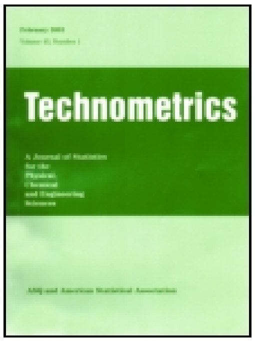


Technometrics

Taylor & Francis
Taylor & Francis Group

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: http://www.tandfonline.com/loi/utch20

A Transfer Learning Approach for Predictive Modeling of Degenerate Biological Systems

Na Zou, Yun Zhu, Ji Zhu, Mustafa Baydogan, Wei Wang & Jing Li

Supplementary materials for this article are available online. Please go to http://www.tandfonline.com/r/TECH

A Transfer Learning Approach for Predictive Modeling of Degenerate Biological Systems

Na Zou

School of Computing, Informatics,
and Decision Systems Engineering
Arizona State University
Tempe, AZ, 85281
(*nzou1@asu.edu*)

Yun Zhu

Department of Chemistry and Biochemistry
University of California, San Diego
La Jolla, CA, 92093
(*zhuyun97@gmail.com*)

Ji Zhu

Department of Statistics
University of Michigan
Ann Arbor, MI, 48109
(*jizhu@umich.edu*)

Mustafa Baydogan

Department of Industrial Engineering
Bogazici University, 34342
Bebek/İstanbul, Turkey
(*mustafa.baydogan@boun.edu.tr*)

Wei Wang

Department of Chemistry and Biochemistry
University of California, San Diego
La Jolla, CA, 92093
(*wei-wang@ucsd.edu*)

Jing Li

School of Computing, Informatics, and Decision Systems
Engineering
Arizona State University
Tempe, AZ, 85281
(*jinglz@asu.edu*)

Modeling of a new domain can be challenging due to scarce data and high-dimensionality. Transfer learning aims to integrate data of the new domain with knowledge about some related old domains, to model the new domain better. This article studies transfer learning for degenerate biological systems. Degeneracy refers to the phenomenon that structurally different elements of the system perform the same/similar function or yield the same/similar output. Degeneracy exists in various biological systems and contributes to the heterogeneity, complexity, and robustness of the systems. Modeling of degenerate biological systems is challenging and models enabling transfer learning in such systems have been little studied. In this article, we propose a predictive model that integrates transfer learning and degeneracy under a Bayesian framework. Theoretical properties of the proposed model are studied. Finally, we present an application of modeling the predictive relationship between transcription factors and gene expression across multiple cell lines. The model achieves good prediction accuracy, and identifies known and possibly new degenerate mechanisms of the system. Supplementary materials for this article are available online.

KEY WORDS: Bayesian; Regression.

1. INTRODUCTION

An essential problem in biological system informatics is to build a predictive model with high-dimensional predictors. This can be a challenging problem for a new domain in which the data are scarce due to resource limitation or timing of the modeling. Often times, there may be some old domains related to but not exactly the same as the new domain, in which abundant knowledge have existed. Transfer learning, in the context of this article, refers to statistical methods that integrate knowledge of the old domains and data of the new domain in a proper way, to develop a model for the new domain that is better than using the data of the new domain alone. Next, we give three examples in which transfer learning is desirable:

(i) Modeling the predictive relationship between transcription factors (TFs) and gene expression is of persistent interest in system biology. TFs are proteins that bind to the upstream region of a gene and regulate the expression level of the gene. Knowledge of TFs-expression relationship may have existed for a number of known cell lines. To model a new cell line, it is advantageous to adopt transfer learning to make good use of the existing knowledge of the known cell lines, because the experimental data for the new cell line may be limited.

(ii) In cancer genomics, a prominent interest is to use gene expression to predict disease prognosis. Knowledge may have existed for several known subtypes of a cancer. When a new subtype is discovered, the patient number is usually limited. Transfer learning can help establish a model for the new subtype timely and reliably by transferring knowledge of the known subtypes to the modeling of the new subtype.

© 2015 American Statistical Association and
the American Society for Quality
TECHNOMETRICS, AUGUST 2015, VOL. 57, NO. 3
DOI: 10.1080/00401706.2015.1044117
Color versions of one or more of the figures in the article can be found
online at *www.tandfonline.com/r/tech*.

(iii) Biomedical imaging has been used to predict cognitive performance. In longitudinal studies, a particular interest is to follow along a cohort of patients with a brain disease such as the Alzheimer's disease to identify the imaging-cognition associations at different stages of the disease advancement. Patient drop-off is common, leaving less data for use in modeling later stages of the disease. Transfer learning can play an important role here by integrating the limited data with knowledge from the earlier stages.

This article studies transfer learning in *degenerate* biological systems. Degeneracy is a well-known characteristic of biological systems. In the seminal article by Edelman and Gally (2001), degeneracy was referred to as the phenomenon that *structurally different elements perform the same/similar function or yield the same/similar output*. The article also provided ample evidence to show that degeneracy exists in many biological systems and processes. A closely related concept to degeneracy is redundancy, which may be more familiar to the engineering society. Degeneracy is different from redundancy in three major aspects: (a) Degeneracy is a characteristic for structurally *different* elements, whereas redundancy is one for structurally *identical* elements. In fact, although prevalent in engineering systems, true redundancy hardly exists in biological systems due to the rare presence of identical elements. (b) Degenerate elements work in a stochastic fashion, whereas redundant elements work according to deterministic design logic, for example, A will work if B fails. (c) Degenerate elements deliver the same/similar function under *some* condition. When the condition changes, these degenerate elements may deliver different functions. This property leads to strong selection under environmental changes. In essence, degeneracy is a prerequisite for natural selection and evolution. Redundancy, on the other hand, does not have such a strong tie to environment.

Degeneracy exists in all the three examples presented earlier. In (i), due to the difficulty of measuring TFs directly and precisely, the association between TFs and gene expression is usually studied by modeling the association between TF binding sites and gene expression. The binding site of a TF is a short DNA sequence where the TF binds. It is known that the same TF can have alternative binding sites (Li and Zhang 2010), and as a result, these alternative binding sites should have similar association with gene expression. The alternative binding sites of the same TF are degenerate elements. In (ii), genes in the same pathway may be degenerate elements in the sense that different genes in the pathway may have similar association with disease prognosis. This explains the growing interest in cancer genomics that aims at identifying how gene pathway as a whole affects prognosis rather than the effect of individual genes (Vogelstein and Kinzler 2004). In (iii), brain regions that are strongly connected in a brain connectivity network may be degenerate elements because their functions may have similar association with cognition (Huang et al. 2013).

Although degeneracy has been extensively discussed in the biological literature, its implication to statistical modeling has not been rigorously defined. Consider a biological system with Q elements, $X_1, \ldots, X_Q$, jointly performing a function or yielding an output Y. For example, $X_1, \ldots, X_Q$ may be Q potential binding sites of some TFs of interest, which bind to the upstream region of a gene to regulate the gene's expression. Y is expression level of the gene. In the context of a predictive model, $X_1, \ldots, X_Q$ are predictors and Y is the response variable. If a subset $\{X_{(1)}, \ldots, X_{(q)}\} \subset \{X_1, \ldots, X_Q\}$ consists of degenerate elements, for example, they are potential binding sites of a TF, then according to the definition of degeneracy, $\{X_{(1)}, \ldots, X_{(q)}\}$ should satisfy two conditions: (1) they are structurally different; (2) they perform a similar function, which means that their respective coefficients, $\{w_{(1)}, \ldots, w_{(q)}\}$, that link them to Y should satisfy $\|w_{(i)} - w_{(j)}\| < \varepsilon$, $\forall i, j \in \{1, \ldots, q\}, i \neq j$. $\|\cdot\|$ is an appropriate norm and ε is a biologically defined threshold. A degenerate system may contain more than one subset of degenerate elements such as the subsets corresponding to different TFs. The challenge in modeling a degenerate system is how to build the biological knowledge about the degeneracy into statistical modeling, especially considering that the knowledge is often qualitative and with uncertainty.

In this article, we propose a predictive model that integrates transfer learning and degeneracy under a Bayesian framework. A Bayesian framework is appropriate in the sense that it can encode the available but largely qualitative/uncertain biological knowledge about degeneracy into a prior, and then use data to refine the knowledge. A Bayesian framework is also appropriate for accounting for the correlation between the old domains and new domain to enable transfer learning. The major contributions of this research include:

- *Formulation*: We propose a unique prior for the model coefficients of the old domains and new domain. This prior has two hyperparameters characterizing the degeneracy and the correlation structure of the domains, respectively. We propose to use a graph to represent the qualitative knowledge about degeneracy, and set the corresponding hyperparameter to be the Laplacian matrix of the graph. This has an effect of pushing the coefficients of degenerate elements to be similar, thus nicely reflecting the nature of degenerate elements that they perform a similar function. We also propose an efficient algorithm that allows estimation of the other hyperparameter together with the model coefficients, so that the correlation structure between domains does not need to be specified *a priori* but can be learned from data.
- *Theoretical properties*: We perform theoretical analysis to answer several important questions, such as: what difference it will make by transferring the knowledge/models of the old domains instead of the data? It is common in biology and medicine that when a new domain is being studied, the researcher can only access the knowledge/models of the old domains through published literature, but not the data of these domains due to ownership or confidentiality. Other questions include: Is transfer learning always better than learning using the data of the new domain alone? What knowledge from old domains or what type of old domains is most helpful for transfer learning?
- *Application*: We apply the proposed method to a real-world application of using TF binding sites to predict gene expression across multiple cell lines. Our method shows better prediction accuracy compared with competing methods. The biological findings revealed by our model are also consistent with the literature.

