Transfer Learning across Feature-Rich Heterogeneous Feature Spaces via Feature-Space Remapping (FSR)

Kyle D. Feuz, Washington State University
Diane J. Cook, Washington State University

Transfer learning aims to improve performance on a target task by utilizing previous knowledge learned from source tasks. In this paper we introduce a novel heterogeneous transfer learning technique, Feature-Space Remapping (FSR), which transfers knowledge between domains with different feature spaces. This is accomplished without requiring typical feature-feature, feature instance, or instance-instance co-occurrence data. Instead we relate features in different feature-spaces through the construction of meta-features. We show how these techniques can utilize multiple source datasets to construct an ensemble learner which further improves performance. We apply FSR to an activity recognition problem and a document classification problem. The ensemble technique is able to outperform all other baselines and even performs better than a classifier trained using a large amount of labeled data in the target domain. These problems are especially difficult because in addition to having different feature-spaces, the marginal probability distributions and the class labels are also different. This work extends the state of the art in transfer learning by considering large transfer across dramatically different spaces.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Heterogeneous Transfer Learning, Domain Adaption, Text Classification, Activity Recognition

ACM Reference Format:
Kyle D. Feuz and Diane J. Cook, 2014. Transfer Learning across Feature-Rich Heterogeneous Feature Spaces via Feature-Space Remapping (FSR) *ACM Trans. Intell. Syst. Technol.* V, N, Article A (January YYYY), 27 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Traditional supervised machine learning techniques rely on the assumptions that the training data and test data are drawn from the same probability distributions and that the classification task is the same for both datasets. However, in practice it is often convenient to relax these assumptions and allow the test data to be drawn from a different probability distribution or to allow the classification task to change. In these cases, traditional machine learning techniques often fail to correctly classify the test data.

As an example, consider the problem of activity recognition in a smart home. Based on motion sensor data of a particular resident in a particular home, a model can be trained to predict the current activity occurring in the home. However, the model may

Author's addresses: Kyle D. Feuz, School of Electrical Engineering and Computer Science, Washington State University; Diane J. Cook, School of Electrical Engineering and Computer Science, Washington State University;

DOI: http://dx.doi.org/10.1145/0000000.0000000

then be tested with a different resident, in a different home, or with different activity labels. If the model is not adapted to the new situations the prediction accuracy will typically drop significantly.

Transfer learning techniques have been proposed to specifically handle these types of situations. Transfer learning algorithms seek to apply knowledge learned from a previous task to a new, but related, task. The intuition behind transfer learning stems from the ability of humans to extend what has been learned in one context to a new context. In the field of machine learning, the benefits of transfer learning are numerous; less time is spent learning new tasks, less information is required of experts (usually human), and more situations can be handled effectively, making the learned model more robust. These potential benefits have led researchers to apply transfer learning techniques to many domains with varying degrees of success.

Most transfer learning techniques focus on situations where the difference between the source and target domains stems mainly from differences in the marginal probability distributions of the domains or different task labels [Cook et al. 2012; Pan and Yang 2010]. In this paper we propose a novel heterogeneous transfer learning technique, Feature-Space Remapping (FSR), which is capable of handling different feature spaces without the use of a translation oracle or instance-instance co-occurrence data. We term the technique a "remapping" because the original raw data is already mapped onto a feature space and FSR remaps the data to a different feature space. The technique can be used in either the informed or uninformed transfer learning setting and we provide details for both cases. FSR uses only a small amount of labeled data in the target domain to infer relations to the source domain and can optionally operate without any labeled data in the target domain or other linkage data. For simplicity, we present FSR here assuming the feature-space is a vector of real-valued numbers. However, it is straightforward to extend the FSR approach to handle categorical or discrete values as well.

In addition to presenting FSR for transferring knowledge from a single source domain to a target domain, we also show how FSR can effectively combine the information from multiple source domains by using an ensemble learner to increase the classification accuracy in the target domain. We illustrate our techniques using examples from activity recognition and document classification.

2. BACKGROUND

Many of the ideas and principles of machine learning have originated from comparisons and analogies to human learning . The same is true with transfer learning. The ability to identify deep, subtle connections, what we term *transfer learning*, is the hallmark of human intelligence. Byrnes [Byrnes 1996] defines transfer learning as the ability to extend what has been learned in one context to new contexts. Thorndike and Woodworth [Thorndike and Woodworth 1901] first coined this term as they explored how individuals transfer learned concepts between contexts that share common features. Barnett and Ceci provide a taxonomy of features that influence transfer learning in humans [Barnett and Ceci 2002].

In the field of machine learning, transfer learning is studied under a variety of names including learning to learn, life-long learning, knowledge transfer, inductive transfer, context-sensitive learning, and metalearning [Arnold et al. 2007; Elkan 2001; Thrun 1996; Thrun and Pratt 1998; Vilalta and Drissi 2002]. It is also closely related to self-taught learning, multi-task learning, domain adaptation, and co-variate shift. Because of this broad variance in the terminology used to describe transfer learning it is helpful to provide a formal definition of the terms we will use throughout the rest of this paper.

2.1. Definitions

Definitions for domain and task have been provided by Pan and Yang [Pan and Yang 2010]:

Definition 2.1 (*Domain*). A domain D is a two-tuple $(\chi, P(X))$. χ is the feature space of D and $P(X)$ is the marginal distribution where $X = \{x_1, ..., x_n\} \in \chi$.

Definition 2.2 (*Task*). A task T is a two-tuple $(Y, f())$ for some given domain D. Y is the label space of D and $f()$ is an objective predictive function for D. $f()$ is sometimes written as a conditional probability distribution $P(y|x)$. $f()$ is not given, but can be learned from the training data.

Using these terms, we can now define transfer learning. In this paper we use the definition given by Cook et al. [Cook et al. 2012] which is similar to that presented by Pan and Yang [Pan and Yang 2010] but allows for transfer learning from multiple source domains.

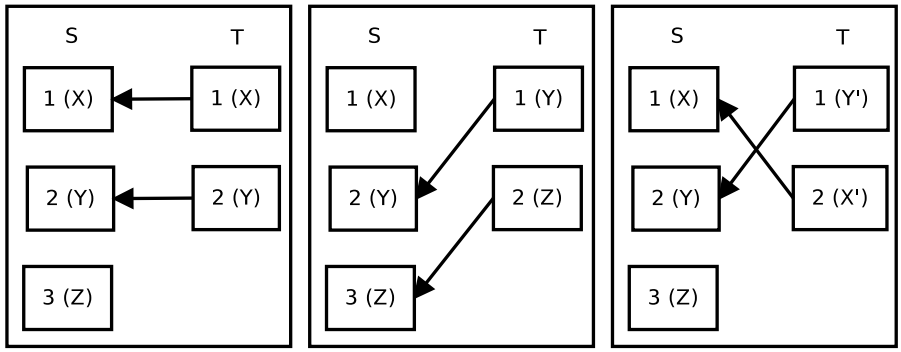
Definition 2.3 (*Transfer Learning*). Given a set of source domains $DS = D_{s_1}, ..., D_{s_n}$ where $n > 0$, a target domain, D_t, a set of source tasks $TS = T_{s_1}, ...T_{s_n}$ where $T_{s_i} \in TS$ corresponds with $D_{s_i} \in DS$, and a target task T_t which corresponds to D_t, transfer learning helps improve the learning of the target predictive function $f_t()$ in D_t where $D_t \notin DS$ and $T_t \notin TS$.

This definition of transfer learning is broad and encompasses a large number of different transfer learning scenarios. The source tasks can differ from the target task by having a different label space, a different predictive function for labels in that label space, or both. The source data can differ from the target data by having a different domain, a different task, or both. The FSR algorithm focuses on the challenge of the source and target domain coming from different feature spaces. This is commonly referred to as heterogeneous transfer learning in the literature and is formally defined below.

Definition 2.4 (*Heterogeneous Transfer Learning*). Given a set of source domains $DS = D_{s_1}, ..., D_{s_n}$ where $n > 0$, a target domain, D_t, a set of source tasks $TS = T_{s_1}, ...T_{s_n}$ where $T_{s_i} \in TS$ corresponds with $D_{s_i} \in DS$, and a target task T_t which corresponds to D_t, transfer learning helps improve the learning of the target predictive function $f_t()$ in D_t where $\chi_t \cap (\chi_{s_1} \cup ...\chi_{s_n}) = \emptyset$.

Although FSR focuses on different feature spaces, it does not rely on the other dimensions of the transfer learning problem remaining constant. Indeed the datasets we use in the experimental section have differences in the marginal probability distributions as well as in the label space. As with all transfer learning problems we do rely on the basic assumption that there exists some relationship between the source and target areas which allows for the successful transfer of knowledge from the source to the target.

When the feature spaces of the domains are different, we assume that they can be different both in terms of the number of dimensions and in the organization of the dimensions. To illustrate this point, consider two different domains, one consisting of two dimensional data and the other consisting of three dimensional data. It could be the case that the first two dimensions are the same in both domains (see Figure 1a); however, it could also be the case that the first two dimensions of the target domain correspond with the last two dimensions of the source domain (see Figure 1b), or perhaps only the first dimension of the target domain corresponds with the last dimension of the source domain. It may even be the case that the dimensions are entirely different, but a mapping between dimensions could still allow the knowledge gained in one



(a) Map 1 (b) Map 2 (c) Map 3

Fig. 1. Example mappings from target T (two-dimensional data) to source S (three-dimensional data)

domain to be used effectively in the other domain (see Figure 1c). FSR learns a mapping from the target feature space to the source feature space regardless of the exact differences between dimensions.

In traditional machine learning, there are three basic types of techniques that are utilized based upon the availability of labeled data: supervised, unsupervised, and semi-supervised. In transfer learning, the availability of labeled data can be different in the source and target domains, thus four general classes of techniques arise, informed supervised, informed unsupervised, uninformed supervised, uninformed unsupervised. We follow the definitions of Cook et al. [Cook et al. 2012], where informed or uninformed refers to the presence or absence, respectively, of labeled data in the target domain, and supervised or unsupervised refers to the presence or absence of labeled data in the source domain. We propose both uninformed and informed variations of FSR.

2.2. Related Work

Domain adaptation is a specific branch of transfer learning that targets the case when the source and target data are not from the same domain. However, most of those works assume the difference is in the marginal probability distribution of the domains.

Daumé and Marcu model the probability distribution using a mixture model [Daumé and Marcu 2006]. They assume that the source data comes from a mixture of a source probability distribution and a general probability distribution and that the target data similarly comes from a mixture of a target probability distribution and a general probability distribution. They then learn the parameters of these distributions from the data and use the source data to bolster the estimation of the the target data.

