A Comprehensive Survey on Transfer Learning

Authors: Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu,

Hengshu Zhu, Senior Member, IEEE, Hui Xiong, Fellow, IEEE, and Qing He

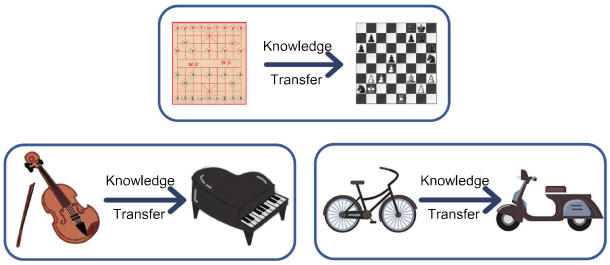
Harvard AC295/CS115

SUMMARY BY: EDUARDO PEYNETTI, JESSICA WIJAYA, ROHIT BERI, STUART NEILSON

Introduction

According to psychologist C.H. Judd, learning to transfer is the result of the generalization of experience.

- Prerequisite is that there needs to be a connection between two learning activities.



Negative Transfer

- Transfer learning (TL) doesn't always bring positive impact on new tasks
  - Learning to ride bicycle cannot help us learn piano faster
- Negative Transfer depend on relevance between source and target domains and the learner's capacity to find the transferable and beneficial parts of the knowledge across the domains
- Target learner is negatively affected by the transferred knowledge
  - Learning Spanish can make it difficult to learn French though the languages share a lot in common, as previous learning interferes with learning word formation, usage, pronunciation, etc. in French.

Homogeneous Transfer

- This paper focused on Homogeneous Transfer Learning
- Domains are in the same feature space
- Differ only in marginal distributions - dealt by correcting sample selection bias or covariate shift.
- This assumption doesn't hold in some cases, i.e. word may have different meaning in different domains i.e. context feature bias – Adapt conditional distributions

Heterogeneous Transfer

- Knowledge transfer where domains have different feature space
- Requires feature space adaptation in addition to distribution adaptation
- More complicated
- Not a focus for this paper
- Also not covered are Reinforcement Transfer Learning, Lifelong Transfer Learning & Online Transfer Learning

Aim of the Survey

- Over 40 Representative Transfer Learning approached are summarized
- Experiments are conducted to compare over 20 different approaches

Related Areas

Areas related to TL are:

Semi-Supervised Learning (SSL)

- Combines abundant un-labeled with a limited number of labeled instances to train a learner
- Relaxes the dependence on labeled instances thereby reducing labeling costs
- Both instances are drawn from same distribution
- In contrasts, the distributions of source and target domains are different in Transfer Learning
- Key assumptions of smoothness, cluster, and manifold hold both in case of semi-supervised and transfer learning
- Many a times TL absorbs the technology of SSL

Multi-View Learning (MVL)

- MVL focuses on ML for multi-view data – Object is described from multiple views
- Example: Video Object with image signal and audio signal
- Learning can be improved by considering information from all the available views
- Strategies include – Subspace Learning, Multi-kernel learning, and co-training
- Approaches are also adopted in TL – Zhang et al. proposed a multi-view TL framework which imposes the consistency among multiple views
- Yang and Gao – Multi-view information across different domains for knowledge transfer
- Feuz and Cook – Multi-view TL for activity learning: Knowledge transfer between heterogenous sensor platform

Multi-Task Learning (MTL)

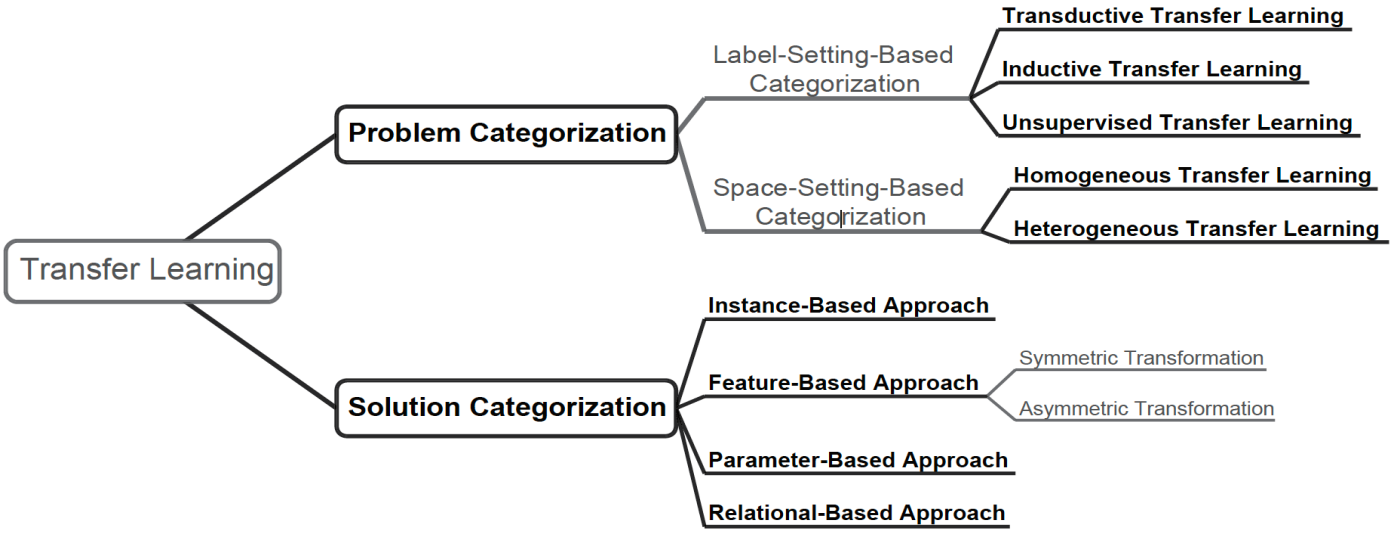
- Jointly learn a group of related tasks
- Reinforces each task by taking advantage of interconnections
- Considers inter-task relevance and inter-task difference – enhances generalization
- MTL vs TL: MTL pays equal attention to each task; TL pays more attention to Target task
- Zhang et al. employs MTL and TL for biological image analysis
- Liu et al. proposes a framework for human-action recognition based on MTL and TL