2. REVIEW OF EXISTING RESEARCH

The existing transfer learning methods primarily fall into three categories: instance transfer, feature transfer, and parameter transfer. The basic idea of instance transfer is to reuse some samples/instances in the old domains as auxiliary data for the new domain (Dai et al. 2007). For example, Dai et al. (2007) proposed a boosting algorithm called TrAdaBoost to iteratively reweight samples in the old domains to identify samples that are helpful for modeling the new domain. Although intuitive, instance transfer may be questioned for its validity. For example, if the old and new domains are two subtypes of a cancer, using the data of some patients in one subtype to model another subtype suggests that these patients are misdiagnosed, which is not a reasonable assumption. Feature transfer aims to identify good feature representations shared by the old and new domains. In an earlier work by Caruana (1997), the features are shared hidden layers for the neural network models across the domains. More recently, Argyriou et al. (2007) and Evgeniou and Pontil (2007) proposed to map the original high-dimensional predictor space to a low-dimensional feature space and the mapping is shared across the domains. Nonlinear mapping was studied by Jebara (2004) for support vector machines (SVMs) and by Rückert and Kramer (2008) who designed a kernel-based approach aiming at finding a suitable kernel for the new domain. Interpretability, for example, physical meaning of the shared features, is an issue for feature transfer especially nonlinear approaches. Parameter transfer assumes that the old and new domains share some model parameters. For example, Liu, Ji, and Ye (2009) adopted L21-norm regularization for linear models to encourage the same predictors to be selected across the domains. Regularized approaches for nonlinear models like SVMs were also studied (Evgeniou and Pontil 2004). In addition to regularization, Bayesian statistics provide a nice framework by assuming the same prior distribution for the model parameters across the domains, which has been adopted for Gaussian process models (Lawrence and Platt 2004; Bonilla, Chai, and Williams 2008).

The proposed method in this article falls into the category of parameter transfer. Our method is different from the existing transfer learning methods in the following aspects: First, the existing methods do not model degeneracy. Second, they usually assume that the old domains have similar correlations to the new domain, which may not be a robust approach when the old domains have varying or no correlations with the new domain. In contrast, our method estimates the correlation structure between domains from data, and therefore can adaptively decide how much information to transfer from each old domain. Third, while showing good empirical performance, the existing research provides limited investigation on the theoretical properties of transfer learning.

3. THE PROPOSED TRANSFER LEARNING MODEL FOR DEGENERATE SYSTEMS

3.1 Formulation Under a Bayesian Framework

Let $\mathbf{X} = (X_1, \ldots, X_Q)$ denote Q predictors and Y denote the response. Assume that there are K related domains. Domains 1 to $K-1$ are old domains and domain K is a new domain. For each domain k, there is a model that links $\mathbf{X}$ to Y by coefficients $\mathbf{w}_k$. If $Y \in R$, a common model is a linear regression, $Y = \mathbf{X}\mathbf{w}_k + \varepsilon_k$. If $Y \in \{-1, 1\}$, the model can be a logistic regression, $\log \frac{P(Y=1)}{P(Y=-1)} = \mathbf{X}\mathbf{w}_k$. We propose the following prior for $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)$:

$$p\left(\mathbf{W} | \boldsymbol{\Omega}, \boldsymbol{\Phi}, b\right) \propto \prod_{k=1}^{K} \text{Laplace}\left(\mathbf{w}_k; b\right) \times \text{MN}\left(\mathbf{W}; 0, \boldsymbol{\Omega}, \boldsymbol{\Phi}\right). \quad (1)$$

This prior is formed based on the following considerations:

- Laplace($\mathbf{w}_k; b$) is a Laplace distribution for $\mathbf{w}_k$. Using a Laplace distribution in the prior is to facilitate “sparsity” in model estimation. The well-known lasso model facilitates sparsity by imposing an L1 penalty on regression coefficients. Tibshirani (1996) showed that the lasso estimate is equivalent to a Bayesian maximum-a-posteriori (MAP) estimate with a Laplace prior. Sparsity is an advantageous property for high-dimensional problems, which is the target setting of this article.
- MN($\mathbf{W}; \mathbf{0}, \boldsymbol{\Omega}, \boldsymbol{\Phi}$) is a zero-mean matrix-variate normal distribution. $\boldsymbol{\Omega} \in \mathbb{R}^{K\times K}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{Q\times Q}$ are called column and row covariance matrices, respectively. It can be shown that $\text{cov}(\mathbf{w}^q) = \boldsymbol{\Phi}_{qq}\boldsymbol{\Omega}$. $\mathbf{w}^q$ is the qth row of $\mathbf{W}$, which consists of regression coefficients for all the K domains corresponding to the qth predictor. $\boldsymbol{\Phi}_{qq}$ is the qth diagonal element of $\boldsymbol{\Phi}$. cov($\cdot$) denotes the covariance matrix of a vector. Therefore, $\boldsymbol{\Omega}$ encodes the prior knowledge about the correlation structure of the domains. Furthermore, it can be shown that $\text{cov}(\mathbf{w}_k) = \boldsymbol{\Omega}_{kk}\boldsymbol{\Phi}$. Therefore, $\boldsymbol{\Phi}$ encodes the prior knowledge about the correlation structure of the regression coefficients, that is, the degeneracy.

Next, we propose two modeling strategies depending on the availability of data. In Case I, data of the old domains 1 to $K-$ 1 is available. In Case II, data of the old domain is not available but only the knowledge/models. The latter case is more common especially in biology and medicine. At the time a new cell line or a new subtype of a disease is being studied, the researcher may only have access to the data of the new domain. Although he/she may gather abundant knowledge about existing cell lines or disease subtypes from the published works of other researchers, he/she can hardly access the data due to ownership or confidentiality.

Case I: *Model the new domain using data of all the domains.*

Let $\mathbf{y}_k$ and $\mathbf{X}_k$ denote the data for the response and predictors of the kth domain $k = 1, \ldots, K$. The likelihood for $\mathbf{y}_k$ given $\mathbf{X}_k$ and $\mathbf{w}_k$ is $p(\mathbf{y}_k|\mathbf{X}_k, \mathbf{w}_k) \sim N(\mathbf{y}_k; \mathbf{X}_k\mathbf{w}_k, \sigma^2\mathbf{I}_{n_k})$. The posterior distribution of $\mathbf{W}$ based on the likelihood and the prior in (1) is

$$p\left(\mathbf{W} | \{\mathbf{y}_k, \mathbf{X}_k\}_{k=1}^{K}, \boldsymbol{\Omega}, \boldsymbol{\Phi}, b\right) \propto p\left(\mathbf{W} | \boldsymbol{\Omega}, \boldsymbol{\Phi}, b\right) \prod_{k=1}^{K} p\left(\mathbf{y}_k | \mathbf{X}_k, \mathbf{w}_k\right). \quad (2)$$

One way for estimating the regression coefficients of the new domain, $\mathbf{w}_K$, is to find a $\hat{\mathbf{W}}$ that maximizes the posterior distribution of $\mathbf{W}$ in (2), that is, $\hat{\mathbf{W}}$ is a Bayesian MAP estimate for $\mathbf{W}$. This will naturally produce an estimate for $\mathbf{w}_K$, $\hat{\mathbf{w}}_K$, as in $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_K)$, and estimates for the old domains, $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{K-1}$, as a side product. Through some algebra, it can be derived that $\hat{\mathbf{W}}$ can be obtained by solving the following optimization:

$$\hat{\mathbf{W}}^{\mathrm{I}} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma^2} \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k\|_2^2 + \frac{1}{b} \|\mathbf{W}\|_1 \right. \\ \left. + \frac{1}{2} \left(Q \log |\mathbf{\Omega}| + K \log |\mathbf{\Phi}| + \operatorname{tr}\left(\mathbf{\Phi}^{-1} \mathbf{W} \Omega^{-1} \mathbf{W}^T \right) \right) \right\}, \tag{3}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the L2 and L1 norms, respectively. The superscript "I" is used to differentiate this estimate from the one that will be presented in Case II.

Equation (3) assumes that $\mathbf{W}$ is the only parameter to be estimated whereas σ^2, b, $\mathbf{\Omega}$, and $\mathbf{\Phi}$ are known. This assumption may be too strict. To relax this assumption, we propose the following approach: Let $\lambda_1 = 2\sigma^2/b$ and $\lambda_2 = \sigma^2$. Then, (3) is equivalent to (4):

$$\hat{\mathbf{W}}^{\mathrm{I}} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k\|_2^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \Big(Q \log |\mathbf{\Omega}| \right. \\ \left. + K \log |\mathbf{\Phi}| + \operatorname{tr}\left(\mathbf{\Phi}^{-1} \mathbf{W} \Omega^{-1} \mathbf{W}^T \right) \Big) \right\}. \tag{4}$$

$\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ serve as regularization parameters to control the sparsity of $\hat{\mathbf{W}}^{\mathrm{I}}$ and the amount of prior knowledge used for estimating $\mathbf{W}$, respectively. λ_1 and λ_2 can be selected by a grid search according to some model selection criterion. This strategy for "estimating" σ^2 and b enjoys computational simplicity and was also adopted by other articles (Tibshirani 1996; Genkin et al. 2007; Liu et al. 2009). Furthermore, hyperparameters $\mathbf{\Phi}$ and $\mathbf{\Omega}$ are matrices of potentially high-dimensionality, the specification of which is more involved and will be discussed in detail in Section 3.3. For now, we assume that $\mathbf{\Phi}$ and $\mathbf{\Omega}$ are known.