Several researchers apply feature-space transformations to overcome differences in the marginal probability distributions. Blitzer et al. [Blitzer et al. 2007; Blitzer et al. 2006] propose Structural Correspondence Learning (SCL) to use the correlation between certain pivot features (which have the same semantic meaning in both domains) and other features to create a common feature representation. Pan et al. [Pan et al. 2010] construct a bipartite graph with connections between pivot features and non-pivot features that contain co-occurring feature values. They then apply spectral clustering to align the features and create a common feature-space representation.

Daumé et al. transform the source and target feature spaces into a higher dimensional representation with source, target and common components [Daumé III 2007]. They then extend this to use unlabeled data by introducing co-regularization to force the source and target components to predict the same label on the unlabeled data

[Daumé et al. 2010]. Zhong et al. use kernel mapping to map features in the source and target domains to a new feature space where the conditional and marginal probabilities are more closely aligned [Zhong et al. 2009]. They prove that a classifier trained in the new feature space has a bounded error.

Using a different approach, Pan et al. [Pan et al. 2011] perform domain adaptation via dimensionality reduction. Using Transfer Component Analysis [Pan et al. 2008], they reduce the distance between domains by projecting the features onto a shared subspace. As in the previous approaches, the technique focuses on the differences in the distribution of the data and assumes the feature space is the same.

Chattopadhyay et al. use domain adaptation on multiple source domains to detect fatigue using SEMG signal data [Chattopadhyay et al. 2011]. Their algorithm combines the output from multiple source classifiers to predict a label for unlabeled data in the target domain. These data instances are then combined with labeled data in the target domain and a final classifier is built. The label predictions from the multiple source domains are combined using a weighted voting schemed where the weights are based upon the similarity between the source and target domain at a per-class level.

Heterogeneous transfer learning focuses on transfer learning problems where the source and target domains are different because they have different feature spaces. Dai et al. attempt solving the heterogeneous transfer learning problem by extending the risk minimization framework [Lafferty and Zhai 2001] and developing a translator between feature spaces based upon co-occurrence data (feature-feature, feature-instance, instance-feature, or instance-instance) between the source and target datasets [Dai et al. 2008]. Prettenhofer extends SCL to the heterogeneous transfer learning case by use a translation oracle (i.e. a domain expert or bi-lingual dictionary) to enumerate several pivot features. These pivot features are then correlated to the other features in both domains and a cross-lingual classifier is trained [Prettenhofer and Stein 2011].

Shi and Yu apply dimensionality reduction to heterogeneous feature spaces. In order to project the features from different feature spaces onto a single unified subspace they require that the data instances be linked as in multi-view learning. The ith data instance in the jth feature space is also the ith data instance in the kth feature space. Yang et al. extend the probabilistic latent semantic analysis (PLSA) [Hofmann 1999] to improve image clustering results [Yang et al. 2009]. Images features are clustered to latent variables while annotations from social media are simultaneously clustered to the same latent variables. By clustering both the annotations and the image features the overall clustering results are improved.

Manual mapping strategies have also been used to overcome differences in the feature spaces. For example, Van Kasteren et al. [van Kasteren et al. 2008; 2010] group sensors by their location/function. Sensors in the source domain are then mapped to similar sensors in the target domain. Rashidi and Cook also map sensors based on location/function but apply additional transfer learning techniques to better align the source and target datasets [Rashidi and Cook 2010; 2011]. Our approach eliminates the need to manually map the feature spaces as this is handled by the algorithm. Additional domain adaptation approaches can then be applied to further improve the knowledge transfer. FSR requires the manual specification of meta-features but this specification only occurs once and can be applied to map multiple source and target domains. The techniques of both Rashidi and Van Kasteren require a mapping to be defined for each source and target pair. Additionally, the manual mapping strategies are domain dependent, while FSR is applicable to a variety of different problems.

Each of the above mentioned heterogeneous techniques require some form of linkage (co-occurrence data, dictionaries, or domain experts) between the source and target dataset. FSR uses only a small amount of labeled data in the target domain to infer relations to the source domain and can optionally operate without any labeled data in

the target domain or other linkage data. Similar to our approach, Duan et al. do not require any co-occurrence data linking the two feature spaces [Duan et al. 2012]. Like Duame et al. [Daumé III 2007], they also transform the feature space into a higher dimensional representation with source, target and common components. However the common components are obtained by projecting the source and target features onto a common subspace. Their approach only handles the informed transfer learning problem as it requires some labeled data in the target domain.

2.3. Illustrative Example

Before describing FSR, we put forward an example transfer learning scenario to illustrate the concepts introduced throughout the discussion. To that end, let us consider the transfer learning problem for activity recognition in a smart environment using ambient sensors.

Ambient sensors are typically embedded in an individual's environment. Examples of ambient sensors may include motion detectors, door sensors, object sensors, pressure sensors, and temperature sensors. As the name indicates, these sensors are designed to disappear into the environment while collecting a variety of activity related information such as human movements in the environment induced by activities, interactions with objects during the performance of an activity, and changes to illumination, pressure and temperature in the environment due to activities.

Suppose there are two homes (a source home and a target home) equipped with these ambient sensors. The source home already has an activity recognition model trained for that home. The target home does not yet have an activity recognition model trained. In order to use the model from the source home to recognize activities in the target home, they must use a common feature-space.

A common approach to activity recognition using ambient sensors is to formulate the problem as a bag of sensors approach over some sliding window of time or sensor events. This means that the sensors from one home must be mapped onto the sensors from the other home. Specifically, the features of one domain must map onto the features (or dimensions) of the other domain. This could be accomplished by mapping the sensors in the target home to the sensors in the source home, mapping the sensors in the source home to sensors in the target home, or mapping both the source and target sensors to a common set of generic labels (for example, location-based mapping such as kitchen, bedroom, etc).

This mapping is just the initial step in the transfer learning. Once a shared feature-space is achieved, additional transfer learning may be necessary to resolve differences in the marginal probabilities (the residents in one home may spend half the day sleeping, while the residents in the other home only sleep 6 hours a day) or differences in the classification task (the set of activities recognized may be different). FSR focuses on achieving this initial transformation of the feature-space.

3. FEATURE SPACE REMAPPING

Traditionally, domain adaptation problems have focused on the case when $D_s \neq D_t$, usually because $P(X_s) \neq P(X_t)$. For example, in document classification, the frequency of a particular word may vary for different domains. When domain adaptation has been applied to problems where $\chi_s \neq \chi_t$ there is usually a trivial transformation between feature spaces. An example of this is found in document classification, where the domain dimensions are typically word counts in each document. To compare documents with different words, a user can set the word counts for the unseen words to zero. This allows the user to easily define a common feature space between documents. Additional transfer learning techniques may still be necessary because $P(X_s) \neq P(X_t)$ but the initial feature-space transformation is trivial. This trivial transformation works

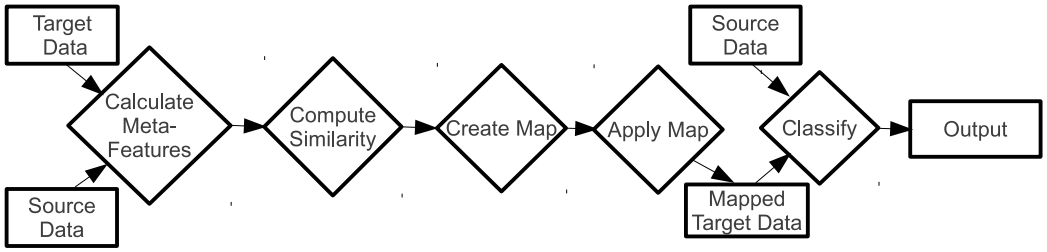


Fig. 2. Flowchart of the FSR mapping process.

because the semantic meaning of the dimensions is assumed to be known. For document classification, the known semantics of the dimensions makes it straightforward to determine the frequency for each word.

FSR, on the other hand, is a heterogeneous transfer learning algorithm and transforms feature spaces in a non-trivial manner. The semantic meaning of the dimensions is assumed to be either unknown or incompatible between the source and target domains. In the document classification example this would be equivalent to having a list of word frequencies in the document but not knowing which frequency is associated with which word or having the documents come from different languages without having any translation information. In the activity recognition domain it is equivalent to having sensor values but not knowing from which sensor (type or location) it originated. Unlike many other heterogeneous transfer learning techniques, FSR does not rely on co-occurrence data such as dictionaries, social annotations of images, or multi-view data. Additionally, we do not assume that $P(Y_s|X_s) = P(Y_t|X_t)$ or even that $Y_s = Y_t$ but we do assume that they must still be related.

In this paper, we specifically consider the case when both the source and target domains can be represented by a bag-of-features and related features have similar value distributions. This does not account for features which may be related through a linear or non-linear transformation such as $x = 10y + 3$. Differences in linear scaling can be removed through the application of normalization techniques but this may cause the FSR technique to incorrectly map features which would otherwise be clearly unrelated.

To achieve the desired feature-space transformation, we view the problem as a new machine learning task to learn a mapping from each dimension in the target feature space to a corresponding dimension in the source feature space. More formally this can be written as follows: Given source data X_s, target data X_t and a hypothesis $H_s : \mathcal{X}_s \rightarrow Y_s$ find a mapping $\theta(\mathcal{X}_t, \mathcal{X}_s)$ such that $error_\theta(H_s)$ is minimized where $error_\theta(H_s)$ represents the empirical error on the target domain by using H_s on the mapped target data. Notice the distinction between this problem definition and other approaches typically applied to heterogeneous transfer learning. Traditional heterogeneous transfer learning approaches usually map source features to target features or source and target features to a common feature space and then learn a hypothesis on this common feature space. In our approach however, we map the target features to source features and we use an already learned hypothesis to avoid the duplication of work. This also leads to the ability to combine multiple data sources through ensemble learning which will be discussed in Section 4. It is possible to relearn a new hypothesis after performing the mapping. It is also possible to apply additional transfer learning approaches after first obtaining a unified feature-space. The full FSR process is depicted in Figure 2.