Overview

Definitions

- Domain (***D***): comprises of feature space ***X*** and a marginal distribution ***P(X)***
- Task (***T***): consists of label space ***Y*** and a decision function ***f*** to be learned from the data
- Transfer Learning (**TL**): Utilized knowledge implied in source domain to improve performance of the learned decision function f^T on the target domain
- Domain Adaptation: Process of adapting one or more source domains to transfer knowledge and improve the performance of the target learner

Categorization of Transfer Learning



Problem Categories

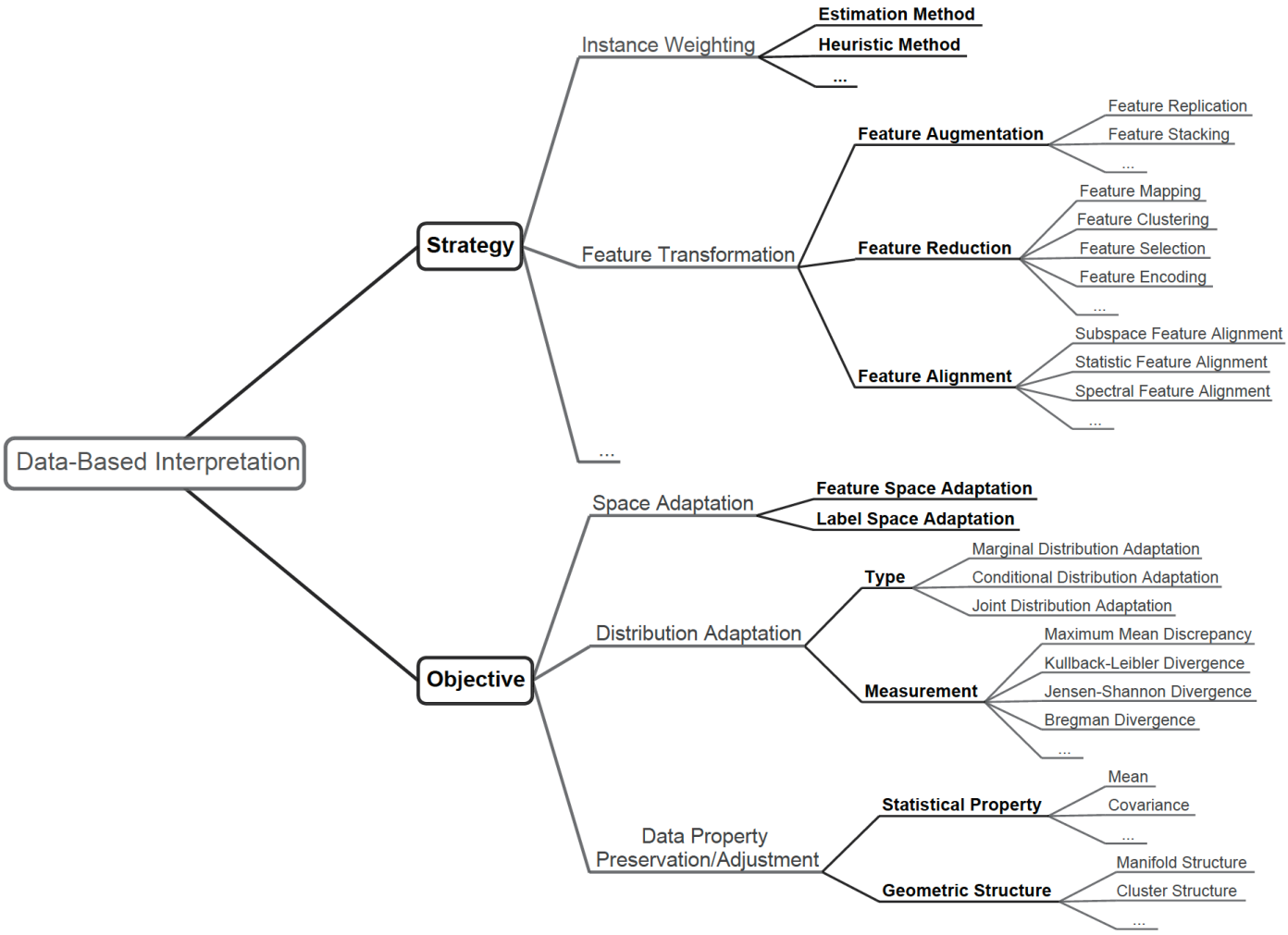
- *Based on Label Information*
  - Transductive TL – Only source domain has label information
  - Inductive TL – Both source and target domain have label information
  - Unsupervised TL – Neither source nor target domain have label information
- *Based on Feature Space and Label Space*
  - Homogeneous TL – Both Feature Space and Label space of source and target domains are similar
  - Heterogenous TL – Either Feature Space or Label Space or both not similar for source and target domains

Solution Categories

- *Instance Based*
  - Instance weighting strategy
- *Feature Based*
  - Transforms the original features to create new feature representation
  - Symmetric Transformation – Attempts to find common feature latent space and then transform both the source and the target
  - Asymmetric Transformation – Transforms source features to match target ones
- *Parameter Based*
  - Transfer of knowledge at model/parameter level
- *Relational Based*
  - Focus on problems in relational domain
  - Transfer of logical relationship or rules learned
  - This survey does not cover Relational-based approaches

Data-based Interpretation

Broadly uses instance based and feature based transfer learning. Focused on transferring the knowledge via the adjustment and the transformation of the data.