Case II: *Model the new domain using data of the new domain and knowledge/models of old domains.*

To develop a model for this case, we first reorganize the terms in (4) to separate the terms involving old domains from those involving only the new domain. Denote the objective function in (4) by $f(\mathbf{W})$. Let $\tilde{\mathbf{W}} = (\mathbf{w}_1, \ldots, \mathbf{w}_{K-1})$, so $\mathbf{W} = (\tilde{\mathbf{W}}, \mathbf{w}_K)$. Also let $\mathbf{\Omega} = [\begin{smallmatrix} \tilde{\mathbf{\Omega}} & \boldsymbol{\varpi}_K \\ \boldsymbol{\varpi}_K^T & \varsigma_K \end{smallmatrix}]$. Then, it can be shown that (please see derivation in supplementary materials):

$$f(\mathbf{W}) = f\left(\tilde{\mathbf{W}}\right) + g\left(\mathbf{w}_K | \tilde{\mathbf{W}}\right). \tag{5}$$

$f(\tilde{\mathbf{W}})$ takes the same form as $f(\mathbf{W})$ but for the $K-1$ old domains, that is,

$$f\left(\tilde{\mathbf{W}}\right) = \sum_{k=1}^{K-1} \|\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k\|_2^2 + \lambda_1 \left\|\tilde{\mathbf{W}}\right\|_1 + \lambda_2 \Bigg(Q \log \left|\tilde{\mathbf{\Omega}}\right| \\ + (K-1) \log |\mathbf{\Phi}| + \operatorname{tr}\left(\mathbf{\Phi}^{-1} \tilde{\mathbf{W}} \tilde{\mathbf{\Omega}}^{-1} \tilde{\mathbf{W}}^T \right) \Bigg), \tag{6}$$

and

$$g\left(\mathbf{w}_K | \tilde{\mathbf{W}}\right) = \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 + \lambda_1 \|\mathbf{w}_K\|_1 + \lambda_2 \big(\log |\mathbf{\Sigma}_K| \\ + (\mathbf{w}_K - \boldsymbol{\mu}_K)^T \mathbf{\Sigma}_K^{-1} (\mathbf{w}_K - \boldsymbol{\mu}_K) \big), \tag{7}$$

where

$$\boldsymbol{\mu}_K = \tilde{\mathbf{W}} \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K, \tag{8}$$

and

$$\mathbf{\Sigma}_K = \left(\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K \right) \mathbf{\Phi}. \tag{9}$$

When data from the old domains are not available but only the knowledge/model in the form of $\tilde{\mathbf{W}} = \tilde{\mathbf{W}}^*$, the $f\left(\tilde{\mathbf{W}}^*\right)$ in (5) becomes a constant. Therefore, minimizing $f(\mathbf{W})$ becomes minimizing $g(\mathbf{w}_K | \tilde{\mathbf{W}}^*)$, that is,

$$\hat{\mathbf{w}}_K^{\mathrm{II}} = \underset{\mathbf{w}_K}{\operatorname{argmin}} \quad g\left(\mathbf{w}_K | \tilde{\mathbf{W}}^*\right) = \underset{\mathbf{w}_K}{\operatorname{argmin}} \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 \\ + \lambda_1 \|\mathbf{w}_K\|_1 + \lambda_2 \left(\log |\mathbf{\Sigma}_K| + (\mathbf{w}_K - \boldsymbol{\mu}_K)^T \mathbf{\Sigma}_K^{-1} (\mathbf{w}_K - \boldsymbol{\mu}_K) \right), \tag{10}$$

with $\boldsymbol{\mu}_K = \tilde{\mathbf{W}}^* \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K$ and $\mathbf{\Sigma}_K$ given in (9).

Finally, we would like to assess the difference between the estimates in Case I and Case II, that is, $\hat{\mathbf{w}}_K^{\mathrm{I}}$ as in $\hat{\mathbf{W}}^{\mathrm{I}} = (\hat{\mathbf{w}}_1^{\mathrm{I}}, \ldots, \hat{\mathbf{w}}_K^{\mathrm{I}})$ and $\hat{\mathbf{w}}_K^{\mathrm{II}}$. Theorem 1 shows that the estimate in Case II is no better than Case I in terms of minimizing the objective function in the estimation (proof in supplementary materials). Case II is only as good as Case I when the knowledge/model of the old domains can be provided in its optimal form. The intuitive explanation about this finding is that since Case II uses the knowledge of the old domains, which may contain uncertainty or noise, it is only sub-optimal compared with using the data of the old domains directly (i.e., Case I).

Theorem 1. $f((\tilde{\mathbf{W}}^*, \hat{\mathbf{w}}_K^{\mathrm{II}})) \geq f(\hat{\mathbf{W}}^{\mathrm{I}})$. When $\tilde{\mathbf{W}}^* = (\hat{\mathbf{w}}_1^{\mathrm{I}}, \ldots, \hat{\mathbf{w}}_{K-1}^{\mathrm{I}})$, $\hat{\mathbf{w}}_K^{\mathrm{II}} = \hat{\mathbf{w}}_K^{\mathrm{I}}$, and $f((\tilde{\mathbf{W}}^*, \hat{\mathbf{w}}_K^{\mathrm{II}})) = f(\hat{\mathbf{W}}^{\mathrm{I}})$.

3.2 Theoretical Properties of Transfer Learning

This section aims to perform theoretical analysis to address the following questions: Is transfer learning always better than single-domain learning, that is, learning using only the data of the new domain but neither the data nor the knowledge of the old domains (Theorem 2)? What knowledge from old domains or what type of old domains is most helpful for learning of the new domain (Theorems 3)? Please see proofs of these Theorems in supplementary materials.

Let (11) and (12) be the transfer learning and single-domain learning formulations targeted in this section, respectively. $\lambda \geq 0$. When $\lambda = 0$, (11) becomes (12):

$$\breve{w}_K = \operatorname{argmin}_{\mathbf{w}_K} \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 + \lambda (\mathbf{w}_K - \boldsymbol{\mu}_K)^T (\mathbf{w}_K - \boldsymbol{\mu}_K), \tag{11}$$

$$\breve{w}_K = \operatorname{argmin}_{\mathbf{w}_K} \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2. \tag{12}$$

Comparing (11) with (10) in the previous section, it can be seen that (11) is obtained from (10) by dropping the L1 norm, $\mathbf{w}_K 1$, and making $\mathbf{\Phi} = \mathbf{I}$ and $\lambda = \frac{\lambda_2}{\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K}$. This is to single out transfer learning from the sparsity and degeneracy considerations in (10), so that the discussion in this section will be focused on transfer learning. Let MSE($\cdot$) denote the mean square error (MSE) of an estimator. It is known that the MSE is the sum of the variance and squared bias of an estimator, and is a commonly used criterion for comparing/choosing estimators.

Theorem 2. There always exists a $\lambda > 0$ such that $\text{MSE}(\hat{\mathbf{w}}_K) < \text{MSE}(\breve{w}_K)$.

Theorem 2 provides theoretical assurance that the model coefficients of the new domain, $\mathbf{w}_K$, can be better estimated by transfer learning than single-domain learning in the sense of a smaller MSE. Next, we would like to investigate what type of knowledge from old domains or what type of old domains helps learning of the new domain better. Because knowledge from old domains is represented by $\boldsymbol{\mu}_K$ in (8), the question becomes what property of $\boldsymbol{\mu}_K$ leads to a better transfer learning. Definition 1 defines a distance measure between the knowledge from old domains, $\boldsymbol{\mu}_K$, and the new domain, $\mathbf{w}_K$, called the *transfer learning distance*. Theorem 3 further proves that the knowledge for old domains that has a smaller transfer learning distance to the new domain will help achieve a smaller MSE in modeling the new domain.

Definition 1 (transfer learning distance). Define a transfer learning distance to be $d(\boldsymbol{\mu}_K; \lambda) \triangleq (\mathbf{w}_K - \boldsymbol{\mu}_K)^T \mathbf{B}^T \mathbf{B} (\mathbf{w}_K - \boldsymbol{\mu}_K)$, where $\mathbf{B} = (\mathbf{X}_K^T \mathbf{X}_K + \lambda \mathbf{I})^{-1}$.

The geometric interpretation of this distance measure is the following: Let $\boldsymbol{\Lambda}$ be a diagonal matrix of eigenvalues $\gamma_1, \ldots, \gamma_Q$ for $\mathbf{X}_K^T \mathbf{X}_K$ and $\mathbf{P}$ be a matrix consisting of corresponding eigenvectors, that is, $\mathbf{X}_K^T \mathbf{X}_K = \mathbf{P}^T \boldsymbol{\Lambda} \mathbf{P}$. Furthermore, let $\boldsymbol{\alpha} \triangleq \mathbf{P}(\boldsymbol{\mu}_K - \mathbf{w}_K)$. The elements of $\boldsymbol{\alpha}$, $\alpha_1, \ldots, \alpha_Q$, are indeed projections of $\boldsymbol{\mu}_K - \mathbf{w}_K$ onto the principal component axes of the data. Then, it can be derived that the transfer learning distance is $d(\boldsymbol{\mu}_K; \lambda) = \sum_{i=1}^{Q} \frac{\alpha_i^2}{(\gamma_i + \lambda)^2}$.

Furthermore, suppose that there are two sets of knowledge from old domains to be compared, that is, $\boldsymbol{\mu}_K^{(1)}$ and $\boldsymbol{\mu}_K^{(2)}$. Let $\text{MSE}(\hat{\mathbf{w}}_K^{(i)}; \lambda)$ be the MSE of the estimator for $\hat{\mathbf{w}}_K$ using (8) with $\boldsymbol{\mu}_K = \boldsymbol{\mu}_K^{(i)}$. Let $\min_\lambda \text{MSE}(\hat{\mathbf{w}}_K^{(i)})$ denote the smallest MSE over all possible values of λ. $i = 1, 2$.