The number of possible mappings between source and target feature spaces grows exponentially as the number of features increases. Even for lower dimensional data,

searching through all possible mappings quickly becomes computationally infeasible. Using feature-feature, feature-instance or instance-instance co-occurrence data could be used to guide the search but FSR operates under the assumption that this type of data is not available. When labeled data is available in the target domain, the empirical error of a classifier tested on the mapped data could provide a quantitative method for evaluating candidate mappings. It may even be possible to learn a mapping function directly. However, the search techniques would still be computationally expensive. Instead, FSR computes meta-features as means to relate source and target features. Essentially, the meta-features are used to select features in the source space which are most similar to features in the target space. These meta-features can be defined and computed multiple ways which will be discussed in Sections 3.1 and 3.2 but to simplify the presentation of the FSR algorithm let us assume that meta-features have already been calculated for the source and target features.

Next, FSR computes a similarity matrix S between source features and target features. This is done by computing a similarity score for each feature-feature pair based upon the meta-features computed for the given features. The similarity score is computed as the average similarity between the source and target meta-feature values. Formally, this score is given by Equations 1 and 2.

$$S_{xy} = \frac{1}{N} \sum_{i=1}^{N} \Omega(m_x^i, m_y^i) \tag{1}$$

where x is the xth source feature, y is the yth target feature, N is the number of meta-features and Ω is the normalized similarity between two meta-features m_x^i and m_y^i, the ith meta-feature of feature x and y respectively. We calculate the normalized similarity between two meta-features as the absolute value of the difference between meta-feature values divided by the maximum possible difference between the meta-features to obtain a normalized value between 0 and 1. This is shown in Equation 2.

$$\Omega(m_x^i, m_y^i) = 1 - \frac{|m_x^i - m_y^i|}{\max(m_x^i, m_y^i \forall x \in D_s \forall y \in D_t) - \min(m_x^i, m_y^i \forall x \in D_s \forall y \in D_t)} \tag{2}$$

If the meta-feature values are all positive, which is the case for the experiments we show here, the normalized similarity equation can be simplified to:

$$\Omega(m_x^i, m_y^i) = 1 - \frac{|m_x^i - m_y^i|}{\max(m_x^i, m_y^i \forall x \in D_s \forall y \in D_t)} \tag{3}$$

Then, FSR computes a mapping $L : y \rightarrow x$ by selecting source feature x with maximal similarity to target feature y as given by the similarity matrix S.

$$L(y) = \arg\max_{x \in D_s} S_{xy} \tag{4}$$

FSR generates a many-to-one mapping. This is because multiple dimensions (features) in the target space can be mapped to a single feature in the source space but one feature in the target domain will never map to multiple features in the source domain. FSR could alternatively be run in the opposite direction. This would result in a many-to-one mapping in the alternative direction. One of the benefits to running FSR from target to source, as will be discussed later, is the ability to easily combine multiple source domains.

Finally, FSR applies the computed mapping to the target data to be classified using the hypothesis learned on the source data. Because FSR produces a many-to-one mapping, the procedure for combining the multiple dimensions must also be defined. For dimensions with numerical values, one could use an aggregate value such as minimum, maximum, total, or average. For categorical values, one could use a voting protocol. For each instance in the target data the features are mapped to the source features. When multiple features in the target data are mapped to single feature in the source data, the feature values are combined using the specified aggregation protocol. In this work we use the summed value for activity recognition and the maximum value for document classification. We also include some experimental results when different aggregation protocols are used.

If we assume the meta-feature computation is linear, FSR has a running time of O($d_s * d_t + n + m$) where d_s and d_t is the dimensionality of the source and target data, respectively, and n and m are the number of source and target instances, respectively. This runtime is explained by the following observations. First, each dimension in the target domain is compared to each dimension in the source domain, resulting in the $d_s * d_t$ term. Second, assuming the meta-feature computation is linear in the number of data instances, then computing the meta-features requires $O(n + m)$ time. Finally, applying the mapping requires a single pass through the target data or $O(m)$ time. Despite the d^2 running time component, in our experiments FSR finishes in a reasonable amount of time even on the high dimensional text data.

As mentioned earlier, the defining and calculating of meta-features can be done in multiple ways. If some labeled target data is available, it can be used to calculate domain-independent meta-features (i.e. meta-features that can be applied to any heterogeneous transfer learning problem). We refer to this as Informed Feature Space Remapping because it requires the labeled target data. If no labeled target data is available then domain-dependent meta-features must be defined. We refer to this as Uninformed Feature Space Remapping (UFSR) because it does not require the label target data.

3.1. Informed Feature Space Remapping

Searching through all possible mappings to find the mapping which minimizes the error of the hypothesis on the target data is computationally expensive. However, since the hypothesis has been learned using the source training data one would expect the error to be minimized by selecting mappings for which the feature-label co-occurrence data is similar in the source and target datasets. This leads to our first heuristic for mapping source and target features. IFSR computes the feature-label co-occurrence data for each feature in the source and target space by calculating the expected value of the feature given the label using the labeled training data. More formally, if $Y = Y_s \cup Y_t$ then the feature-label co-occurrence data for each feature and label is computed as:

$$E(x|c) = \frac{1}{n_c} \sum_{i=1}^{n} x_i \tag{5}$$

where x is the feature, c is the label such that $c \in Y$, n_c is the number of data instances with label c, x_i is the value of feature x on the ith data instance with a label of c. This assumes a real-valued number space. One could easily extend this to categorical values by using the count of occurrences of each category as an estimation of the probability that the given feature will have the given categorical value.

Each feature-label co-occurrence value now becomes a meta-feature for the given feature. Thus $E(x|c)$ is a meta-feature for feature x and x will have $z = |Y|$ such meta-features, one for each label c. Using feature-label co-occurrence data as a meta-feature

keeps the FSR asymptotic run time within the previously stated bound of O($d_s * d_t + n + m$). This is because the meta-feature calculation is linear in the number of instances. We compute $E(x|c)$ for each label $c \in Y$. This can be done in a single pass through the datasets and thus requires $O(n + m + y)$ time. Typically $n >> y$ and $m >> y$ so this term can be simplified to just $O(n + m)$.

Additionally, using feature-label co-occurrence data for the meta-features provides domain independent meta-features so that meta-features for the specific problem do not need to be specified by a domain expert. Thus any domain for which labeled data exists can apply this feature mapping technique without setting any parameters, defining any relations, or defining any additional meta-features.

To understand why using the the feature-label co-occurrence data as a heuristic to find an approximation to the optimal mapping works we go back to the original problem definition. Given source data X_s, target data X_t and a hypothesis $H_s : \chi_s \rightarrow Y_s$ find a mapping $\theta(\chi_t, \chi_s)$ such that $error_\theta(H_s)$ is minimized. This error is minimized by maximizing the number of agreements between $H_S(\theta(q))$ and $f_t(q)$ as shown in Equation 6 where q is a data instance in X_t and $\theta(q)$ is the mapped data in the source domain space.

$$\max_\theta \sum_{q \in X_t} \begin{cases} 1, & \text{if } H_s(\theta(q)) = f_t(q). \\ 0, & \text{if } H_s(\theta(q)) \neq f_t(q). \end{cases} \tag{6}$$

A naïve Bayes classifier can learn a hypothesis by estimating $P(c)$ and $P(q_i|c)$ based upon their observed frequencies and applying Bayes rule to estimate the posterior probability $P(c|q)$. The class c with the highest posterior probability is select as the class label for q [Mitchell 1997]. Thus, if the hypothesis is expressed as a naïve Bayes classifier and if we approximate the true predictive function $f_t()$ also using a naïve Bayes formulation then Equation 6 can be expressed as shown in Equation 7.

$$\max_\theta \sum_{q \in X_t} \begin{cases} 1, & \text{if } \arg\max_{c \in Y} P(c) \prod_{i=1}^{d_t} P(\theta(q_i)|c) = \arg\max_{c \in Y} P(c) \prod_{i=1}^{d_t} P(q_i|c). \\ 0, & \text{if } \arg\max_{c \in Y} P(c) \prod_{i=1}^{d_t} P(\theta(q_i)|c) \neq \arg\max_{c \in Y} P(c) \prod_{i=1}^{d_t} P(q_i|c). \end{cases} \tag{7}$$

Under this representation, selecting the mapping for each feature that has the most similar feature-label co-occurrence value can be seen as a greedy approximation to minimize the empirical error on the mapped target data. Indeed, when the feature values are restricted to either 0 or 1, the feature-label co-occurrence value $E(x|c)$ is equivalent to the estimation of the probability that the feature has a value of 1 given the class label, $P(x = 1|c)$.

3.2. Uninformed Feature-Space Remapping

When labeled data is unavailable in the target domain we still need some way to link correlated source and target features. In this case we define meta-features which can be used as a heuristic to guide the mapping process. Meta-features should have the following attributes: 1) The meta-features should not depend on any relationship between different features. 2) Features with similar meta-feature values should also have similar conditional probability distributions. The first stipulation allows meta-features to be applied, calculated and compared between different feature spaces.

To clarify this concept consider the following examples. In activity recognition using motion sensors, the time of day when motion sensor A fires would be an acceptable meta-feature. On the other hand, the amount of time between sensor A firing and sensor B firing would not be an acceptable meta-feature because it depends on the relationship between sensor A and sensor B. However, the amount of time between

sensor A firing and any unspecified sensor firing is acceptable because it again depends only upon sensor A.

The second stipulation is important because it provides a basis for using the meta-features as a heuristic to select a mapping between features. The meta-features provide some indication that the features have similar conditional probability distributions and if the conditional probability distributions of two features are similar then the mapping process will be more likely to select that pair for mapping.

Defining meta-features and creating the feature dataset is a domain specific task. The meta-features used for activity recognition may not be applicable to the document classification domain. In Table I we describe the meta-features we use for the activity recognition problem and using the example data shown in Table II we show the meta-feature values for some of the sensors. These meta-features have been chosen to be consistent with the previously discussed meta-feature stipulations.

Table I. Meta-features defined for activity recognition.

Meta-feature Description	**Meta-Feature**	**M021**	**MA020**	**M018**
average sensor event frequency over 1 hour time periods (x24)	03:00	3	1	0
	04:00	2	0	0
	05:00	2	0	0
	06:00	4	0	0
	07:00	0	0	0
	08:00	14	14	2
average sensor event frequency over 3 hour time periods (x8)	03:00	7	1	0
	06:00	18	14	2
average sensor event frequency over 8 hour time periods (x3)	00:00	11	1	0
	08:00	14	14	2
average sensor event frequency over 24 hour time periods (x1)	00:00	25	15	2
average and standard deviation of the time of day of this sensor event (seconds)	avg.	26015.36	30356.40	31594.5
	std. dev.	6831.72	4555.85	2.50
average and standard deviation of the time between this sensor event and the previous sensor event (seconds)	avg.	760.79	1.34	1.31
	std. dev.	1862.31	1.02	0.52
average and standard deviation of the time between this sensor event and the next sensor event (seconds)	avg.	722.94	13.66	5.85
	std. dev.	1833.35	44.59	2.11
average and standard deviation of the time between this event and the next event from this sensor (seconds)	avg.	761.41	1305.67	4.53
	std. dev.	1862.06	4699.58	0.0
probability the next sensor event is from the same sensor	prob.	0.76	0.72	0.0

All of these meta-features can be computed in linear time therefore the asymptotic run time of O($d_s * d_t + n + m$) is still achieved.