Strategies

- Instance Weighting
- Feature Transformation

Objectives

- Space Adaptation – Mostly required in Heterogenous TL – Not a focus for this paper
- Distribution Adaptation – Main objective in case of Homogeneous TL: To reduce the distribution difference between the source and the target domain instances
- Data Property Preservation/Adjustment – Certain advanced approaches and use cases

Instance Weighting Strategy

- Large number of labeled source and a limited number of target domain instances are available
- Domains differ in only marginal distributions i.e. $P^s(X) \neq P^t(X)$ but $P^s(Y|X) = P^t(Y|X)$
- Adapting the marginal distributions by assigning weights to the source instances in the loss function
- Weighting strategy is based on the following equation and objective function:

$$\mathbb{E}_{(\mathbf{x},y)\sim P^T}\left[\mathcal{L}(\mathbf{x},y;f)\right] = \mathbb{E}_{(\mathbf{x},y)\sim P^S}\left[\frac{P^T(\mathbf{x},y)}{P^S(\mathbf{x},y)}\mathcal{L}(\mathbf{x},y;f)\right] = \mathbb{E}_{(\mathbf{x},y)\sim P^S}\left[\frac{P^T(\mathbf{x})}{P^S(\mathbf{x})}\mathcal{L}(\mathbf{x},y;f)\right].$$

$$\min_f \frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i\mathcal{L}\left(f(\mathbf{x}_i^S), y_i^S\right) + \Omega(f),$$

where β_i $(i = 1, 2, \cdots, n^S)$ is the weighting parameter. The theoretical value of β_i is equal to $P^T(\mathbf{x}_i)/P^S(\mathbf{x}_i)$.

- However, β_i's are difficult to compute.

Kernel Mean Matching (KMM): Huang et al.

$$\arg\min_{\beta_i\in[0,B]}\left\|\frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i\Phi(\mathbf{x}_i^S) - \frac{1}{n^T}\sum_{j=1}^{n^T}\Phi(\mathbf{x}_j^T)\right\|_{\mathcal{H}}^2$$

$$s.t.\ \left|\frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i - 1\right| \leq \delta,$$

- Estimates β_i's by matching means between the source and target domain instances in a Reproducing Kernel Hilbert Space (RKHS).
- δ is a small parameter, and B is a parameter for constraint

2-Stage Weighting Framework for Multi-Source Domain Adaptation (2SW-MDA): Sun et al.

- Stage 1: Instance Weighting – similar to KMM
- Stage 2: Domain Weighting – Weights are assigned to each domain for reducing conditional distribution difference based on smoothness assumption
- Source domain instances are reweighted using instance weights and domain weights
- Reweighted instances along with label target domain instances are used to train the classifier

Kullback-Leibler Importance Estimation Procedure (KLIEP): Sugiyama et al.

- Depends on minimizing KL divergence
- Incorporates built-in model selection procedure

TrAdaBoost – Bosting for Transfer Learning: Dai et al.

- Adjust weights iteratively – extends AdaBoost to transfer learning scenario
- Mechanism to decrease weights on instances that have negative effects on the target learner
- The labeled source and target domains are combined to train the weak learner, but the weighting operations are different for the source-domain and the target-domain instances
- Final classifier combines and ensembles half of newly minted weak classifier through voting scheme
- Different weighting operations for source and target domains

$$\beta_{k,i}^S = \beta_{k-1,i}^S(1+\sqrt{2\ln n^S/N})^{-|f_k(\mathbf{x}_i^S)-y_i^S|}\ (i = 1, \cdots, n^S),$$
$$\beta_{k,j}^T = \beta_{k-1,j}^T(\bar{\delta}_k/(1-\bar{\delta}_k))^{-|f_k(\mathbf{x}_j^T)-y_j^T|}\ (j = 1, \cdots, n^T).$$

Multi-Source TrAdaBoost (MsTrAdaBoost): Yao and Doretto

- Two steps in each iteration
- Step 1: Candidate Classifier Construction
- Step 2: Instance Weighting

Others

- TaskTrAdaBoost – Parameter based algorithm
- General Weighting Framework (Heuristic Method) by Jiang & Zhai: Minimize cross-entropy loss of:

- Labeled Target Instance, Unlabeled Target Instance, and Labeled Source Instance

Feature Transformation Strategy

- Often adopted in feature-based approaches and consists of several operations
- Example – cross domain text classification problem
- Find latent features through transformation use them as bridge for knowledge transfer
- Objective of constructing latent space include:
  - minimizing marginal and conditional distribution difference (**primary objective**),
  - preserving properties/potential structures of the data,
  - finding correspondence between features
- Three types of feature transformation:
  - Feature Augmentation
    - Feature Replication
    - Feature Stacking
  - Feature Reduction
    - Feature Mapping
    - Feature Clustering
    - Feature Selection
    - Feature Encoding
  - Feature Alignment

Distribution Difference Metric

- How to measure distribution difference or similarity is an important issue - Commonly used metrics:
  - Maximum Mean Discrepancy (MMD): **Widely used in Transfer Learning**

$$\mathrm{MMD}(X^S, X^T) = \left\| \frac{1}{n^S} \sum_{i=1}^{n^S} \Phi(\mathbf{x}_i^S) - \frac{1}{n^T} \sum_{j=1}^{n^T} \Phi(\mathbf{x}_j^T) \right\|_{\mathcal{H}}^2$$