Theorem 3. If $d(\boldsymbol{\mu}_K^{(1)}; \lambda) \le d(\boldsymbol{\mu}_K^{(2)}; \lambda)$ for $\forall \lambda > 0$, then $\min_\lambda \text{MSE}(\hat{\mathbf{w}}_K^{(1)}) \le \min_\lambda \text{MSE}(\hat{\mathbf{w}}_K^{(2)})$.

For better illustration, we show the comparison of MSEs between five sets of knowledge from old domains in Figure 1. This is a simple example that consists of only one predictor. Therefore, $\mathbf{w}_K$ and $\boldsymbol{\mu}_K$ are scalars, w_K and μ_K. Assume that $w_K = 3$. $\mu_K^{(1)}$ through $\mu_K^{(5)}$ are 1.3, 1.6, 1.9, 2.2, 2.5, respectively, that is, they are more and more close to the new domain in transfer learning distance. Figure 1 plots the MSEs of transfer learning using each of the five sets of knowledge. The observations are: (i) for each curve, there exists a $\lambda > 0$ whose corresponding MSE is smaller than the MSE of single-domain learning (i.e., the intercept on the vertical axis). This demonstrates Theorem 2. (ii) The smaller the transfer learning distance, the smaller the minimum MSE. This demonstrates Theorem 3.

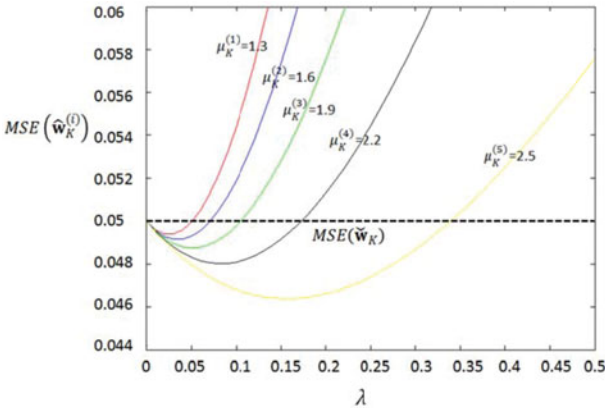


Figure 1. Transfer learning MSEs of five sets of knowledge from old domains, $\text{MSE}(\hat{\mathbf{w}}_K^{(i)})$, $i = 1, \ldots, 5$, and single-domain MSE, $\text{MSE}(\hat{\mathbf{w}}_K)$ with true $w_K = 3$.

Finally, we would like to discuss some practical implication of the theorems. Theorem 2 needs little assumption to hold. However, this does not imply that transfer learning always gives better results than single-domain learning in practice. This is because in practice, λ is selected by a grid search according to some model selection criterion such as Bayesian information criterion (BIC) and cross-validation. The λ that makes the MSE of the transfer learning estimator smaller than single-domain learning may be missed in this practical search. Further, as indicated by Theorem 3 and Figure 1, this risk is higher when knowledge from old domain is farther from the new domain in transfer learning distance. For example, in Figure 1, when the knowledge is far away from the new domain, for example, the top red curve, the range of λ within which the curve falls below the MSE of single-domain learning, MSE ($\breve{w}_K$), is small. This small range of λ may be missed in a practical grid search, resulting in a transfer learning approach with worse performance than single-domain learning.

3.3 Strategies for Handling Hyperparameters and an Efficient Algorithm

3.3.1 Specifying Hyperparameter $\boldsymbol{\Phi}$ by a Graph that Encodes Degeneracy. According to the discussion about the prior in (1), $\boldsymbol{\Phi}$ encodes the prior knowledge about the degeneracy of the system. In real-world applications, it is common that some qualitative knowledge about the degeneracy exists, which can be represented by a graph $G = \{\mathbf{X}, \mathbf{E}\}$. The nodes in the graph are elements of the system, that is, predictors $\mathbf{X}$ in the predictive model. $\mathbf{E} = \{X_i \sim X_j\}$ is a set of edges. a_{ij} is the edge weight. No edge between two nodes implies that the nodes are not degenerate elements to each other. If there is an edge between two nodes, the edge weight reflects the level of certainty that the nodes are degenerate elements. Next, we will discuss how to construct such a graph for the three examples presented in Introduction.

In (i), nodes/predictors are potential TF binding sites. A potential binding site is a short DNA sequence in the upstream promoter region of a gene, for example, ACGCGT, ATGCGC. The letters in each word (i.e., each binding site) can only be from the DNA alphabet $\{A, C, G, T\}$. If focusing on all κ-letter-long words, called κ-mers, there will be 4^κ nodes in the graph. It is known that the binding sites with similar word composition are more likely to be alternative binding sites of the same TF (Li et al. 2010). The similarity between two binding sites can be measured by the number of letters they have in common in their respective words.

For example, the similarity between ACGCGT and ATGCGC is 4, because they share four letters in the same position. A formal definition of this similarity between two binding sites X_i and X_j is $\kappa - H\{X_i, X_j\}$. $H\{X_i, X_j\}$ is the so-called Hemming distance defined as $H\{X_i, X_j\} = \sum_{l=1}^{L} I\left(c_{il} \neq c_{jl}\right)$ (Li and Zhang 2010). $I(\cdot)$ is an indicator function. c_{il} is the lth letter in the word of binding site X_i. Using this similarity measure, two nodes X_i and X_j do not have an edge if they do not have any common letter in the same position; they have an edge otherwise and the edge weight is their similarity. Likewise, in example (ii), nodes of the graph are genes and edges can be put between genes according to known pathway databases such as KEGG (*http://www.genome.jp/kegg/*) and BioCarta (*http://www.biocarta.com/*). In example (iii), nodes of the graph are brain regions and edges can be put between brain regions with known functional or anatomical connectivity.

To incorporate the graph into our model, the graph is first converted to a Laplacian matrix, $\mathbf{L}$, that is,

$$\mathbf{L}_{ij} = \begin{cases} d_i & \text{if } i = j \\ -a_{ij} & \text{if} i \neq j \text{ and } X_i \sim X_j \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $d_i = \sum_{X_i \sim X_j} a_{ij}$ is called the degree of node X_i. It is known that $\mathbf{L}$ is always nonnegative definite and it encodes many properties of the graph (Chung 1997). If the graph encodes the degeneracy of the system, $\mathbf{L}$ can be reasonably used to replace the $\mathbf{\Phi}^{-1}$ in the optimization problems in Case I and Case II, that is, (4) and (9). Then, we obtain the following optimization problems for Case I and Case II, respectively.

$$\text{Case I}: \hat{\mathbf{W}}^{\mathrm{I}} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k\|_2^2 + \lambda_1 \|\mathbf{W}\|_1 \right. \\ \left. + \lambda_2 \left(Q \log |\mathbf{\Omega}| + \operatorname{tr}\left(\mathbf{L} \mathbf{W} \Omega^{-1} \mathbf{W}^T\right)\right)\right\}, \tag{14}$$

$$\text{Case II}: \hat{\mathbf{w}}_K^{\mathrm{II}} \\ = \operatorname{argmin}_{\mathbf{w}_K} \left\{ \begin{array}{c} \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 + \lambda_1 \|\mathbf{w}_K\|_1 \\ + \lambda_2 \left(Q \log \left(\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K \right) \right. \\ \left. + \dfrac{1}{\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K} \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right)^T \mathbf{L} \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right) \right) \end{array} \right\}. \tag{15}$$

Next, we would like to provide some theoretical analysis to reveal the role of the graph in the optimizations/estimations. We will focus on Case II; a similar result can be obtained for Case I. For notation simplicity, we further simply (15) into

$$\hat{\mathbf{w}}_K^{\mathrm{II}} = \operatorname{argmin}_{\mathbf{w}_K} \left\{ \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 + \lambda_1 \|\mathbf{w}_K\|_1 \right. \\ \left. + \lambda_2 \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right)^T \mathbf{L} \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right) \right\} \tag{16}$$

by dropping the constant $Q \log(\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K)$ and reusing λ_2 to represent $\frac{\lambda_2}{\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K}$.

Theorem 4. Let $\hat{w}_{ik}^{II}, \hat{w}_{jk}^{II} \in \hat{\mathbf{w}}_K^{\mathrm{II}}$ be the estimated coefficients for predictors X_i and X_j. Let $\mathbf{x}_{iK}$ and $\mathbf{x}_{jK}$ be the data vectors for X_i and X_j, respectively. Suppose that $\hat{w}_{ik}^{II} \hat{w}_{jk}^{II} > 0$ and $a_{ij} \gg a_{uv}$. a_{uv} is the weight of any edge other than $X_i \sim X_j$. Then, for fixed λ_1 and λ_2 and a square-error loss,

$$\left|\left(\hat{w}_{ik}^{II} - \mu_{iK}\right) - \left(\hat{w}_{jk}^{II} - \mu_{jK}\right)\right| \leq \frac{\left\|\mathbf{x}_{iK} - \mathbf{x}_{jK}\right\|_2}{2\lambda_2} \\ \times \sqrt{\frac{\|\mathbf{y}_K\|_2^2}{a_{ij}^2} + \frac{2\lambda_2 \left(\mu_{iK} - \mu_{jK}\right)^2}{a_{ij}}}. \tag{17}$$

Please see proof in supplementary materials. In the upper bound in (17), the data for the new domain, $\mathbf{x}_{iK}$, $\mathbf{x}_{jK}$, and $\mathbf{y}_K$, knowledge transferred from the old domains, μ_{iK} and μ_{jK}, and λ_2 can be considered as given. Then, the upper bound is inversely related to the edge weight a_{ij}.

