constraints can be imposed in the next subsection. We approximate the offset function in two steps: (2a) generate "offset data", and (2b) fit a GP using this offset data.

In step (2a), we approximate the offset at the target data points using (5) and the limited number of labeled target points. If $f^s(x^t)$ is the approximation of the source function value at the target points using (5), then: $\Delta y^t = y^t - f^s(x^t)$ where f^s is the source GP from step (1). Due to noise and prediction uncertainty, $\Delta y^t \sim \mathcal{N}(y^t - \mu_s, \Sigma_s + \sigma^2 I)$.

In step (2b), we use the offset data to approximate Δf using a Gaussian process with kernel K_Δ. The conditional distribution of the offset at some new set of points x^* is:

$$P(\Delta f(x^*)|x^*, x^t, \Delta y^t) = \mathcal{N}(\mu_\Delta, \Sigma_\Delta), \tag{6}$$

where $\mu_\Delta = K_\Delta(x^*, x^t)(K_\Delta(x^t, x^t) + \Sigma_s + \sigma^2 I)^{-1}(y^t - \mu_s)$ and $\Sigma_\Delta = K_\Delta(x^*, x^*) - K_\Delta(x^*, x^t)(K_\Delta(x^t, x^t) + \Sigma_s + \sigma^2 I)^{-1} K_\Delta(x^t, x^*)$. As before, $K_\Delta(x^i, x^j)$ is the $N_i \times N_j$ covariance matrix generated by evaluating the kernel function K_Δ for all pairs of the data in sets x^i and x^j.

In their original manuscript, Wang et al. included a third step which explicitly generated an approximation of f^t. We forgo this step and generate our approximation of f^t via (4) and the approximations of f^s and Δf. This difference is for more than just convenience; as we will show in the next section, a consequence of incorporating expert knowledge is that the resulting distribution of predictions may no longer be Gaussian, so uncertainty can no longer be accurately propagated using GPs.

Soliciting and Incorporating Expert Knowledge

Our main contribution to the offset method is the ability to incorporate expert knowledge in lieu of additional labeled target points. For classes of expert knowledge that can be represented as inequality constraints, such as the point-wise and set-wise constraints mentioned earlier, this can be accomplished in a straightforward manner *because we approximate the offset function using constrained GPs*. In particular, we evaluate μ_Δ and Σ_Δ at points x^c along with points where predictions are desired, which generates the conditional distribution shown in (2). Next we impose the constraints and draw samples from the resulting truncated normal distribution using a Gibbs sampler. Then we approximate the expected sample values and their covariances at the points of interest using (3) (or sample from the distribution using the Gibbs sampler).

Equally important as the ability to incorporate expert knowledge is the ability to identify where this knowledge would be useful. For that, we adapt techniques from active learning to help guide our interactions with experts. Traditional active learning approaches require labels, which can be problematic in certain commercial settings where target labels will become available on a pre-determined schedule (e.g., the cost of maintenance can only truly be evaluated after the maintenance is performed). Due to the larger set of possible responses from the expert (e.g., tight pair of bounds, tight bounds on one side, or uninformative bounds) and the lack of expert models characterizing their accuracy, computing optimal sequential policies for querying an expert remains an

open question. Instead, to show the potential benefit of active learning with an expert that provides qualitative information rather than labels, we use the variance reduction methods in [Wang *et al.*, 2014] and [Cohn *et al.*, 1996b], but ask the expert for ACs, rather than actual labels, at query points. While this approach is suboptimal , and clearly dependent on the quality of the expert, it appears to be effective in practice as we will show in the following section.

3.3 Experimental Results

In this section we apply our approach to a business inspired problem where the we predict the maintenance cost of units given the expected condition of their components at the time of replacement. In this example, we consider the effects of a change in the maintenance procedure where the cost of replacing more damaged components has increased due to the cost of new parts that must be replaced. The true maintenance cost will not be available until the unit comes in for repair, but the domain expert can provide estimates of the difference in the pre- and post-change costs given the expected conditions of the units.

In this example, we have 200 pairs of unit conditions and their associated maintenance cost that were obtained from historical data. The expected condition of a unit is determined by eight features, which correspond to the expected levels of damage in the modules that comprise it. Given this data, we fit a GP with the kernel

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \tag{7}$$

where $\sigma = 22$, which was chosen using MLE, and had $R^2 = 0.44$. Data for the new unit is sparse; although we have the expected levels of damage for a set of 200 units, costs are only available for ten of them. Due to the lack of data, a GP using only the labeled target data has an $R^2 = -0.20$, and due to the change in maintenance policy, reusing the old model as-is results in $R^2 = 0.13$.

Figure 2 shows the R^2 values for the offset approach using all the labeled source data, the 10 labeled target data points, and increasing amounts of expert knowledge. In the absence of expert knowledge, the offset method has $R^2 = 0.24$. We compare 2 schemes for choosing the next point to query the expert for: the simple active learning method described in Sec. 3.1 and a method that randomly selects queries. To enable a fair comparison to be made, the "expert" provides lower and upper bounds (a_i and b_i respectively). We ensure the bounds contain the true offset value by setting them to:

$$a_i = \Delta f(x_i^s) + \mathcal{U}(-0.2|\Delta f(x_i^s)|, 0) + \mathcal{U}(-0.1, 0),$$
$$b_i = \Delta f(x_i^s) + \mathcal{U}(0, 0.2|\Delta f(x_i^s)|) + \mathcal{U}(0, 0.1).$$

The bounds associated with all possible expert queries were computed beforehand, so both the active and random methods receive the same information if the expert is queried about the same point. In both schemes, the inclusion of expert knowledge increases the R^2 value by more than a factor of two after five expert queries. For this realization, the active learning approach outperforms random selection for the ten iterates (queries) shown here, ultimately obtaining a value of $R^2 = 0.57$.