  - Kullback-Leibler Divergence
  - Jensen-Shannon Divergence
  - Bregman Divergence
  - Hisbert-Schmidt Independence Criterion
  - Others:
    - Wasserstien Distance, Central Moment Discrepancy
    - Munti-Kernel Maximum Mean Discrepancy (MK-MMD) by Gretton et al.
    - Weighted version of MMD by Yan et al. - attempts to address class weight bias

Feature Augmentation

- Widely used, particularly in symmetric feature-based approaches
- Feature Replication: Feature Augmentation Method (FAM) by Daumé – a simple feature replication
  - Feature space augmented to three times its size – general features, source-specific features, target-specific features
  - For transformed source-domain, target-specific features are set to Zero & vice-versa

$$\Phi_S(\mathbf{x}_i^S) = \langle \mathbf{x}_i^S, \mathbf{x}_i^S, \mathbf{0} \rangle, \ \Phi_T(\mathbf{x}_j^T) = \langle \mathbf{x}_j^T, \mathbf{0}, \mathbf{x}_j^T \rangle,$$

$$\text{where } \Phi_S \text{ and } \Phi_T \text{ denote the mappings to the new feature}$$

  - Final classifier is trained on transformed labeled instances
  - Generalizes well to multi-source scenarios thought at the cost of redundancy
  - Utilizes unlabeled instances to further facilitate the knowledge transfer
- Feature Stacking: FAM may not work well with Heterogenous TL tasks
  - Padding zero vectors and directly replicating features as in FAM less is less effective when source and target domains have different feature representations
  - Heterogenous Feature Augmentation (HFA) by Li et al. maps original features into a common features space and then performs a feature stacking operation

$$\Phi_S(\mathbf{x}_i^S) = \langle W^S \mathbf{x}_i^S, \mathbf{x}_i^S, \mathbf{0}^T \rangle, \ \Phi_T(\mathbf{x}_j^T) = \langle W^T \mathbf{x}_j^T, \mathbf{0}^S, \mathbf{x}_j^T \rangle,$$

Feature Mapping

- In traditional ML, there are many feasible mapping methods of feature extraction like PCA, Kernel PCA, etc.
    - Focus on data variance
    - Not on distribution difference
- A simple objective function can be used to find mapping for feature extraction when there is little difference in conditional distribution:

$$\min_{\Phi} \left(\mathrm{DIST}(X^S, X^T; \Phi) + \lambda\Omega(\Phi)\right) / \left(\mathrm{VAR}(X^S \cup X^T; \Phi)\right),$$

where Φ is a low-dimensional mapping function, $\mathrm{DIST}(\cdot)$ represents a distribution difference metric, $\Omega(\Phi)$ is a regularizer controlling the complexity of Φ, and $\mathrm{VAR}(\cdot)$ represents the variance of instances. This objective function aims to find a mapping function Φ that minimizes the marginal distribution difference between domains and meanwhile makes the variance of the instances as large as possible.

- Finding explicit formulation of $\Phi(\cdot)$ is non-trivial – we need to minimize the numerator (Distance) while maximizing the denominator (Variance)
- One way to deal with this challenge is to first optimize the objective of the numerator and then realize the objective of the denominator. Three main pathways to deal with problem:
    - Mapping Learning + Feature Extraction
        - Find high-dimensional feature space by solving kernel matrix or a learning transformation matrix, and
        - Then use PCA, etc. to form low-dimensional representation
    - Mapping Construction + Mapping Learning
        - Find high-dimensional feature space by solving kernel matrix learning problem
        - Then learn transformation matrix to form low-dimensional representation
    - Direct Low-dimensional Mapping Learning
        - Usually difficult to directly find low-dimensional mapping. However, it is solvable in certain conditions – e.g. when mapping is restricted to linear.
- Other approach: Matching conditional distribution and preserve the structures of the data. The above objective function needs to be modified - requires additional terms and constraints.

$$\begin{aligned}\min_{\Phi} \mu\mathrm{DIST}(X^S, X^T; \Phi) + \lambda_1\Omega^{\mathrm{GEO}}(\Phi) + \lambda_2\Omega(\Phi) \\ + (1-\mu)\mathrm{DIST}(Y^S|X^S, Y^T|X^T; \Phi), \\ s.t.\ \Phi(X)^{\mathrm{T}} H \Phi(X) = I,\ \text{with } H = I - (\mathbf{1}/n) \in \mathbb{R}^{n\times n},\end{aligned}$$

- More advanced mapping techniques are required in such case.
    - Maximum Mean Discrepancy Embedding (MMDE)
    - Transfer Component Analysis (TCA)
    - Joint Distribution Adaptation (JDA)
    - Balanced Distribution Adaptation (BDA) and Weighted BDA (WBDA)
    - Adaptation Regularization Based Transfer Learning (ARTL)
    - Domain Transfer Multiple Kernel Learning (DTMKL)

Featuring Clustering

- Find more abstract representation of original features – different from mapping-based extraction
- Example: reducing features using co-clustering – simultaneously clusters rows and columns using information theory
- Co-Clustering Based Classification (CoCC) by Dai et al. uses co-clustering for document classification
    - Co-clustering as a technique to transfer knowledge – source and target document-to-word matrix is co-clustered
    - Minimize the joint loss in mutual-information – iterative process with 2-step single iteration
        - Document Clustering – reorder document-to-word matrix for target document
        - Word Clustering – adjust word clusters to minimize joint mutual-information loss