3.3.2 Jointly Estimating Hyperparameter $\mathbf{\Omega}$ *and Parameter* $\mathbf{W}$ *by an Efficient Alternating Algorithm.* $\mathbf{\Omega}$ is a hyperparameter that encodes the prior knowledge about the correlation structure between domains, which is difficult to specify precisely. Therefore, we choose to estimate $\mathbf{\Omega}$ and $\mathbf{W}$ together. This will change (14) to (18):

$$\text{Case III}: \left(\hat{\mathbf{W}}^{\mathrm{III}}, \hat{\Omega}^{\mathrm{III}}\right) = \underset{\mathbf{W}, \mathbf{\Omega}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k\|_2^2 \right. \\ \left. + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \left(Q \log |\mathbf{\Omega}| + \operatorname{tr}\left(\mathbf{L} \mathbf{W} \Omega^{-1} \mathbf{W}^T\right)\right)\right\}. \tag{18}$$

Equation (18) is the same as (14) except for treating $\mathbf{\Omega}$ as unknown.

Next, we will discuss an algorithm for solving (18). Equation (18) is not a convex optimization with respect to all unknown parameters. However, given $\mathbf{\Omega}$, it becomes a convex optimization with respect to $\mathbf{W}$, which can be solved efficiently. Furthermore, given $\mathbf{W}$, the optimization problem with respect to $\mathbf{\Omega}$ can be solved analytically, that is,

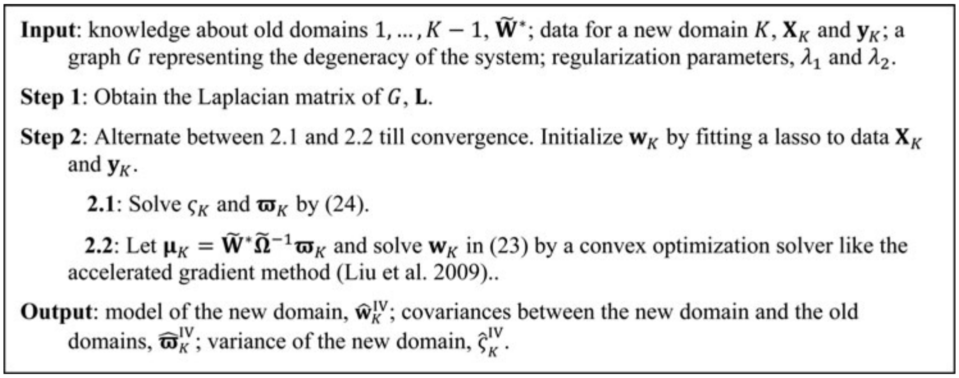
$$\hat{\Omega} = \frac{\mathbf{W}^T L \mathbf{W}}{Q}. \tag{19}$$

Therefore, we propose an iterative algorithm that alternates between two sub-optimizations: solving $\mathbf{W}$ with $\mathbf{\Omega}$ fixed at their estimates in the previous iteration, and solving $\mathbf{\Omega}$ with $\mathbf{W}$ fixed at its estimate just obtained. Because each sub-optimization decreases the objective function, this iterative algorithm is guaranteed to converge to a local optimal solution. Note that joint estimation of parameters and hyperparameters has also been adopted by other researchers (Idier 2010; Zhang and Yeung 2010).

A similar case to Case II (15) takes the form of (20):

$$\text{Case IV}: \left(\hat{\mathbf{w}}_K^{\mathrm{IV}}, \hat{\varsigma}_K^{\mathrm{IV}}, \hat{\boldsymbol{\varpi}}_K^{\mathrm{IV}}\right) \\ = \operatorname{argmin}_{\mathbf{w}_K, \varsigma_K, \boldsymbol{\varpi}_K} \left\{ \begin{array}{c} \|\mathbf{y}_K - \mathbf{X}_K \mathbf{w}_K\|_2^2 + \lambda_1 \|\mathbf{w}_K\|_1 \\ + \lambda_2 \left(\log \left(\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K \right) \right. \\ \left. + \dfrac{1}{\varsigma_K - \boldsymbol{\varpi}_K^T \tilde{\mathbf{\Omega}}^{-1} \boldsymbol{\varpi}_K} \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right)^T \mathbf{L} \left(\mathbf{w}_K - \boldsymbol{\mu}_K\right) \right) \end{array} \right\}. \tag{20}$$

Equation (20) can be solved by an iterative algorithm that alternates between solving $\mathbf{w}_K$ with ς_K and $\boldsymbol{\varpi}_K$ fixed—a convex optimization, and solving ς_K and $\boldsymbol{\varpi}_K$ with $\mathbf{w}_K$ fixed analytically



Input: knowledge about old domains $1, \ldots, K-1$, $\widetilde{\mathbf{W}}^*$; data for a new domain K, $\mathbf{X}_K$ and $\mathbf{y}_K$; a graph G representing the degeneracy of the system; regularization parameters, λ_1 and λ_2.

Step 1: Obtain the Laplacian matrix of G, $\mathbf{L}$.

Step 2: Alternate between 2.1 and 2.2 till convergence. Initialize $\mathbf{w}_K$ by fitting a lasso to data $\mathbf{X}_K$ and $\mathbf{y}_K$.

2.1: Solve ς_K and $\boldsymbol{\varpi}_K$ by (24).

2.2: Let $\boldsymbol{\mu}_K = \widetilde{\mathbf{W}}^* \widetilde{\boldsymbol{\Omega}}^{-1} \boldsymbol{\varpi}_K$ and solve $\mathbf{w}_K$ in (23) by a convex optimization solver like the accelerated gradient method (Liu et al. 2009)..

Output: model of the new domain, $\hat{\mathbf{w}}_K^{\mathrm{IV}}$; covariances between the new domain and the old domains, $\hat{\boldsymbol{\varpi}}_K^{\mathrm{IV}}$; variance of the new domain, $\hat{\varsigma}_K^{\mathrm{IV}}$.

Figure 2. An algorithm for solving the transfer learning formulation in Case IV.

using (21):

$$\hat{\boldsymbol{\varpi}}_K = \widetilde{\mathbf{W}}^T \mathbf{L} \mathbf{w}_K / Q, \;\; \hat{\varsigma}_K = \mathbf{w}_K^T \mathbf{L} \mathbf{w}_K / Q. \tag{21}$$

The derivation for (21) is in supplementary materials.

Finally, we present the algorithm for solving the proposed transfer learning formulation in Case IV in Figure 2 (Case III can be solved similarly). Note that the algorithm also works for classification problems that replace the square-error loss in Case III and Case IV, $||\mathbf{y}_k - \mathbf{X}_k \mathbf{w}_k||_2^2$, with a logistic loss, $\sum_{i=1}^{n_K} \log(1 + \exp(-y_{iK}\mathbf{x}_{iK}\mathbf{w}_K))$. Because this loss function is also convex with respect to $\mathbf{w}_K$, the convex optimization solver in step 2.2 of Figure 2 naturally applies. Step 2.1 does not involve the loss function so it needs no change.

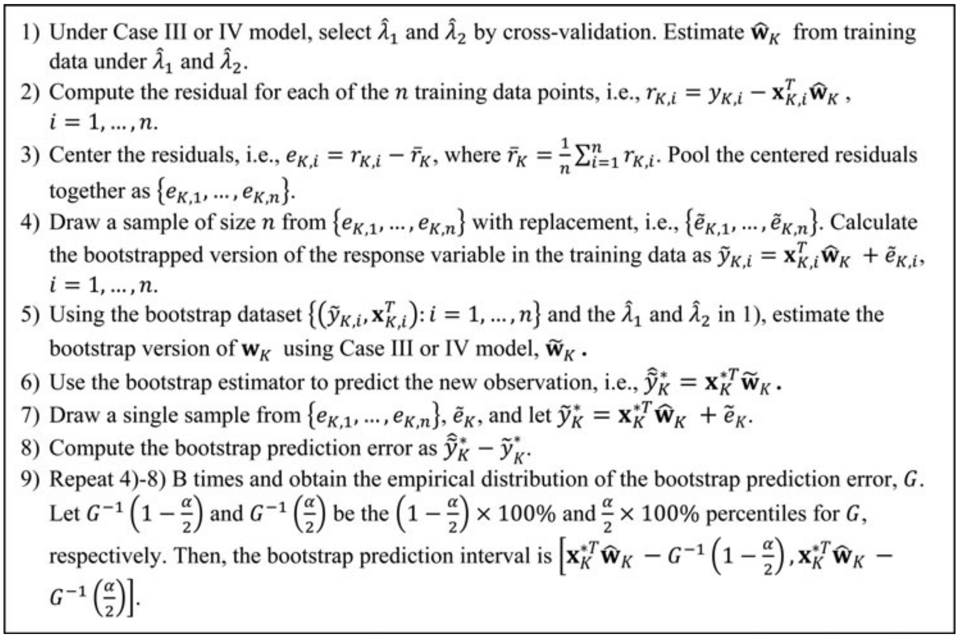
3.4 Prediction

Given a new observation in the new domain, $\mathbf{x}_K^*$, we can predict its response variable by $\hat{y}_K^* = \mathbf{x}_K^{*T} \hat{\mathbf{w}}_K$. $\hat{\mathbf{w}}_K$ can be the $\hat{\mathbf{w}}_K^{\mathrm{III}}$ in Case III or the $\hat{\mathbf{w}}_K^{\mathrm{IV}}$ in Case IV, obtained from training data. Because the proposed transfer learning method only produces a point estimator for $\mathbf{w}_K$, statistical inference on $\mathbf{w}_K$ and the prediction has to be performed using resampling approaches such as bootstrap. This is a similar situation to lasso, for which bootstrap-based statistical inference on the model coefficients has been studied by a number of articles (Knight and Fu 2000; Chatterjee and Lahiri 2010). Following the similar idea, we propose a residual bootstrap procedure to compute the prediction interval, which includes nine steps shown in Figure 3.