As an extension, if labeled target data is available, one could easily combine the domain-dependent meta-features with the feature-label co-occurrence meta-features to provide additional information when selecting a feature-space mapping. One could also compute the features on a per-class basis. For example, the frequency of a sensor event could instead be computed as the frequency of a sensor event given the activity label. However, in order to avoid overfitting the data, this may require more labeled data than is typically available in transfer learning scenarios.

4. COMBINING MULTIPLE DATA-SOURCES

One of the major benefits of the FSR mapping approach is that it can be used to combine data from multiple source domains in a straightforward manner. An ensemble

Table II. Sample of Sensor Events

Date	Time	Sensor	Value
2011-06-15	03:41:50.30088	M021	OFF
2011-06-15	03:41:50.402649	MA020	OFF
2011-06-15	03:44:50.862962	M021	ON
2011-06-15	03:44:51.929508	M021	OFF
2011-06-15	04:41:28.179357	M021	ON
2011-06-15	04:41:29.333803	M021	OFF
2011-06-15	05:33:44.024833	M021	ON
2011-06-15	05:33:45.118382	M021	OFF
2011-06-15	06:33:30.363675	M021	ON
2011-06-15	06:33:31.437863	M021	OFF
2011-06-15	06:33:33.878588	M021	ON
2011-06-15	06:33:35.956492	M021	OFF
2011-06-15	08:45:45.685723	M021	ON
2011-06-15	08:45:46.789252	M021	OFF
2011-06-15	08:45:47.675812	M021	ON
2011-06-15	08:45:49.382375	M021	OFF
2011-06-15	08:45:50.869002	M021	ON
2011-06-15	08:45:53.115439	M021	OFF
2011-06-15	08:45:55.016408	M021	ON
2011-06-15	08:45:56.17543	M021	OFF
2011-06-15	08:46:00.115612	M021	ON
2011-06-15	08:46:00.665277	MA020	ON
2011-06-15	08:46:01.219705	M021	OFF
2011-06-15	08:46:01.787696	MA020	OFF
2011-06-15	08:46:03.646237	M021	ON
2011-06-15	08:46:03.817155	MA020	ON
2011-06-15	08:46:08.513192	M021	OFF
2011-06-15	08:46:08.712314	MA020	OFF
2011-06-15	08:46:09.87972	MA020	ON
2011-06-15	08:46:12.103082	MA020	OFF
2011-06-15	08:46:13.763861	MA020	ON
2011-06-15	08:46:14.876471	MA020	OFF
2011-06-15	08:46:17.157816	MA020	ON
2011-06-15	08:46:18.296485	MA020	OFF
2011-06-15	08:46:21.859339	MA020	ON
2011-06-15	08:46:22.752142	M021	ON
2011-06-15	08:46:23.885996	M021	OFF
2011-06-15	08:46:25.199775	MA020	OFF
2011-06-15	08:46:26.713111	MA020	ON
2011-06-15	08:46:27.590115	M019	ON
2011-06-15	08:46:29.876241	MA020	OFF
2011-06-15	08:46:30.760636	M019	OFF
2011-06-15	08:46:32.587806	M018	ON
2011-06-15	08:46:36.329587	MA013	ON
2011-06-15	08:46:37.117772	M018	OFF
2011-06-15	08:46:45.86861	MA013	OFF

classifier can be built by mapping the target domain to each source domain and training a separate base classifier for each source domain. The output from these source classifiers can then be combined by the ensemble meta-classifier to make the final prediction. We refer to this as Ensemble Learning via Feature-Space Remapping (ELFSR).

Ensemble methods have been used in a variety of situations with great success. According to Hansen and Salamon, a necessary and sufficient condition for ensemble classifiers to be more accurate than any of the individual classifiers are for the classifiers to be accurate and diverse [Hansen and Salamon 1990]. An *accurate* classifier is one which has a classification accuracy better than random guessing [Dietterich 2000]. Two classifiers are diverse if the errors they make are different (and preferably uncorrelated) [Dietterich 2000]. Most ensemble techniques defined to date generate a set of diverse classifiers. Bagging, for example, generates classifiers by repeatedly subsampling the original data with replacement [Breiman 1996]. Boosting iteratively reweights samples based on the accuracy of the previous iteration [Freund and Schapire 1997]. In ELFSR, each classifier is drawn from a different domain, leading to a naturally diverse set of classifiers.

Once the classifiers are generated, the output must be combined to obtain the final result. Several approaches have been used including majority voting, weighted voting, summing the probabilities, and training a new learner on the output of the classifiers or *stacking* [Wolpert 1992]. Stacking is a supervised technique and thus requires additional labeled data to train the ensemble classifier. This means that stacking can be readily combined with IFSR, which already uses labeled data.

Work on ensemble classifiers for transfer learning has mainly focused on boosting techniques [Pan et al. 2012; Xian-ming and Shao-zi 2009; Yao and Doretto 2010]. As

there has been very little work on transfer learning using voting or stacking ensemble classifiers, we compare the results of several different ensemble configurations. Specifically, we consider two voting ensembles (a majority voting ensemble and a summation voting ensemble), and two stacking ensembles (via naïve Bayes and via a decision tree). The voting ensembles have the advantage of not requiring any labeled data in the target domain, while the stacking techniques require a small amount of labeled data.

4.1. Voting Ensemble

One of the simplest methods for combining multiple classifiers is through majority voting. Each classifier votes for the class label it predicts for the given instance and the label receiving the most votes wins.

The drawback to the majority voting ensemble classifier is that the ensemble throws away important information by only considering the most likely label as predicted by each classifier. The summation voting ensemble classifier rectifies this weakness by summing up the predicted probability of each label for each classifier and then assigning the label with the highest summed probability.

4.2. Stacking

In stacking, the output of each source classifier is fed into the ensemble classifier which then produces the final classification. Here we consider two different classification algorithms for the ensemble classifier, naïve Bayes and decision trees. One of the drawbacks to using stacking is the requirement of labeled data to train the ensemble classifier. Rather than test both FSR and IFSR with the stacking technique we only consider the result of using IFSR since IFSR already uses a small amount of labeled data in the target domain. We use stacking with IFSR without requiring any additional labeled data in the target domain.

5. EXPERIMENTAL RESULTS

FSR and its proposed extensions can be applied to a variety of different transfer learning problems. We evaluate the performance of these techniques in both the activity recognition domain and in the document classification domain.

First, we evaluate the performance of UFSR, IFSR, and ELFSR on 18 datasets from different smart apartments. Specific statistics for each dataset are found in Table III. Each apartment is equipped with motion sensors and door sensors. The number of sensors range from 17 to 39 with an average of 28.7 sensors and a standard deviation of 6.21. Each dataset has been annotated with 37 different activities, shown in Table IV, with the total amount of labeled data spanning one month of time per dataset. We consider all possible combinations of source and target datasets, yielding a total of 306 possible pairings.

We also test IFSR on the newsgroups dataset [Lang et al. 1995]. The newsgroups dataset is a collection of approximately 20,000 documents across 20 different topics. The topics are organized in a hierarchical manner. Following the processing steps used by Dai et al. [Dai et al. 2007] and Pan et al. [Pan et al. 2011], the source and target datasets are created by first selecting two top-level categories as the class labels. The documents are then split by sub-categories to form a source dataset and a target dataset. The resulting datasets are shown in Table V. We also show basic statistics about the datasets in Table VI.

Each pair of datasets is processed separately so that the alignment and number of attributes is the same for datasets in the same row but different for datasets in different rows (i.e. the feature-space of D_s sci. vs. talk is the same as the feature space of D_t sci. vs. talk but the feature-space of D_s rec. vs. sci. is not the same as the feature

Table III. Summary statistics of the activity recognition dataset

Id	# Features	# Labels	# Instances	# UFSR Meta-Features	# IFSR Meta-Features
1	35	29	133157	1575	1295
2	17	26	53669	765	629
3	37	31	178137	1665	1369
4	29	29	57918	1305	1073
5	39	32	141181	1755	1443
6	26	32	149391	1170	962
7	26	30	183945	1170	962
8	26	28	98768	1170	962
9	34	30	102466	1530	1258
10	24	30	143145	1080	888
11	38	30	157736	1710	1406
12	24	29	135451	1080	888
13	32	32	116641	1440	1184
14	26	31	195611	1170	962
15	23	29	100255	1035	851
16	33	32	179693	1485	1221
17	23	29	92740	1035	851
18	24	30	117067	1080	888

Table IV. List of activities and the relative frequency of occurrence of each activity

Activity	Frequency	Activity	Frequency
Enter Home	0.0031	Personal Hygiene	0.0545
Eat Lunch	0.0070	Leave Home	0.0026
Cook Dinner	0.0534	Eat Dinner	0.0100
Exercise	0.0002	Cook Lunch	0.0274
Wash Dinner Dishes	0.0127	Relax	0.0191
Read	0.0103	Wash Lunch Dishes	0.0077
Phone	0.0029	Evening Meds	0.0037
Eat Breakfast	0.0101	Watch TV	0.0405
Cook	0.0348	Wash Breakfast Dishes	0.0126
Eat	0.0066	Groom	0.0087
Housekeeping	0.0113	Toilet	0.0434
Wash Dishes	0.0088	Work At Desk	0.0004
Sleep Out Of Bed	0.0034	Work At Table	0.0253
Morning Meds	0.0053	Cook Breakfast	0.0320
Take Medicine	0.0036	Bed Toilet Transition	0.0156
Bathe	0.0175	Work	0.0329
Other Activity	0.2789	Entertain Guests	0.0837
Sleep	0.0407	Work On Computer	0.0498
Dress	0.0194		

space of D_t sci. vs. talk. Additionally, the source distribution is different from the target distribution for all datasets because the documents come from different sub-categories; however, they are still related because they come from the same top-level categories.