- Self-Taught Cluster (STC) – unsupervised co-clustering proposed by Dai et al.
    - Does not need label information but two domains should share the same feature clusters in their common feature space
    - Each iteration involves 2-steps minimizing mutual-information loss:
        - Instance Clustering: Clustering updated to minimize respective loss
        - Feature Clustering: Feature clusters are updated to minimize the joint loss
- Concept-based Transfer Learning Approaches
    - Latent Semantic Analysis (LSA) – maps document-to-word matrix to latent space using SVD
        - SVD can remove the irrelevant information and the noise
    - Probabilistic LSA (PLSA) – Constructs a Bayesian network & uses EM to estimate parameters
        - Latent variable ***z***, reflects the concept and associates document ***d*** with the word ***w***
    - Dual-PLSA – 2 latent variables z^d & z^w reflecting concepts behind documents and words
    - Topic-Bridged PLSA (TPSLA) – Assumes source and target instances share the same mixing concepts of the words
    - Collaborative Dual-PLSA (CD-PLSA) – For multi-domain text classification
    - Homogeneous-Identical-Distinct-Concept-Model (HIDC) – Extension of Dual-PLSA
        - Composed of three generative models – identical-concept, homogeneous-concept, and distinct-concept models and used EM-algo to estimate parameters

Feature Selection

- Feature reduction method to extract pivot features – features that behave in the same way in different domains
- Stability of pivot features aid in acting as bridge for transfer of knowledge
- Structural Correspondence Learning (SCL) by Blitzer et al.: Consists of the following steps:
    - Feature Selection – to obtain pivot features
    - Mapping Learning – find low dimensional latent representation using Structured Learning
    - Feature Stacking – construct new feature representation by stacking original features with latent ones
- Example: Part of speech tagging problem – final classifier is trained in augmented feature space

Feature Encoding

- Autoencoder are often used for feature encoding to produce more abstract representation of the input
- Stacked Denoising Autoencoder (SDA) by Glorot et al.:
    - Denoising autoencoder which can enhance the robustness
    - Contains randomly corrupting mechanism that adds noise to the input before mapping
    - Mainly composed of the following steps:
        - Autoencoder Training: source and target instances are used to train a stack of denoising autoencoders in a greedy layer by layer way
        - Feature encoding & Stacking: a new representation is constructed by stacking the encoding output of intermediate layers
        - Learner Training: The target classifier is trained on the transformed labeled instances
    - Challenge: High computational and parameter estimation costs
- Marginalized Stacked Linear Denoising Autoencoder (mSLDA) by Chen et al. adopts linear autoencoder and marginalizes the randomly corrupting step in a closed form to shorter the training time
    - Linear autoencoder often sufficient when dealing with high-dimensional data
    - Basic architecture is single layer linear autoencoder
    - Minimizing expected squared reconstruction loss function

$$W = \arg\min_{W} \frac{1}{2n} \sum_{i=1}^{n} \mathbb{E}_{P(\tilde{\mathbf{x}}_i|\mathbf{x})} \left[||\mathbf{x}_i - W\tilde{\mathbf{x}}_i||^2 \right],$$

where $\tilde{\mathbf{x}}_i$ denotes the corrupted version of the input $\mathbf{x}_i$. The solution of W is given by [98], [99]:

$$W = \left(\sum_{i=1}^{n} \mathbf{x}_i \mathbb{E}[\tilde{\mathbf{x}}_i]^{\mathrm{T}} \right) \left(\sum_{i=1}^{n} \mathbb{E}\left[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^{\mathrm{T}} \right] \right)^{-1}.$$

Feature Alignment

- In addition to explicit features, feature alignment also focuses on implicit features such as:
  - Statistic Features
  - Spectral Features
- Example, explicit features can be aligned to generate new representation; implicit features can be aligned to construct a satisfied feature transformation
- Typical alignment process, let's say for subspace features looks like:
  - Subspace Generation: generate respective subspaces from source and target domains
  - Subspace Alignment: mapping to align subspaces orthonormal bases is learned
  - Learner Training: Target learner is trained on transformed instances using projection to aligned bases
- Subspace Alignment (SA) by Fernando et al.:
  - Subspaces generated using PCA and bases obtained by selecting leading eigenvectors
  - Transformation matrix ***W*** is learned to align subspaces

$$W = \arg\min_{W} ||M_S W - M_T||_F^2 = M_S^{\mathrm{T}} M_T,$$

- Subspace Distribution Alignment between Two Subspaces (SDA-TS) by Sun & Saenko:
  - Aligns both subspaces and the distributions
- Geodesic Flow Kernel (GFK) by Gong et al. based on Geodesic Flow Subspaces integrates infinite number of subspaces located on geodesic curve from the source to the target subspace
- Co-Relation Alignment (CORAL) by Sun et al. constructs transformation matrix by aligning the second-order statistic features i.e. covariance matrices

$$W = \arg\min_{W} ||W^{\mathrm{T}} C_S W - C_T||_F^2,$$

where C denotes the covariance matrix. Note that, compared to the above subspace-based approaches, CORAL avoids subspace generation as well as projection and is very easy to implement.