4. SIMULATION STUDIES

We conduct simulation studies to evaluate variable selection accuracy of the proposed method in terms of false positive rate (FPR) and false negative rate (FNR). FPR is the proportion of truly zero regression coefficients that are misidentified to be nonzero by the model. FNR is the proportion of truly nonzero regression coefficients that are misidentified to be zero by the model. In particular, we use area under the curve (AUC), which is an integrated measure for FPR and FNR.

Because the proposed method consists of two major aspects: transfer learning and degeneracy modeling, we would like to evaluate each aspect separately. In Section 4.1, we compare



1) Under Case III or IV model, select $\hat{\lambda}_1$ and $\hat{\lambda}_2$ by cross-validation. Estimate $\hat{\mathbf{w}}_K$ from training data under $\hat{\lambda}_1$ and $\hat{\lambda}_2$.
2) Compute the residual for each of the n training data points, i.e., $r_{K,i} = y_{K,i} - \mathbf{x}_{K,i}^T \hat{\mathbf{w}}_K$, $i = 1, \ldots, n$.
3) Center the residuals, i.e., $e_{K,i} = r_{K,i} - \bar{r}_K$, where $\bar{r}_K = \frac{1}{n}\sum_{i=1}^n r_{K,i}$. Pool the centered residuals together as $\{e_{K,1}, \ldots, e_{K,n}\}$.
4) Draw a sample of size n from $\{e_{K,1}, \ldots, e_{K,n}\}$ with replacement, i.e., $\{\tilde{e}_{K,1}, \ldots, \tilde{e}_{K,n}\}$. Calculate the bootstrapped version of the response variable in the training data as $\tilde{y}_{K,i} = \mathbf{x}_{K,i}^T \hat{\mathbf{w}}_K + \tilde{e}_{K,i}$, $i = 1, \ldots, n$.
5) Using the bootstrap dataset $\{(\tilde{y}_{K,i}, \mathbf{x}_{K,i}^T) : i = 1, \ldots, n\}$ and the $\hat{\lambda}_1$ and $\hat{\lambda}_2$ in 1), estimate the bootstrap version of $\mathbf{w}_K$ using Case III or IV model, $\tilde{\mathbf{w}}_K$.
6) Use the bootstrap estimator to predict the new observation, i.e., $\hat{\tilde{y}}_K^* = \mathbf{x}_K^{*T} \tilde{\mathbf{w}}_K$.
7) Draw a single sample from $\{e_{K,1}, \ldots, e_{K,n}\}$, $\tilde{e}_K$, and let $\tilde{y}_K^* = \mathbf{x}_K^{*T} \hat{\mathbf{w}}_K + \tilde{e}_K$.
8) Compute the bootstrap prediction error as $\hat{\tilde{y}}_K^* - \tilde{y}_K^*$.
9) Repeat 4)-8) B times and obtain the empirical distribution of the bootstrap prediction error, G. Let $G^{-1}\left(1 - \frac{\alpha}{2}\right)$ and $G^{-1}\left(\frac{\alpha}{2}\right)$ be the $\left(1 - \frac{\alpha}{2}\right) \times 100\%$ and $\frac{\alpha}{2} \times 100\%$ percentiles for G, respectively. Then, the bootstrap prediction interval is $\left[\mathbf{x}_K^{*T}\hat{\mathbf{w}}_K - G^{-1}\left(1 - \frac{\alpha}{2}\right), \mathbf{x}_K^{*T}\hat{\mathbf{w}}_K - G^{-1}\left(\frac{\alpha}{2}\right)\right]$.

Figure 3. A residual bootstrap procedure to compute prediction interval.

Table 1. AUC performances of transfer learning and single-domain learning

	Domain 1	Domain 2	Domain 3
Proposed transfer Learning			
Average	0.943447	0.955568	0.949735
Standard deviation	0.042415	0.037728	0.041266
Single-domain learning			
Average	0.871061	0.868485	0.889432
Standard deviation	0.099003	0.101911	0.084644

the proposed method in Case III with $\mathbf{L} = \mathbf{I}$, that is, having transfer learning but no degeneracy modeling. In Section 4.2, we compare the proposed method in Case III with $K = 1$, that is, having degeneracy modeling but for a single domain. Note that although we only present the results for Case III, similar results have been obtained for Case IV.

4.1 Comparison Between Transfer Learning and Single-Domain Learning

Consider three domains and the model, $Y_k = \sum_{q=1}^{Q} w_{qk} X_{qk} + \varepsilon_k$, $k = 1, 2, 3$. In each domain, there are 50 predictors, that is, $Q = 50$. Domains 1 and 2 are highly correlated with each other but little correlated with domain 3. To achieve this, we set the coefficients of the first five predictors in domains 1 and 2 to be nonzero, that is, $w_{qk} \neq 0, q = 1, \ldots, 5; k = 1, 2$. To make the two domains nonidentical, we randomly select one different predictor from X_6 to X_{50} in each domain to have a nonzero coefficient. For domain 3, we set the coefficients $w_{q3} \neq 0, q = 5, \ldots, 10$. Therefore, in each domain, there are six predictors with nonzero coefficients and all 44 others with zero coefficients. The value of each nonzero coefficient is randomly generated from a normal distribution $N(5, 1)$. After generating the coefficients, we check the correlation between the three domains using their respective coefficients. The correlations are 0.81 between domains 1 and 2, and 0.05(0.06) between domain 1(2) and 3, which are good to serve our purpose. Next, we generate samples for the 50 predictors from a multivariate normal distribution with zero mean and covariance matrix $\mathbf{\Sigma}_{ij} = 0.5^{|i-j|}, i, j = 1, \ldots, 50$. To focus on small-sample-size scenarios, 50 samples of the predictors are generated for each domain. The response variable of each sample is generated by the model $Y_k = \sum_{q=1}^{Q} w_{qk} X_{qk} + \varepsilon_k$, where ε_k is generated from $N(0, 15)$.

The proposed method of transfer learning is compared with single-domain learning, that is, a lasso model applied to each domain separately, on the simulation dataset. The process is repeated for 50 times; the average and standard derivation of the 50 AUCs for each method are reported in Table 1. It can be seen that transfer learning has a better average AUC performance than single-domain learning. It is also more stable by having a smaller standard deviation. Furthermore, having a little-correlated domain, that is, domain 3, does not hurt the performance of transfer learning in domains 1 and 2. This is because the proposed transfer learning method can estimate the correlation structure of the domains from data, and therefor can adaptively decide how much information to transfer from one domain to another.

4.2 Comparison Between Models With and Without Degeneracy Modeling

Consider a single domain and the model $Y = \sum_{q=1}^{Q} w_q X_q + \varepsilon$ with 50 predictors, that is, $Q = 50$. Suppose that the 50 predictors fall into 10 nonoverlapping subsets; each subset consists of five predictors as its degenerate elements. Coefficients of the first two subsets, $\{w_1, w_2, w_3, w_4, w_5\}$ and $\{w_6, w_7, w_8, w_9, w_{10}\}$ are nonzero and generated from $N(5, 1)$ and $N(1, 1)$, respectively. Coefficients of the rest three subsets are zero. This is to reflect the reality that some degenerate elements of the system may not relate to the particular response of interest. Next, we want to generate samples for the 50 predictors. The way these samples are generated must follow the biology of how the degenerate elements are formed, so it is different from Section 4.1. Specifically, assuming that the 10 subsets correspond to 10 TFs, we first generate 10 TFs, $\mathrm{TF}_1, \ldots, \mathrm{TF}_{10}$, from $N(0, 1)$. Next, to reflect the stochastic nature of the degenerate elements corresponding to each TF_i, we generate TF_i's corresponding five predictors/degenerate elements from $N\left(\rho \times \mathrm{TF}_i, 1 - \rho^2\right)$. ρ corresponds to the correlation between TF_i and its corresponding degenerate elements. We try different correlation levels for generality. Fifty samples are generated for each correlation level.

To apply the proposed method, we first build a graph that puts an edge between each pair of predictors in each of the five subsets (no edge between the subsets) to represent the qualitative prior knowledge about the degeneracy. The edge weight is set to be one. The graph is then converted to a Laplacian matrix $\mathbf{L}$ and used in the proposed method. A lasso model is also applied to the simulation datasets as a model not taking the degeneracy into account. The process is repeated for 50 times. The average AUC performances of the two methods are comparable. However, when the best AUC performances of the two methods are compared, the proposed method is significantly better, as can be seen in Table 2.

Table 2. Best AUC performances of proposed model considering degeneracy and lasso

ρ	Proposed model considering degeneracy	Lasso (no consideration of degeneracy)
0.6	0.8138	0.6963
0.7	0.8475	0.6875
0.8	0.8388	0.6888

5. APPLICATION

We present an application of modeling the predictive relationship between TF binding sites and gene expression. Eight human cell lines (H1, K562, GM12878, HUVEC, HSMM, NHLF, NHEF, and HMEC) are considered as eight domains. Since the simulation studies presented the results for Case III, here we present the results for Case IV. To apply the model in Case IV, one cell line is treated as the new domain and all the others are treated as the old domains. The data for the predictors are obtained as follows: We download the RefSeq Gene annotation track for human genome sequence (hg19) from the University of California Santa Cruz Genome Browser (USCS, *http://genome.ucsc.edu*). Then, we scan the promoter region of each gene (i.e., 1000 bp upstream of the transcription state site) and count the occurrence of each κ-mer. Recall that a κ-mer is a κ-letter-long word describing a potential binding site. We do this for $\kappa = 6$ and obtain data for 4^6 predictors, and for $\kappa = 7$ and obtain data for 4^7 predictors. $\kappa = 6, 7$ are common choices for binding site studies (Li et al. 2010). A minor technical detail is that in human cell lines, a word and its reverse complement should be considered the same predictor. This reduces the 6-mer predictors to 2080 and 7-mer predictors to 8192. Furthermore, we obtain data for the response variable, that is, gene expression, for the eight cell lines from the Gene Expression Omnibus (GEO) database under the accession number GSE26386. A total of 16,324 genes on all chromosomes are included. This is the sample size.