As in the work of Dai et al. [Dai et al. 2007] and Pan et al. [Pan et al. 2011] we train IFSR on D_s and then test IFSR on D_t for each row in the table. However, we also take the transfer learning problem one step further and test each D_t on classifiers trained on the D_s of the other rows. This means that in addition to $P(X_s) \neq P(X_t)$ because the source and target data comes from different sub-domains, now $\chi_s \neq \chi_t$ because the source and target data come from different top-level domains. In this new problem we know longer no which words are the same in the different domains (i.e. "bit" may

Table V. Breakdown of the 20 newsgroups dataset for transfer learning

Dataset	$\mathbf{D_s}$	$\mathbf{D_t}$
comp vs. sci	comp.graphics comp.os.ms.windows.misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sce.space
comp vs. talk	comp.graphics comp.sys.mac.hardware comp.windows.x talk.politics.mideast talk.religion.misc	comp.os.ms.windows.misc comp.sys.ibm.pc.hardware talk.politics.guns talk.politics.misc
rec vs. sci	rec.autos rec.sport.baseball sci.med sci.space	rec.motorcycles rec.sport.hockey sci.crypt sci.electronics
rec vs. talk	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
sci vs. talk	sci.electronics sci.med talk.politics.misc talk.religion.misc	sci.crypt sci.space talk.politics.guns talk.politics.mideast

Table VI. Summary statistics of the newsgroups datasets

Id	# Features	# + Instances	# - Instances	# Meta-Features
$D_s(cs)$	9892	1958	1972	19784
$D_t(cs)$	9892	2923	1977	19784
$D_s(ct)$	10624	2914	1568	21248
$D_t(ct)$	10624	1967	1685	21248
$D_s(rs)$	14974	1984	1977	29948
$D_t(rs)$	14974	1993	1972	29948
$D_s(rt)$	15253	1984	1685	30506
$D_t(rt)$	15253	1993	1568	30506
$D_s(st)$	15327	1971	1403	30654
$D_t(st)$	15327	1978	1850	30654

be the ith word in the source domain but we have no idea which index corresponds to "bit" in the target domain or even if the word "bit" is found in the target domain, let alone if it has the same semantic meaning in both domains. This also means that although technically $Y_s = Y_t$ because we use (0,1) for the class labels, semantically $Y_s \neq Y_t$ because the source task may be to classify documents as either belonging to recreation or science while the target task may be to classify documents as belonging either to talk or computers.

5.1. Smarthome

We compare *UFSR* and *IFSR* against several other baselines. *UFSR* uses the meta-features described in Table I. *IFSR* uses the feature-label co-occurrence meta-features as described in Equation 5. The first baseline, *Manual*, uses the generalized sensor locations (kitchen, bedroom, etc) to map sensors from one apartment to another. Lastly, the *None* classifier treats all sensor events as coming from a single source. Essentially this eliminates the sensor dimension and only considers the time of day and day of week of the activity. The *Manual* technique is the mapping technique currently used

by most researchers in activity recognition [Cook et al. 2012; Rashidi and Cook 2011; van Kasteren et al. 2008]. It does not require any labeled data in the target domain, but it does require the manual definition of sensor locations. On the other hand, *None* provides a lower bound on the expected performance. All of the techniques use a naïve Bayes classifier trained on the source domain and tested on the target domain. We also include *IFSR-DT* which uses a decision tree trained on source domain and tested on the target domain.

Performance is measured using both the accuracy (given by Equation 8) and the unweighted average recall (see Equation 9). In both of these equations N is the total number of instances, K is the number of labels, and A is the confusion matrix where A_{ij} is the number of instances of class i classified as class j.

$$Acc = \frac{1}{N}\sum_{i=1}^{K} A_{ii} \tag{8}$$

$$Recall = \frac{1}{K}\sum_{i=1}^{K} \frac{A_{ii}}{\sum_{j=1}^{K} A_{ij}} \tag{9}$$

We report both the accuracy and the recall because accuracy scores are biased towards the majority class. For balanced class distributions this has little effect on the metric, but it may not be suitable for unbalanced class distributions. Using the unweighted average recall eliminates this bias and treats all classes equally [van Kasteren et al. 2008]. Note that accuracy can also be considered as the average recall weighted by the number of instances in the class. A one-way ANOVA is performed and the resulting p-value is less than .0001. The 95% confidence interval is depicted with the error bars.

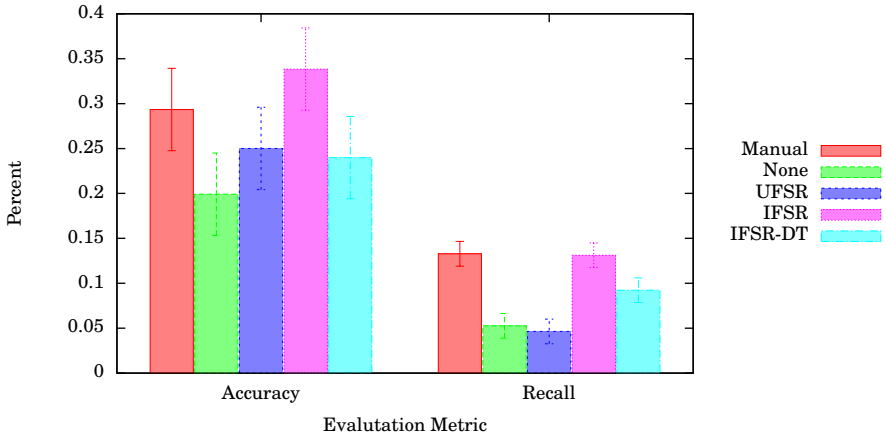


Fig. 3. Classification accuracy on the target domain using a single source domain. *Manual*, and *None* both provide baseline comparisons. *Manual* is the mapping specified by a domain expert. *None* does not apply any mapping at all. Having *UFSR* and *IFSR* come close or surpass the *Manual* mapping results is excellent.

As can be seen in Figure 3, the *UFSR* algorithm performs well if only the accuracy score is considered. Its performance nearly matches that of the manual technique. However, when the recall score is considered, *UFSR* performance drops significantly.

IFSR, on the other hand, performs well under both metrics. The accuracy is better than that of the *Manual* technique and the recall is just slightly worse than the *Manual* technique. Matching the performance of the *Manual* mapped technique is a positive result as it implies that transfer learning can be used to reduce or eliminate the need for a domain expert to supply a mapping between domains. The *IFSR* mapping is able to outperform the *Manual* mapping technique because the manual mapping technique is based solely upon the location of the sensors. This is effective when the resident in the source dataset performs activities in the same locations as the resident in the target dataset. For example, both residents are likely to cook in the kitchen. On the other hand, the manual mapping technique is likely to fail when the residents perform the same activity in different locations. For example, the resident in the source dataset might eat in the living room while the resident in the target dataset might eat in the kitchen. *IFSR* overcomes this problem by mapping features based on correlation with the activity label. The meta-features used by *IFSR* are specifically derived to optimize the mapping when a naïve bayes classifier is used. However, from the performance of *IFSR-DT* we see that the mapping works with other classifiers as well. Exploring other mapping strategies and heuristics may lead to further improvements for specific types of classifiers.

The previously-discussed results are the average of 306 different mappings. Individual results show both higher and lower performance. One direction of transfer learning research focuses on how to select the best source dataset. Assuming this problem is solved then we could select the "best" source dataset for each target dataset. We do not claim that this contributes to avoid negative transfer, only that if negative transfer can be predicted and avoided we can improve the results. Figure 4 shows the results of using the best source dataset with the same mapping techniques discussed earlier. Under this scenario, the performance of *IFSR* improves significantly. *IFSR* still outperforms the *Manual* mapping technique by a small margin although there is a high degree of overlap between the two confidence intervals. Again, a one-way ANOVA is performed and the resulting p-value is less than .0001. The 95% confidence interval is depicted with the error bars.

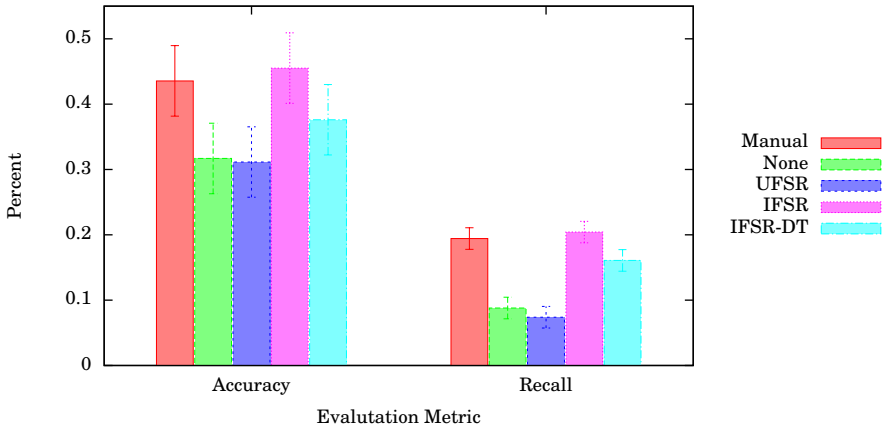


Fig. 4. Classification accuracy on the target domain using the best single source domain. This assumes that the best dataset to transfer from could be identified a priori. *Manual*, and *None* both provide baseline comparisons. *Manual* is the mapping specified by a domain expert. *None* does not apply any mapping at all.

The next experiment shows the effect of the amount of labeled target data on the accuracy and recall score of the *IFSR* algorithm. As in the previous experiments, we use the 306 possible pairings of the activity recognition datasets. However, this time we vary the number of days of labeled target data from .25 to 30. We also include a comparison to a baseline classifier, *Self*, which uses a naïve Bayes classifier which has been trained only on the labeled target data and is tested on the remaining target data. Figure 5 shows the results. Clearly, adding more labeled target data is initially beneficial. However, for *IFSR*, the increase in accuracy begins to level off after approximately ten days of labeled target data. The increase in recall appears to peek between five and ten days of labeled target data after which point the recall score declines slightly. This may indicate that having too much labeled data causes *IFSR* to overfit the data. Comparing *IFSR* against the baseline *Self* we see that initially *IFSR* is able to outperform the baseline. As the amount of labeled data exceeds one day though, *Self* begins to outperform *IFSR*.