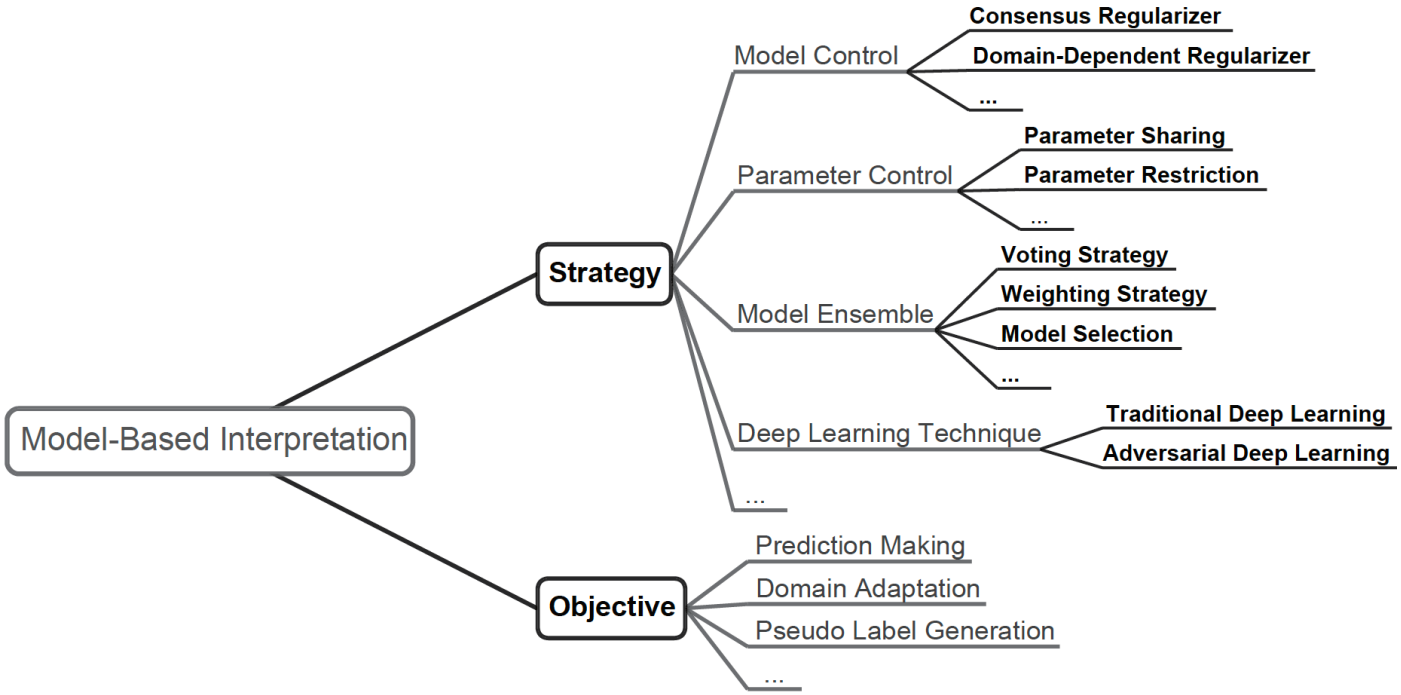
- **Spectral Feature Alignment (SFA) by Pan et al.:**
  - Algorithm for sentiment classification based on spectral clustering and feature alignment
  - Spectral clustering, based on graph theory, uses eigenvalues of similarity matrix to reduce dimension of the features before clustering
  - Identify domain-specific and domain-independent words
  - Aligns domain-specific word features to construct low dimensional representation
  - SFA generally consists of following 5-steps:
    - Feature Selection: Select domain independent/pivot features
      - 3-different strategies: Frequency of words, Mutual information between features and labels, and Mutual information between features and domains
    - Similarity Matrix Construction: Construct a bipartite graph
      - Each edge is assigned a weight that measure the co-occurrence relationship between domain-specific and domain-independent words
      - Based on that construct similarity matrix
    - Spectral Feature Alignment: Adapting spectral clustering algorithm
      - Align domain-specific features and obtain low-dimensional feature space
      - Based on eigenvectors of the graph Laplacian

        - Feature Stacking: Original features and low dimensional features
            - Stacked to produce final representation
        - Learner Training: On final representation using labeled instances
- Cross-Domain Spectral Clustering (CDSC) by Ling et al. is another spectral transfer learning approach which only uses 3-step approach

Model-based Interpretation

Transfer learning approaches can also be interpreted from model perspective. TL model may consist of a few sub-modules such as:

- Classifiers
- Extractors
- Encoders



Strategy

- Model Control
- Parameter Control
- Model Ensemble
- Deep Learning Technique

Objectives

- Prediction Making
- Domain Adaptation
- Pseudo Label Generation

Model Control Strategy

- Add the model-level regularizers to the learner's objective function so the knowledge contained in the pre-obtained source models can be transferred into the target model during the training
- Domain Adaptation Machine (DAM) by Duan et al.:
    - For multi-source transfer learning
    - Goal: to construct a robust classifier for the target domain with the help of some pre-obtained base classifiers that are respectively trained on multiple source domains
    - Objective:

$$\min_{f^T} \mathcal{L}^{T,L}(f^T) + \lambda_1 \Omega^{\mathrm{D}}(f^T) + \lambda_2 \Omega(f^T),$$

        - the first term = loss function used to minimize the classification error of the labeled target-domain instances
        - the second term = regularizers, and
        - the third term is used to control the complexity of the final decision function f^T

- Special Case of DAM:
    - Consensus Regularization Framework (CRF) by Luo et al.
        - No labeled target instances
        - m-Classifiers are required to reach consensus
    - Fast-DAM (Domain-dependent Regularizer)
        - Transfers the knowledge motivated by domain dependence
        - High Computational Effificeny
    - Univer-DAM (Domain-dependent Regularizer + Universum Regularizer)
        - Extension of Fast DAM
        - Contains additional regularizer – Universum Regularizer utilizes additional dataset which instances do not belong to either class.
        - Treat source instances as Universum for target domain