Recall in Section 3.3.1, we mentioned that a graph can be constructed to represent the prior knowledge about the degeneracy. Nodes are predictors, that is, κ-mers. The similarity between two κ-mers is $\kappa - H\{X_i, X_j\}$. $H\{X_i, X_j\}$ is the Hamming distance. We consider an unweighted graph here, that is, there is an edge between X_i and X_j, if $\kappa - H\{X_i, X_j\} \geq s$; there is no edge between X_i and X_j otherwise. s is a tuning parameter in our method.

5.1 Comparison to Methods Without Transfer Learning or Without Degeneracy Modeling

The method without degeneracy modeling is the model in Case IV but with $\mathbf{L} = \mathbf{I}$. The method without transfer learning is a lasso model applied to data of the new domain alone. Each method has some tuning parameters to select. For example, the tuning parameters for the proposed method include λ_1, λ_2, and s. We find that $s = 5$ is a consistently good choice across different choices for λ_1 and λ_2. λ_1 and λ_2 can be selected based on model selection criteria such as BIC and Akaike information criterion (AIC). However, each criterion has some known weakness and there is no such criterion that works universally well under all situations. To avoid drawing biased conclusion, we do not stick to any single model selection criterion. Instead, we run the model on a wide range of values for λ_1 and λ_2, that is, λ_1, $\lambda_2 \in [10^{-5}, 10^3]$, and report the average performance. Similar strategies are adopted for the two competing methods. This is a common practice for comparison of different methods each of which has parameters to be tuned.

All 2080 6-mers are used as predictors. To compare the three methods in challenging predictive problems, that is, problems with small sample sizes, only the 1717 genes on chromosome 1 are included. Furthermore, one cell line is treated as the new domain and all the other cell lines are treated as the old domains. The knowledge of the old domains, that is, $\tilde{\mathbf{W}}^*$, is obtained using the model in Case III applied to the data of the old domains. The data of the new domain are divided into 10-folds. Nine-folds of data are used, together with $\tilde{\mathbf{W}}^*$, to train a model, and the model is applied to the remaining one-fold to compute a performance metric such as the MSE. The average MSE, $\overline{\mathrm{MSE}}$, over the 10-folds is computed. This entire procedure is repeated for each of the eight cell lines as the new domain and the eight $\overline{\mathrm{MSE}}s$ are averaged to get $\overset{\equiv}{\mathrm{MSE}}$. This $\overset{\equiv}{\mathrm{MSE}}$ can be obtained for each pair of λ_1 and λ_2 in their range $[10^{-5}, 10^3]$. Averaging the $\overset{\equiv}{\mathrm{MSE}}$ s over the range gives $\overline{\overset{\equiv}{\mathrm{MSE}}}$. Table 3 shows the results of comparison. It is clear that both transfer learning and degeneracy modeling in the proposed method help prediction in the new domain. Transfer learning is crucially important, without which the prediction is significantly impaired.

Table 3. Comparison of three methods by $\overline{\overset{\equiv}{\mathrm{MSE}}}$

	Proposed method versus transfer learning without consideration of degeneracy	Proposed method versus lasso (no transfer learning)
$\frac{\overline{\overset{\equiv}{\mathrm{MSE}}}(\text{competing}) - \overline{\overset{\equiv}{\mathrm{MSE}}}(\text{proposed})}{\overline{\overset{\equiv}{\mathrm{MSE}}}(\text{proposed})} \times 100\ \%$	16.25%	920.44%

5.2 Robustness of the Proposed Method to Noisy Old Domains

One distinguished feature of the proposed method is the ability to learn the relationship between each old domain and the new domain from data, and adaptively decide how much knowledge to transfer from each old domain. To test this, we can include some "noisy" old domains. If the proposed method has the ability it claims to have, it should transfer little knowledge from the noisy domains and its performance should not be affected much. Specifically, we create the noisy old domains by destroying the correspondence between the response and predictors of each gene in these domains through shuffling. We compare the estimated model coefficients of the new domain and those obtained by keeping all the old domains as they are (i.e., no shuffling) by calculating their correlation coefficient. Table 4 shows this correlation coefficient with four, five, and six old domains shuffled. Cell line GM12878 is the new domain. When applying the proposed method, λ_1 and λ_2 are selected by 10-fold

Table 4. Correlation between model coefficients of the new domain with and without shuffled old domains

Four out of seven old domains are shuffled	Five out of seven old domains are shuffled	Six out of seven old domains are shuffled
0.998065	0.816133	0.763273

Table 5. Comparison between transfer learning with shuffled noisy old domains and single-domain learning

	Four out of seven old domains are shuffled	Five out of seven old domains are shuffled	Six out of seven old domains are shuffled
$\frac{\overset{\equiv}{\mathrm{MSE}}(\mathrm{lasso})-\overset{\equiv}{\mathrm{MSE}}(\mathrm{transferlearning})}{\overset{\equiv}{\mathrm{MSE}}(\mathrm{transferlearning})}$ $\times$ 100 %	22.42%	20.26%	19.95%

cross-validation. It can be seen that the proposed method is almost not affected when less than five out of seven old domains are noisy domains. Furthermore, we also compute the correlation between the model coefficients of the new domain with and without transfer learning (no shuffling) and this correlation is 0.793765, which is at the similar level to that when there are more than five noisy domains. Finally, we would like to know if transfer learning can still outperform single-domain learning (i.e., lasso for the new domain) even with knowledge transferred from noisy domains. This result is summarized in Table 5, which further demonstrates the robustness of the proposed method.

5.3 Understanding the Degenerate System

The purpose of predictive modeling is not only to predict a response but also to facilitate understanding of the problem domain. To achieve this, we apply the proposed method to one cell line, GM12878, treating this cell line as the new domain and all other cell lines as the old domains. Predictors are all 8192 7-mers. 7-mers contain richer binding site information than 6-mers, but analysis of 7-mers has been limited because of the dimension. Focusing on 7-mers can also test the capability of our method in handling very large dimensional predictors. The response is a binary indicator variable that indicates if a gene is expressed or unexpressed, so a logistic loss function is used in our method. This has a purpose of testing the capability of our method in classification problems. Also, it is more reasonable to assume that binding site counts like 7-mers can explain a majority of the variability in expressed/unexpressed genes than the variability in the numerical gene expression levels. The latter is more involved, as the expression level is affected by a variety of other factors than binding site counts. 16,324 genes on all chromosomes are included in the analysis.

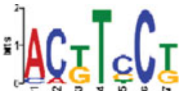
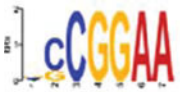
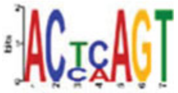
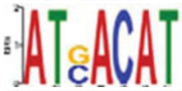
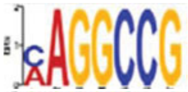
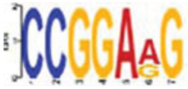
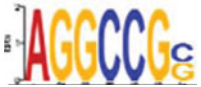
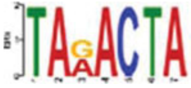
Unlike Section 5.1 in which comparison of prediction accuracy between methods is the primary goal, here we want to obtain a model for the 7-mer-gene-expression relationship, and based on the identified relationship, to understand the system better. For this purpose, model selection is unavoidable. We use 10-fold cross-validation to choose the optimal λ_1 and λ_2, which are ones giving the smallest average classification error over the 10-folds. The true positive rate (TPR), true negative rate (TNR), and accuracy of our method are 0.84, 0.60, and 0.70, respectively. The definition of TPR is: among all the genes classified as expressed, the proportion that is truly expressed. TNR is: among all the genes classified as unexpressed, the proportion that is truly unexpressed. Accuracy is the proportion of correctly classified genes. An observation is that TPR is higher than TNR, which is expected, because classification of unexpressed genes is supposed to be harder than expressed genes. The accuracy is 0.70, which is satisfactory in this application, considering the complexity of the biological system. Given satisfactory accuracy, we can now proceed and use the model for knowledge discovery. To do this, we use all the data of GM12878 to fit a model under the optimal λ_1 and λ_2, which is called "the model" in the subsequent discussion.