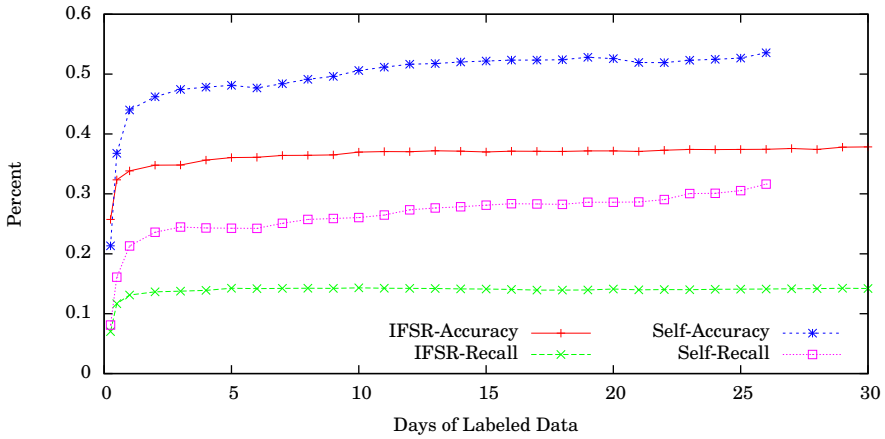


Fig. 5. *IFSR* and *Self* accuracy and recall scores as the amount of labeled target data increases. Accuracy continues to show improvement with the increase of labeled target data while the recall score appears to peek with between five and ten days of labeled data in the target domain

Next, we consider different techniques which utilize data from multiple source datasets. *Self* uses a naïve Bayes classifier which has been trained on the full amount of labeled target data using 3-fold cross-validation. *Combined* combines all of the source domain data into one big dataset with sensor mappings being manually defined by location. A naïve Bayes classifier is trained on all of the source data and then tested on the target data. The ensemble techniques each train one naïve Bayes classifier per source dataset and the ensemble is then tested on the target domain. As in the previous experiments only one day of labeled target data is used by IFSR to make the mapping. Figure 6 shows the results using the voting ensemble techniques while Figure 7 shows the results using the stacking ensemble techniques. In neither case do we attempt to select the best source datasets we simply use all available source dataset. In the next experiment we will consider the effect choosing random subsets of datasets has on the overall results.

Again we use the accuracy and unweighted average recall for performance metrics. The performance of the voting ensembles is mixed. *UFSR* is still unable to compete with the techniques which use more information (labeled data or manual mappings). The *IFSR* voting ensembles perform comparably to the combined dataset. The tradeoff

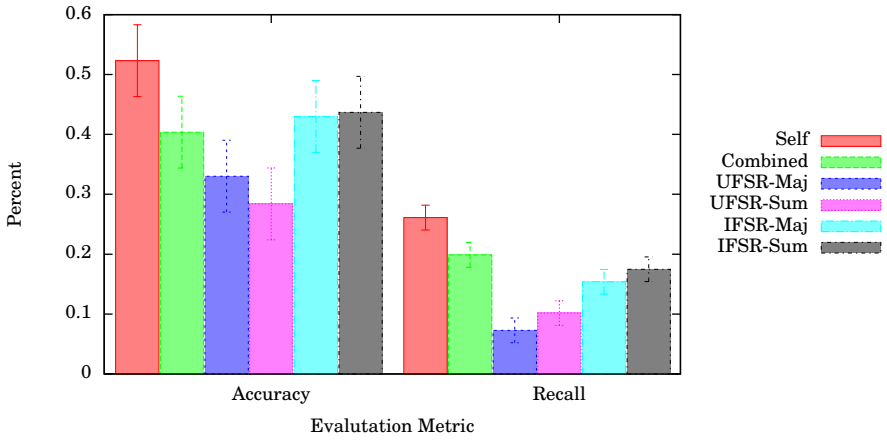


Fig. 6. Classification accuracy on the target domain using multiple source domains with a voting ensemble. *Self* and *Combined* provide baseline comparisons. *Self* is the result when the source and target dataset are the same and uses the all the labeled target data, while *Combined* uses the mappings provided by a domain expert to build a generic classifier. Matching the performance of *Combined* is a positive result.

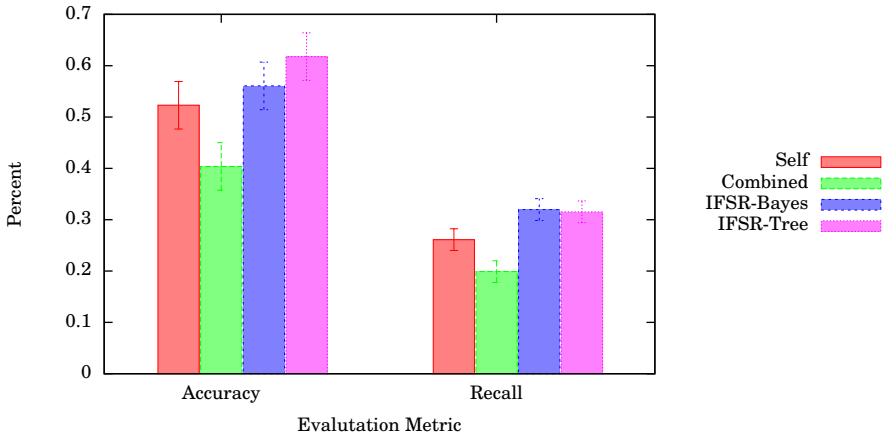


Fig. 7. Classification accuracy on the target domain using multiple source domains with stacking ensembles. *Self* and *Combined* provide baseline comparisons. *Self* is the result when the source and target dataset are the same and uses the all the labeled target data, while *Combined* uses the mappings provided by a domain expert to build a generic classifier. The performance of *IFSR-Bayes* and *IFSR-Tree* both manage to beat these baselines representing a considerable gain for the transfer learning techniques.

is that the combined dataset requires a manually-mapped specification while the *IFSR* voting ensembles require a small amount of labeled data in the target domain.

The performance of the stacking ensembles stand out above the rest. Both stacking ensembles achieve higher performance in terms of the accuracy and recall scores than the combined dataset or the *Self* classifier. It does this using only a single day's worth of labeled data and no manual mapping is required. The *Self* approach uses nearly 30 days of labeled data and is trained and tested on the same dataset (with cross-validation), while the Combined approach uses no labeled data in the target domain but requires a manual mapping to be specified.

In addition to comparing the performance of *IFSR* and *ELFSR* against other techniques we also consider how the number of source datasets affects the performance achieved by the techniques.

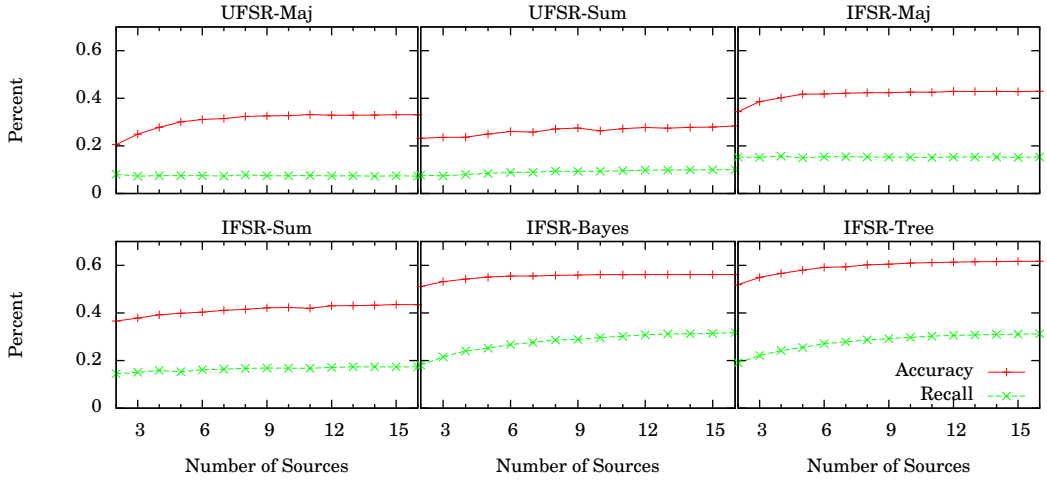


Fig. 8. Learning curve for the ensemble classifiers where the number of source classifiers ranges from 2 to 16. Each ensemble technique quickly improves with more source classifiers but the performance improvements then begin to level off.

Figure 8 shows the learning curve for each ensemble technique as the number of source datasets increases. For *UFSR-Summation, IFSR-Summation, IFSR-Bayes*, and *IFSR-Tree*, the performance increases with an increasing number of datasets. Most of the improvement is achieved within the first seven datasets, after which performance improvement tapers off. For *UFSR-Majority* and *IFSR-Majority*, the accuracy performance improves with an increasing number of datasets, but the recall performance remains almost constant regardless of the number of datasets. This illustrates the fact that important distinguishing information is being discarded by the majority voting scheme.

5.2. Newsgroups

For the Newsgroups dataset, we compare the *IFSR* technique using 10% of the labeled data in D_t to perform the mapping against several baselines. *Self* uses a naïve Bayes classifier which has been trained and tested on the target dataset using 10-fold cross-validation. *None* uses a naïve Bayes classifier which has been trained on the source dataset and tested on the target dataset. The source and target feature spaces are adjusted to have the same number of features by adding zero-valued features as necessary. No attempt is made to adjust for the domain differences. This is similar to applying a random mapping between the feature spaces. *TCA* is a domain adaptation technique which projects both the source and the target domain onto a shared subspace of reduced dimensionality [Pan et al. 2011]. We compare our results against the unsupervised TCA using a linear kernel with 30 dimensions. We also compare against the semi-supervised TCA (SSTCA) using a linear kernel with 30 dimensions. Since the class distribution is balanced in these datasets we report only the accuracy scores. Figure 9 shows the results. Error bars are shown at the 95% confidence level.