Parameter Control Strategy

- Focuses on the parameters of models.
- For example, in the application of object categorization
    - Knowledge from source categories can be transferred into target categories via object attributes such as shape and color
    - Attribute priors i.e. distribution parameters can be learned from the source domain

Parameter Sharing

- Directly share the parameters of the source learner to the target learner
- Widely employed in Network based approaches
    - In a deep neural network for the source task, we can freeze most of its layers and only finetune the last few layers to produce a target network
- Matrix factorization-based parameter sharing:
    - Matrix Tri-Factorization Based Classification Framework (MTrick) by Zhuang et al.
        - Words expressing similar connotative meaning in different domains
        - Find connections between the document classes and concepts conveyed by the word clusters – stable knowledge to be transferred
        - Decompose document to word matrix into three matrices:
            - Document to cluster matrix
            - Connection matrix
            - Cluster to word matrix
    - Triplex Transfer Learning (TriTL) by Zhuang et al. (an extension of MTrick)
        - Assumes domain concepts can be divided into three types (similar to HIDC):
            - Domain-independent
            - Transferable domain-specific
            - Non-transferable domain-specific
    - Wang et al. proposed TL framework for image classification that integrates two matrix tri-factorization into a joint framework
    - Do et. Al. utilizes matrix tri-factorization for discovering implicit and explicit similarities for cross-domain recommendation

Parameter Restriction

- Rather that sharing of parameters, Parameter Restriction only requires parameters to be similar
- Proposed Approaches:
    - Single-Model Knowledge Transfer (SMKL) by Tommasi et al. (based on Least Square SVM)
    - Multi-Model Knowledge Transfer (MMKL) by Tommasi et al. (extension of SMKL)

Model Ensemble Strategy

- This strategy aims to combine several weak classifiers to make the final predictions.
- For Example, TrAdaBoost and MsTrAdaBoost ensemble the weak classifiers via voting and weighting, respectively.
- TaskTrAdaBoost (extension of TrAdaBoost for multi-source scenarios) involves the following 2 stages:
    - Candidate Classifier Construction
        - A group of candidate classifiers are constructed by performing AdaBoost on each source domain
    - Classifier Selection and Ensemble
        - A revised version of AdaBoost is performed on the target domain instances to construct the final classifier.
        - In each iteration, an optimal candidate classifier which has the lowest classification error on the labeled target-domain instances is picked out and assigned with a weight based on the classification error.
        - Then, the weight of each target-domain instance is updated based on the performance of the selected classifier on the target domain.
        - After the iteration process, the selected classifiers are ensembled to produce the final prediction
- Locally Weighted Ensemble (LWE) by Gao et al.
    - Focuses on the ensemble process of various learners
        - These learners could be constructed on different source domains, or
        - Be built by performing different learning algorithms on a single source domain
    - The learner is usually assigned with different weights when classifying different target-domain instances
    - Graph-based approach to estimate the weights
- Ensemble Framework of Anchor Adapters (ENCHOR) by Zhuang et al.
    - Weighting ensemble framework which adjusts weights of instances to train a new learner iteratively
    - Constructs a group of weak learners from different representations of the instances produced by anchors
        - Anchor Selection
        - Anchor-based Representation Generation
        - Learner Training and Ensemble

Deep Learning Technique

- SDA and mSLDA approaches utilize deep learning techniques
- Divided into two types
    - Traditional/Non-Adversarial
    - Adversarial

Traditional Deep Learning

- In additional to SDA and mSLDA, there are other reconstruction-based TL approaches
- Transfer Learning with Deep Autoencoders (TLDA) by Zhuang el al.
    - Adopts 2-autoencoders for the source and target domain
    - 2-autoencoders share the parameters and have 2-layers each for encoder and decoder
    - Several objectives of TLDA
        - Reconstruction error minimization: for decoder output (X)
        - Distribution Adaptation: for intermediate layer output (Q)
        - Regression error minimization: for encoder output with labels (R)

$$\min_{\Theta} \mathcal{L}_{\text{REC}}(X, \tilde{X}) + \lambda_1 \text{KL}(Q^S || Q^T) + \lambda_2 \Omega(W, b, \hat{W}, \hat{b}) + \lambda_3 \mathcal{L}_{\text{REG}}(R^S, Y^S),$$

- Deep Adaptation Network (DAN) by Long et al.
    - Multi-layer adaptation utilizing multi-kernel technique
    - Train of thought is similar to Generative Adversarial Networks (GANs)
- Deep CORAL (DCORAL) by Sun and Saenko
    - Extends CORAL for deep domain adaptation
    - CORAL loss is added to minimize the feature covariance
- Contrastive Adaptation Network (CAN) by Kang et al.
    - Based on discrepancy metric termed contrastive domain discrepancy
- Multi-Representation Adaptation Network (MRAN) by Zhu et al.
    - Adapts the extracted multiple feature representation
- Multiple Feature Spaces Adaptation Network (MSFAN) by Zhu et al.
    - Deep learning technique for multi-source transfer learning
    - Architecture consists of:
        - Common feature extractor
        - Domain specific feature extractor
        - Domain specific classifier
    - MSFAN has following step in each iteration:
        - Common Feature Extraction
        - Specific Feature Extraction
        - Data Classification
        - Parameter Updating
    - Three main objectives:
        - Classification error minimization
        - Distribution Adaptation
        - Consensus Regularization