In knowledge discovery, our goal is to characterize the degeneracy of the new domain, that is, GM12878. Note that although we have used a graph to encode the degeneracy, it is before seeing any data and is only qualitative. It can now be better characterized by the model that incorporates both the graph and the data of the new domain as well as knowledge transferred from the old domains. Specifically, the following steps are performed:

First, we examine the estimated coefficients of the 7-mers and eliminate those 7-mers with zero coefficients from the graph. These 7-mers are considered not significantly affecting gene expression. Then, we rank the remaining 7-mers according to the magnitudes of their coefficients and choose the top 50 7-mers for the subsequent analysis. This helps us focus on identifying the degeneracy most relevant to gene expression. Some of the 50 7-mers are connected in the graph and some are not; in fact, they fall into different clusters. We define a cluster to be a group of 7-mers, each of which is connected with at least one other 7-mer in the group. The clusters are shown in Table 6. Each cluster is suspected to correspond to a TF and the 7-mers in the cluster are believed to be alternative binding sites of the TF. To verify this, we compute a position specific scoring matrix (PSSM) for each cluster. PSSM has been commonly used to characterize binding site uncertainty (Li et al. 2010). A PSSM is a $\kappa \times 4$ matrix. κ is the number of positions in a κ-mer. $\kappa = 7$ in our case. Each row of a PSSM is a probability distribution over $\{A, C, G, T\}$. Let $p_i(s)$ denote the probability of s, $s = \{A, C, G, T\}$, for row/position i, $i = 1, \ldots, \kappa$. $\sum_{s=\{A,C,G,T\}} p_i(s) = 1$. $p_i(s)$ can be calculated by $p_i(s) = \frac{n_i(s)}{C}$, where C is the cluster size and $n_i(s)$ is the number of occurrences of s at position i among all the 7-mers in the cluster. Because our model outputs an estimated coefficient for each 7-mer, we modify this conventional formula by $p_i(s) = \frac{\sum_{c=1}^{C} \hat{w}_c I(\mathbf{r}_{ci}=s)}{\sum_{c=1}^{C} \hat{w}_c}$. $\hat{w}_c$ is the estimated coefficient for the cth 7-mer in the cluster. $\mathbf{r}_{ci}$ is the letter at the ith position of the cth 7-mer. This modified formula works better because it takes the response variable into consideration by incorporating the model coefficients. A PSSM can be represented in a compact form by a motif logo, which stacks up the four letters $\{A, C, G, T\}$ at each position i and the letter height is proportional to its probability $p_i(s)$. Please see Table 6 for the PSSM motif logos for all the clusters.

Furthermore, the PSSM of each cluster can be compared with databases of known TFs to see if there is a

Table 6. Clusters of 7-mers and matching with known TFs for GM12878

Clusters	7-mers	Est. coeff.	Motif logo	Matched known TFs
1	AAGTGCT ACGTGCT ACGTTCT ACGTCCT ACTTCCT ACTTCCG CCTTCCG	0.005808 0.005497 0.005367 0.005963 0.009086 0.010709 0.005685		SPI1, Ets, Elk-1, FLI1, FEV
2	GGCGGAA GCCGGAA ACCGGAA CCCGGAA TCCGGAA	0.007632 0.006837 0.006497 0.006982 0.005488		GABP, Elk-1, Ehf_primary, Eip74EF, ERF
3	ACTAAGT ACTCAGT ACCCAGT	0.005597 0.005881 0.006013		AP-1, NF-E2
4	ATGACAT ATCACAT	−0.00594 −0.00588		N/A
5	CAGGCCG AAGGCCG	0.006636 0.00586		Zfx, CNOT3
6	CCGGAAG CCGGAGG	0.009778 0.005296		ELK-1, GABPA, Eip74EF, SAP-1a, EHF
7	AGGCCGC AGGCCGG	0.005775 0.005715		Zfx
8	TAGACTA TAAACTA	0.006607 0.006447		N/A

match. We used the Motif-based Sequence Analysis Tools (*http://meme.nbcr.net/meme*) for the matching. Table 6 shows the top five matched TFs for each cluster, according to the significance level of each match. If less than five matched TFs are found, then all the matched TFs will be shown. If no match is found, there is an "N/A." Out of the eight clusters, six have at least one match with known TFs. Clusters 1, 2, and 6 are enriched with SPI1, Ets, Elk, FLI1, FEV, GABP, and EHF, which are well-known TFs for important basic cell functions. Cluster 3 is enriched with AP-1 and NF-E2, which are related to Golgi membrane and nucleus that are also basic cell functions. Clusters 5 and 7 are enriched with Zfx and CNOT3. CNOT3 is a Leukocyte Receptor Cluster Member 2 and Zfx is required for the renewal process in hematopoietic cells. As GM12878 is a lymphocyte cell, these blood transcription factors are specific to this cell line. Clusters 4 and 5 do not match with any known TFs. However, only 10%–20% of total human TFs are known so far. The unmatched clusters indeed present an interesting opportunity for identifying new TFs.

This entire analysis for GM12878 is also performed for other cell lines. For each cell line, clusters of 7-mers exist and a large majority of the clusters can be matched to known TFs. Also, some clusters are common across the cell lines. These are the clusters whose matched TFs are related to basic cell functions. There are also some cell-line-specific clusters such as clusters 5 and 7 for GM12878. As other examples, there is a cluster enriched with CTF1 for HMEC. CTF1 is known to be in entracellar region. As HMEC is an epithelial cell, CTF1 is specific to this cell line. In addition, there is a cluster enriched with MyoD and another cluster enriched with MEF-2 for HSMM. MyoD is related to muscle cell differentiation and MEF-2 is a myocyte enhancer factor, both being specific to HSMM. The identified common and cell-line-specific cluster structures verifies transfer learning's ability of modeling related but not exactly the same domains.

6. CONCLUSION

In this article, we developed a transfer learning method for predictive modeling of degenerate biological systems under the Bayesian framework. Theoretical properties of the proposed method were investigated. Simulation studies showed better AUC performance of the proposed method compared with competing methods. A real-world application was presented, which modeled the predictive relationship between TF binding site counts and gene expression. The proposed method showed good accuracy and robustness to noisy old domains, and discovered interesting degenerate mechanisms of the system.

There are several potential future directions for this work. First, the proposed method was formulated under a Bayesian framework but solved from an optimization point of view to gain efficiency. A Bayesian estimation approach such as empirical Bayes and hierarchical Bayes could allow better characterization of the uncertainty. Second, a similar approach may be developed for predictive modeling of nonlinear relationships. Third, future engineering system design may adopt biological principles like degeneracy to be more robust and adaptive to unpredictable environmental situations. By that time, it will be very interesting to study how to migrate the proposed approach to engineering systems.

SUPPLEMENTARY MATERIALS

The supplementary materials include derivations for (6) and (24), proofs of Theorems 1–4, and matlab code for Section 4.

ACKNOWLEDGMENTS

This work is partly supported by NSF Grant 1149602 and NIH Grant GM096194-01.

[Received November 2013. Revised April 2015.]

REFERENCES

Argyriou, A., Pontil, M., Ying, Y., and Micchelli, C. A. (2007), "A Spectral Regularization Framework for Multi-Task Structure Learning," *Advances in Neural Information Processing Systems*, 20, 25–32. [364]

Bonilla, E., Chai, K. M., and Williams, C. (2007), "Multi-Task Gaussian Process Prediction," *Advances in Neural Information Processing Systems*, 20, 153–160 [364]

Caruana, R. (1997), "Multitask Learning," *Machine Learning*, 28, 41–75. [364]

Chatterjee, A., and Lahiri, S. N. (2010), "Asymptotic Properties of the Residual Bootstrap for Lasso Estimators," *Proceedings of the American Mathematical Society*, 138, 4497–4509. [368]

Chung, F. R. (1997), *Spectral Graph Theory* (Vol. 92), Providence, RI: AMS Bookstore. [367]

Dai, W., Yang, Q., Xue, G. R., and Yu, Y. (2007), "Boosting for Transfer Learning," in *Proceedings of the 24th International Conference on Machine Llearning*, ACM, pp. 193–200. [364]

Edelman, G. M., and Gally, J. A. (2001), "Degeneracy and Complexity in Biological Systems," *Proceedings of the National Academy of Sciences*, 98, 13763–13768. [363]

Evgeniou, T., and Pontil, M. (2004), "Regularized Multi-Task Learning," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 109–117. [364]

——— (2007), "Multi-Task Feature Learning," *Advances in Neural Information Processing Systems*, 19, 41. [364]

Genkin, A., Lewis, D. D., and Madigan, D. (2007), "Large-Scale Bayesian Logistic Regression for Text Categorization," *Technometrics*, 49, 291–304. [365]

Huang, S., Li, J., Ye, J., Fleisher, A., Chen, K., Wu, T., and Reiman, E. (2013), "A Sparse Structure Learning Algorithm for Bayesian Network Identification from High-Dimensional Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1328–1342. [363]

Idier, J. (ed.). (2010), *Bayesian Approach to Inverse Problems*, Somerset, NJ: Wiley. [367]

Jebara, T. (2004), "Multi-Task Feature and Kernel Selection for SVMs," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ACM, p. 55. [364]

Knight, K., and Fu, W. (2000), "Asymptotics for the Lassotype Estimators," *The Annals of Statistics*, 28, 1356–1378. [368]

Lawrence, N. D., and Platt, J. C. (2004), "Learning to Learn With the Informative Vector Machine," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ACM, p. 65. [364]

Li, F., and Zhang, N. R. (2010), "Bayesian Variable Selection in Structured High-dimensional Covariate Spaces With Applications in Genomics," *Journal of the American Statistical Association*, 105, 1202–1214. [363,367]

Li, X., Panea, C., Wiggins, C. H., Reinke, V., and Leslie, C. (2010), "Learning "graph-mer" Motifs that Predict Gene Expression Trajectories in Development," *PLoS Computational Biology*, 6, e1000761. [366,370,371]

Liu, J., Ji, S., and Ye, J. (2009), "Multi-Task Feature Learning via Efficient L21-Norm Minimization," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 339–348. [364,365]

Rückert, U., and Kramer, S. (2008), "Kernel-based Inductive Transfer," in *Machine Learning and Knowledge Discovery in Databases*, Berlin: Springer, pp. 220–233. [364]

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society,* Series B, 58, 267–288. [364,365]

Vogelstein, B., and Kinzler, K. W. (2004), "Cancer Genes and the Pathways they Control," *Nature Medicine*, 10, 789–799. [363]

Zhang, Y., and Yeung, D. Y. (2010). "A Convex Formulation for Learning Task Relationships in Multi-Task Learning," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 733–742. [367]