From these results it is clear the *IFSR* is not the best technique for transfer learning when $\mathcal{X}_s = \mathcal{X}_t$ and $P(X_s) \neq P(X_t)$. This is not surprising since IFSR is designed mainly to handle different feature spaces. The performance results of *IFSR* are low on the first

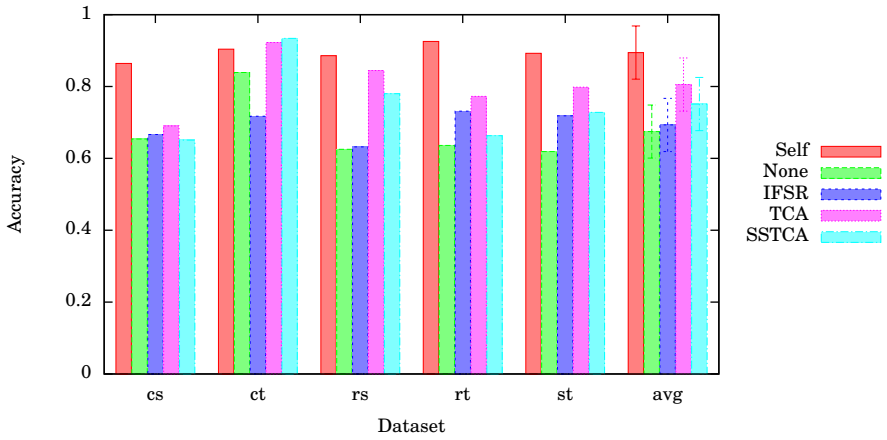


Fig. 9. Newsgroups dataset results with $P(X_s) \neq P(X_t)$. IFSR is not the best choice in this situation but is usually better than not performing any type of transfer. *Self* is the result when the source and target dataset are the same (and uses all the labeled target data).

three datasets (cs, ct, and rs), with *IFSR* only slightly outperforming *None* on cs and rs, and actually performing worse than *None* on ct. *TCA* and *SSTCA* also struggles to improve performance on the cs dataset but do well on the ct and rs datasets. These results show the importance of further research into detecting and avoiding negative transfer. *IFSR* performs much better on the last two datasets (rt and st), improving the classification accuracy by approximately 10% as compared to *None*. The accuracy of *IFSR* is still lower than *TCA*, but the gap is much narrower. On two of the datasets (cs and rt) *IFSR* even performs better than *SSTCA*. Note that *IFSR* is a technique which has been designed to specifically handle the case when $\mathcal{X}_s \neq \mathcal{X}_t$, while *TCA* is designed to handle the case when $P(X_s) \neq P(X_t)$. In this experiment, $\mathcal{X}_s = \mathcal{X}_t$ but $P(X_s) \neq P(X_t)$. When viewed in this light, the results of the two algorithms are not surprising. Of interest is that *IFSR* is able to show some improvement in many cases even when $\mathcal{X}_s = \mathcal{X}_t$ and $P(X_s) \neq P(X_t)$.

The second interesting thing to note is that when *IFSR* performs well, *TCA* tends to do worse (compare ct and rs to rt and st). One possible explanation for this might be that the differences between $P(X_s)$ and $P(X_t)$ are greater in rt and st. As the differences increase, the problem begins to more closely resemble the case when $\mathcal{X}_s \neq \mathcal{X}_t$. The negative correlation between *IFSR* and *TCA* is not manifested on the results for the cs dataset. The reason this occurs is unclear but it may be related to the fact that the performance of *Self* is lowest for the cs dataset, possibly indicating that the dataset is harder to learn than the other datasets. Further research is needed to investigate these ideas.

The second experiment we conduct using the newsgroups dataset involves transferring knowledge between datasets where $\mathcal{X}_s \neq \mathcal{X}_t$. This is a significant step away from the previous experiment where, $\mathcal{X}_s = \mathcal{X}_t$ and $P(X_s) \neq P(X_t)$. It is also the first time this type of problem has been considered for document classification when no translation oracle is available. Since this is the first time such a problem has been tried, we cannot directly compare against any previous results. We report the results for each target dataset, averaged over all five source datasets, and the best results for each target dataset. In this experiment, the only baseline we have to compare against is the performance when no transfer is performed (*None*). The results are shown in Figure 10. Not surprisingly, the accuracy of None is close to random guessing, ranging from

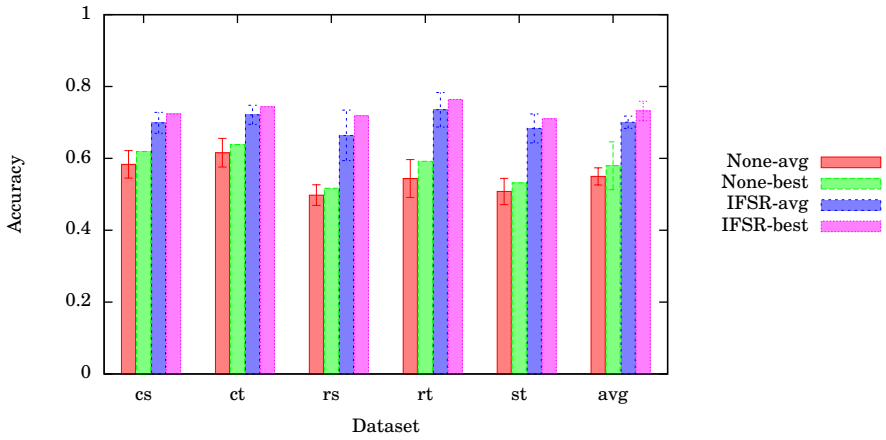


Fig. 10. Newsgroups dataset results with heterogeneous transfer and no translation oracle. IFSR clearly outperforms the baseline technique where features-spaces are not aligned.

50-60%. The exciting result is that the accuracy of *IFSR* is much better, achieving as high as 73% accuracy when averaged over the source datasets and 76% accuracy for the best dataset. A two-tailed paired t-test gives a p-value of .00005 over all the datasets, and p-values between .01 and .002 for the individual datasets.

We emphasize that this transfer problem reflects differences along three of the four possible transfer variables. Specifically, $\chi_s \neq \chi_t$, $P(X_s) \neq P(X_t)$, and $f_t() \neq f_s()$. Additionally, although $Y_s = Y_t$, as we are using 0 and 1 for class labels, semantically the 0 and 1 represent different labels in the different datasets. We have successfully trained a classifier to recognize documents as belonging to the categories of "recreation" or "talk" and used the learned model to classify documents as belonging to either "computers" or "science".

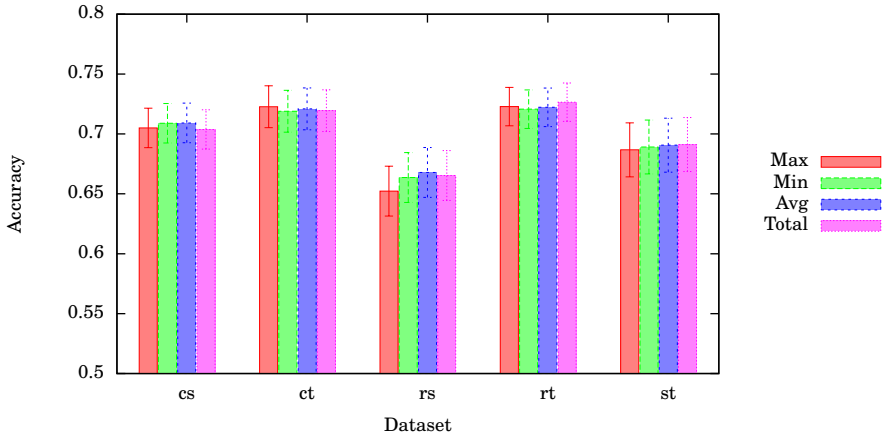


Fig. 11. Newsgroups dataset results comparing different aggregation techniques. The process of aggregating target features which mapped to the same source feature has little effect on the overall performance of IFSR.

The third experiment we conduct, using the newsgroups dataset, shows the effect of choosing a particular aggregation method for mapping multiple dimensions in the target domain to a single dimension in the source domain. Specifically, we compare IFSR using the following aggregation techniques: Maximum, Minimum (greater than 0), Average and Total. The results are shown in Figure 11. Surprisingly, the aggregation method has little effect on the overall accuracy of the technique as applied to the newsgroups datasets. Running an ANOVA on the results yields a p-value of .95 indicating that the results are not statistically significant.

The reason the aggregation technique has little effect on the accuracy results is not clear. However, we can rule out the explanation that there just is not much aggregation to be done. A quick look at the generated mappings shows that hundreds of attributes in the target domain map to tens of attributes in the source domain and many more attributes in the source domain have two or more attributes mapping to them from the target domain. Thus there is indeed a large amount of aggregation occurring. We can think of a few other possible explanations, the features being aggregating may be of little importance in defining class boundaries or the features being aggregated may have similar enough values that any of the aggregation techniques work equally well. We plan to investigate these ideas in future work.

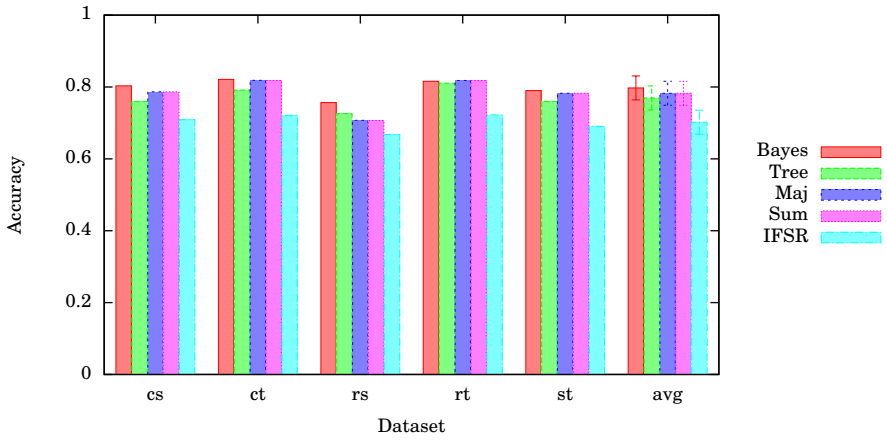


Fig. 12. Newsgroups dataset results comparing ensemble techniques. No single ensemble technique is clearly better than any other ensemble technique. However, all of the ensemble techniques perform better than when only a singe source domain is employed.