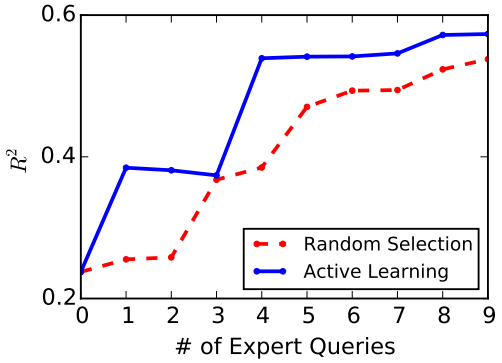


Figure 2: The R^2 value as a function of the number of expert interactions using the active learning for the business case inspired example.

To summarize, this section presented a method for incorporating expert knowledge of anticipated changes (ACs) into the offset method. While qualitative forms of expert knowledge are not necessarily as informative as labels, they were able to provide a measurable increase in the R^2 values. This improvement is obtained even if the expert is asked about random points, but we observed that using a simple active learning approach produces larger increases in R^2 given the same number of expert queries. As a result, in situations where labeled target data are scarce and there is limited opportunity for soliciting expert knowledge, a combination of active and expert-guided transfer learning can result in meaningful improvements over purely data-driven methods.

4 Conclusion

The abundance of unlabeled target data and the ability to query experts for precise labels are assumptions ingrained in most transfer and active learning approaches, respectively. However, industrial organizations often have very limited data from the target task (even unlabeled) and providing exact labels can be a costly or impractical process. We addressed these shortcomings of TL and AL approaches by leveraging high-level, intuitive and generalizable domain expert knowledge of the anticipated changes (AC) between the source and target tasks. We gave examples of ACs and show how to incorporate them in two TL settings; in a covariate shift setting ACs form constraints on the learning procedure while in a functional change setting we use constrained Gaussian processes to capture ACs in the prediction function. We also presented an *active* TL approach that solicits ACs from an expert to reduce prediction uncertainty. We demonstrated the improved performance of our AC-enriched learners compared to "blind" uninformed TL on business-inspired datasets.

References

[Altendorf *et al.*, 2005] Eric E Altendorf, Angelo C Restificar, and Thomas G Dietterich. Learning from sparse data by exploiting monotonicity constraints. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[Cohn *et al.*, 1996a] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

[Cohn *et al.*, 1996b] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

[Da Veiga and Marrel, 2012] Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. In *Annales de la faculté des sciences de Toulouse*, volume 21, pages 529–555, 2012.

[de Campos and Castellano, 2007] Luis M de Campos and Javier G Castellano. Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, 45(2):233–254, 2007.

[Gama *et al.*, 2014] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[Geweke, 1991] John Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computing science and statistics: Proceedings of the 23rd symposium on the interface*, pages 571–578. Citeseer, 1991.

[Gretton *et al.*, 2009] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[Huang *et al.*, 2006] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Rai *et al.*, 2010] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32. Association for Computational Linguistics, 2010.

[Rodriguez-Yam *et al.*, 2004] Gabriel Rodriguez-Yam, Richard A Davis, and Louis L Scharf. Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression. *Unpublished manuscript*, 2004.

[Roy and McCallum, 2001] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *ICML*, pages 441–448, 2001.

[Schohn and Cohn, 2000] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846. Citeseer, 2000.

[Smola *et al.*, 2007] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

[Sugiyama and Storkey, 2006] Masashi Sugiyama and Amos J Storkey. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems*, pages 1337–1344, 2006.

[Sugiyama *et al.*, 2008] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Bnau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *In NIPS*, 2008.

[Sugiyama *et al.*, 2010] Masashi Sugiyama, Motoaki Kawanabe, and Pui Ling Chui. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, 23(1):44–59, 2010.

[Sugiyama *et al.*, 2012] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[Sugiyama *et al.*, 2013] Masashi Sugiyama, Makoto Yamada, and Marthinus Christoffel du Plessis. Learning under nonstationarity: covariate shift and class-balance change. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):465–477, 2013.

[Sun and DeJong, 2005] Qiang Sun and Gerald DeJong. Explanation-augmented svm: an approach to incorporating domain knowledge into svm learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 864–871. ACM, 2005.

[Wang *et al.*, 2014] Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. Active transfer learning under model shift. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1305–1313, 2014.

[Xuezhi Wang, 2015] Jeff Schneider Xuezhi Wang. Generalization bounds for transfer learning under model shift. In *UAI*, 2015.

[Zhou *et al.*, 2014] Yun Zhou, Norman Fenton, and Martin Neil. Bayesian network approach to multinomial parameter learning using data and expert judgments. *International Journal of Approximate Reasoning*, 55(5):1252–1268, 2014.