We also evaluate the performance of the ELFSR techniques on the newsgroups datasets. For each newsgroup target dataset D_t we use all the other newsgroup datasets as source datasets. This gives us a total of nine source datasets for each target dataset. We consider both voting ensembles and stacking ensembles. The results are shown in Figure 12. *Bayes* is a stacking ensemble using Naive Bayes as the ensemble classifier, *Tree* is a stacking ensemble using a Decision Tree as the ensemble classifier, *Maj* is a majority voting ensemble, *Sum* is a sum of probabilities voting ensemble, and *IFSR* is the average result without using an ensemble learner. All of the ensemble techniques evaluated show better results than the basic *IFSR* technique. Applying a one-way ANOVA to the results yields a p-value of .003 indicating that the difference in means are statistically significant. Unlike in the activity recognition domain, here we do not see as much difference between the ensemble techniques themselves as each technique performs similarly to the others. The Naive Bayes stacking ensemble

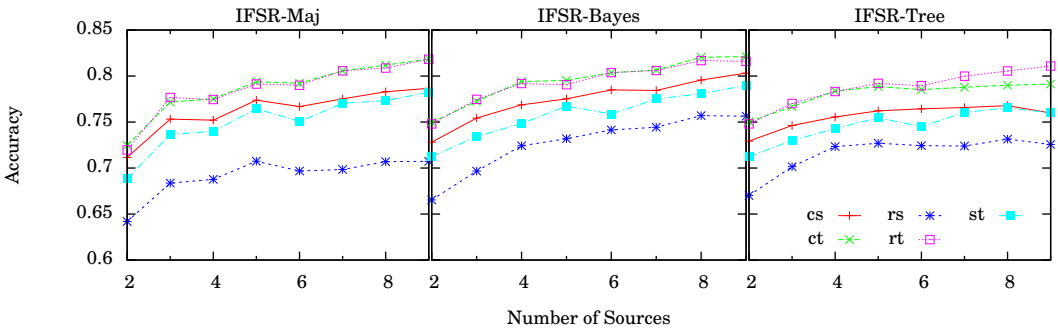


Fig. 13. Learning curve for the ensemble classifiers. As more source classifiers are included the accuracy continues to improve.

has the highest accuracy scores but the other techniques are within a few percentage points. As can be seen by the confidence intervals, the means for each technique show significant overlap with each other except for the baseline technique.

In addition to comparing the performance of IFSR and ELFSR against other techniques we also consider how the number of source datasets affects the performance achieved by the techniques.

We generate learning curves for the newsgroups dataset as shown in Figure 13. As the number of source classifiers increase, so does the overall accuracy. This performance increase occurs most rapidly with the inclusion of the first few classifiers and then slowly tapers off as more source classifiers are added. This same pattern is observed with the activity recognition datasets, but due to space constraints the results are not shown here.

One interesting pattern that emerges in majority voting learning curve is the effect of odd and even number of source datasets. Each time the number of source datasets increases from odd to even there is essentially no improvement. However, each time the number of source datasets increases from even to odd there is a corresponding jump in the resulting accuracy. This makes sense intuitively because with an even number of sources, ties are broken arbitrarily (leading to an average accuracy of 50% for the tied cases). When a new classifier is added it acts as the tie-breaking vote. Since the accuracy of the classifier is greater than 50% we would expect the performance to increase, which it does.

6. CONCLUSION

In this paper we present a novel heterogeneous transfer learning technique, Feature-Space Remapping, which transfers knowledge between domains with different feature spaces without using typical co-occurrence data. The datasets we tested on also had different marginal probability distributions on the domains, and different conditional probabilities. This makes the difference between source and target datasets greater than many previously attempted transfer learning problems. We present both informed and uniformed variations based upon the availability of labeled data in the target domain. The Informed Feature-Space Remapping uses labeled data in the target domain to allow the mapping to occur in a domain independent fashion and can be applied to any heterogeneous transfer learning problem where labeled target data exists without requiring any additional parameters or user-configurations. The FSR techniques are compatible with most other transfer learning techniques and could be applied as a pre-processing step to obtain a common feature space before applying traditional domain adaptation techniques.

Ensemble Learning via Feature-Space Remapping is introduced to combine multiple source datasets and achieve even greater classification accuracy. Using ELFSR we are able to outperform a classifier trained and tested only in the target domain for the activity recognition problem. The results of the ELFSR technique in the document classification domain are promising as well achieving accuracies of 80% or more.

We also take the document classification problem and extend it to a new realm by removing the translation oracle or other co-occurrence data. To our knowledge, this is the first attempt to solve such a problem. While there is still room for improvement, the results are promising, achieving as high as 76% accuracy despite the great differences between source and target datasets. When the ensemble techniques are applied the accuracy increases to as high as 82%.

There are still many open research questions to pursue, including avoiding negative transfer effects and identifying the best sources for transfer. We have shown positive results for FSR when the dataset is feature-rich. The activity recognition problem has a large number of classes but relatively few features, while the document classification problem has a large number of features but only two possible classes. In the future we plan to explore additional mapping techniques that work when the data is more sparse, having both a large number of features and a large number of classes. An additional future direction involves the combining of multiple dimensions. FSR generates a many-to-one mapping of target dimensions to source dimensions. We suggest exploring additional ways of combining multiple dimensions as well as exploring enforcing a one-to-one mapping or running the mapping in the other direction to generate a one-to-many mapping. While there is still much research to be done, FSR is a promising new technique to improve the transfer of knowledge between domains which will in turn lead to more robust learning systems.

REFERENCES

A. Arnold, R. Nallapati, and W.W. Cohen. 2007. A Comparative Study of Methods for Transductive Transfer Learning. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. 77 –82. DOI:http://dx.doi.org/10.1109/ICDMW.2007.109

S.M. Barnett and S.J. Ceci. 2002. When and where do we apply what we learn?: A taxonomy for far transfer. *Psychological bulletin* 128, 4 (2002), 612–637.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistic*.

John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. 120–128.

Leo Breiman. 1996. Bagging predictors. *Machine Learning* 24 (1996), 123–140. DOI:http://dx.doi.org/10.1007/BF00058655

J.P. Byrnes. 1996. *Cognitive development and learning in instructional contexts*. Allyn and Bacon, Boston.

R. Chattopadhyay, N.C. Krishnan, and S. Panchanathan. 2011. Topology Preserving Domain Adaptation for Addressing Subject Based Variability in SEMG Signal. In *2011 AAAI Spring Symposium Series*. http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/download/2395/2907

Diane J. Cook, Kyle D. Feuz, and Narayanan C. Krishnan. 2012. Transfer Learning for Activity Recognition. *Knowledge and Information Systems* 36 (2012), 537–556.

Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2008. Translated learning: Transfer learning across different feature spaces. In *Advances in Neural Information Processing Systems*. 353–360.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 210–219. DOI:http://dx.doi.org/10.1145/1281192.1281218

Hal Daumé, Abhishek Kumar, and Avishek Saha. 2010. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems*. 478–486.

Hal Daumé, III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intillegence Research* 26, 1 (May 2006), 101–126. http://dl.acm.org/citation.cfm?id=1622559.1622562

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting Association of Computing Linguistics*. 256–263.

Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*. Springer-Verlag, 1–15. http://dl.acm.org/citation.cfm?id=648054.743935

Lixin Duan, Dong Xu, and Ivor W. Tsang. 2012. Learning with Augmented Features for Heterogeneous Domain Adaptation. In *Proceedings of the International Conference on Machine Learning*. Omnipress, Edinburgh, Scotland, 711–718.

Charles Elkan. 2001. The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2 (IJCAI'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 973–978. http://dl.acm.org/citation.cfm?id=1642194.1642224

Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. System Sci.* 55, 1 (1997), 119 – 139. DOI:http://dx.doi.org/10.1006/jcss.1997.1504

L.K. Hansen and P. Salamon. 1990. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 10 (1990), 993–1001. DOI:http://dx.doi.org/10.1109/34.58871

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 50–57.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 111–119.

K. Lang and others. 1995. News weeder: Learning to filter netnews. In *12th International Conference of Machine Learning*. 331–339.

Thomas M. Mitchell. 1997. *Machine Learning* (1 ed.). McGraw-Hill, Inc., New York, NY, USA, Chapter Bayesian Learning, 154 – 200.

S.J. Pan and Q. Yang. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22, 10 (2010), 1345–1359. DOI:http://dx.doi.org/10.1109/TKDE.2009.191

Sinno Jialin Pan, James T Kwok, and Qiang Yang. 2008. Transfer Learning via Dimensionality Reduction.. In *AAAI*, Vol. 8. 677–682.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*. 751–760.

S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. 2011. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks* 22, 2 (2011), 199–210. DOI:http://dx.doi.org/10.1109/TNN.2010.2091281

Weike Pan, Erheng Zhong, and Qiang Yang. 2012. Transfer learning for text mining. In *Mining Text Data*. Springer, 223–257.

Peter Prettenhofer and Benno Stein. 2011. Cross-lingual adaptation using structural correspondence learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 1 (2011), 13.

P. Rashidi and D.J. Cook. 2010. Multi home transfer learning for resident activity discovery and recognition. In *KDD Knowledge Discovery from Sensor Data*. 56–63. http://www.eecs.wsu.edu/~cook/pubs/kdd10p1.pdf

P. Rashidi and D.J. Cook. 2011. Activity knowledge transfer in smart environments. *Pervasive and Mobile Computing* 7, 3 (2011), 331–343. DOI:http://dx.doi.org/10.1016/j.pmcj.2011.02.007

E. Thorndike and R.S. Woodworth. 1901. The influence of improvement in one mental function upon the efficiency of other functions.(I). *Psychological review* 8, 3 (1901), 247–261.

S. Thrun. 1996. *Explanation-based neural network learning: A lifelong learning approach*. Kluwer Academic Publishers.

S. Thrun and L. Pratt. 1998. *Learning to learn*. Kluwer Academic Publishers. http://books.google.com/books?id=WY4UzYZNpN4C

T. van Kasteren, G. Englebienne, and B. Kröse. 2008. Recognizing activities in multiple contexts using transfer learning. In *AAAI AI in Eldercare Symposium*. https://www.aaai.org/Papers/Symposia/Fall/2008/FS-08-02/FS08-02-023.pdf

T. van Kasteren, G. Englebienne, and B. Kröse. 2010. Transferring Knowledge of Activity Recognition across Sensor Networks. In *Pervasive Computing*, Patrik Floren, Antonio Krger, and Mirjana Spa-

sojevic (Eds.). Lecture Notes in Computer Science, Vol. 6030. Springer Berlin / Heidelberg, 283–300. http://www.springerlink.com/index/5u5115k1h2k2h558.pdf

Ricardo Vilalta and Youssef Drissi. 2002. A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review* 18 (2002), 77–95. DOI:http://dx.doi.org/10.1023/A:1019956318069

David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5 (1992), 241–259.

Lin Xian-ming and Li Shao-zi. 2009. Transfer AdaBoost learning for action recognition. In *IT in Medicine Education, 2009. ITIME '09. IEEE International Symposium on*, Vol. 1. 659 –664. DOI:http://dx.doi.org/10.1109/ITIME.2009.5236340

Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. 2009. Heterogeneous transfer learning for image clustering via the social web. 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 1-Volume 1*. Association for Computational Linguistics, 1–9.

Yi Yao and Gianfranco Doretto. 2010. Boosting for transfer learning with multiple sources. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1855–1862.

Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak Turaga, and Olivier Verscheure. 2009. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1027–1036.

Received January 2013; revised February 2014; accepted March 2014