Adversarial Deep Learning

- Many TL approaches, motivated by GAN, are based on assumption that a good feature representation contains almost no discriminative information about the instance's original domain
- Deep Adversarial Neural Network (DANN) by Ganin et al.
    - Assumes no labeled target instances
    - Architecture consists of:
        - feature extractor: acts like generator
        - label predictor: label prediction of the instances
        - domain classifier: acts like discriminator

$$\begin{array}{l} \xrightarrow[\text{Predictor}]{\text{Label}} \hat{Y}^{S,L} \\ \qquad\qquad\quad \hat{Y}^{T,U} \end{array}$$

$$\begin{array}{l} X^{S,L} \\ X^{T,U} \end{array} \xrightarrow[\text{Extractor}]{\text{Feature}} \left.\begin{array}{l} Q^{S,L} \\ Q^{T,U} \end{array}\right\} \xrightarrow[\text{Classifier}]{\text{Domain}} \begin{array}{l} \hat{S} \\ \hat{T} \end{array} \text{(Domain Label)}$$

- Conditional Domain Adversarial Network (CDAN) by Long et al. utilizes conditional domain discriminator to assist adversarial adaptation
- Importance Weighted Adversarial Nets-Based Domain Adaptation (IWANDA) by Zhang et al.
    - 2-domain-specific feature extractors for source and target domains
    - 2-domain classifiers but only 1 label predictor

$$\begin{array}{l} \xrightarrow[\text{Predictor}]{\text{Label}} \hat{Y}^{S,L} \\ \qquad\qquad\quad \hat{Y}^{T,U} \end{array}$$

$$\begin{array}{l} X^{S,L} \xrightarrow[\text{Extractor}]{\text{Source Feature}} Q^{S,L} \\ X^{T,U} \xrightarrow[\text{Extractor}]{\text{Target Feature}} Q^{T,U} \end{array} \left.\begin{array}{l} \\ \\ \end{array}\right\} \xrightarrow[+\hat{Y}^{T,U}]{+\boldsymbol{\beta}^{S}} \xrightarrow[\text{Classifier}]{\text{2nd Domain}} \begin{array}{l} \hat{S}_2 \\ \hat{T}_2 \end{array}$$

$$\xrightarrow[\text{Classifier}]{\text{1st Domain}} \begin{array}{l} \hat{S}_1 \\ \hat{T}_1 \end{array} \xrightarrow[\text{Function}]{\text{Weight}} \boldsymbol{\beta}^{S}$$

Application

Transfer Learning for text analysis and image analysis is playing an important role in the following areas:

Medical Application

- Medical Imaging: ML and computer aided diagnosis
    - Labeling often relies on experienced doctors
    - Makes it harder to collect sufficient training data
- Maqsood et al. finetunes AlexNet for the detection of Alzheimer
- Shin et al. finetuned pretrained deep neural network for solving computer aided detection
- Byra et al. utilized transfer learning to help assess knee osteoarthritis
- Tang et al. combines the active learning and the domain adaptation technologies for the classification of various medical data
- Zeng et al. utilizes transfer learning for automatically encoding ICD-9 codes that are used to describe patient diagnosis

Bioinformatics Application

- Biological Sequence Analysis: Understanding of some organisms can be transferred to other organisms
    - Transfer Learning can be applied to facilitate this task
    - Function of some biological substances may remain unchanged but with the composition changed between two organisms, may result in the marginal distribution difference
- Schweikart et al. uses mRNA splice site prediction problem to analyze the effectiveness of transfer learning with promising results.
- Gene Expression Analysis: Predicting association between genes and phenotypes
    - Suffers from data sparsity – Transfer Learning can come to rescue
- Petegrosso et al. proposed Label Propagation Algorithm (LPA) based approach to anylyze and predict the gene-phenotype
- Xue et al. proposes collective matrix factorization technique to transfer the linkage knowledge from the source Protein-Protein Interaction (PPI) network to the target network

Transportation Application

- Understanding Traffic Image Scenes: Suffer from variations due to weather and light conditions
- Di et al. proposes and approach to transfer the information of the images of the same location taken in different conditions
- Driver Behavior Modeling: Sufficient personalized data of each driver is usually unavailable
- Lu et al. proposes an approach to driver model adaptation in lane changing scenarios
- Liu et al. applied transfer learning to driver poses recognition
- Wang et al. adopted the regularization technique for vehicle type recognition
- TL can also be used for anomalous activity detection and traffic sign recognition

Recommender-System Application

- Effectively recommending personalized content is an important issue: Traditional method rely on user-item interaction matrix
    - Requires large amount of data; data is however sparse and doesn't even exist for new users
    - Instance and feature based approaches can come for rescue
- Pan et al. leverage uncertain ratings from source domain as constraints to complete rating matrix factorization for target domain
- Hu et al. uses transfer meeting hybrid to extract knowledge from unstructured text by using attentive memory network and selectively transferring useful information
- Pan et al. uses Coordinate System Transfer (CST) to leverage user side and item side latent features
- He et al. proposes TL framework based on Bayesian neural network

Communication Application

- WiFi localization task and wireless-network applications
- Energy saving scheme for cellular radio access networks
- Topology management for reducing energy consumption

Urban-Computing Application

- Urban computing is a promising research area focusing on traffic monitoring, health care, social security, etc.: TL can help deal with data scarcity
- Guo et al. proposes a chain store site recommendation leveraging knowledge from semantically relevant domains (i.e. other cities with same store, other stores in the same city)
- Wei et al. proposes a multi-modal TL to transfer knowledge from city with sufficient data and labels to city with sparse data
- Hand Gesture Recognition, Face Recognition, Activity Recognition, and Speech Emotion Recognition
- Sentiment Analysis, Fraud Detection, Social Network Analysis, and Hyperspectral Image Analysis

Experiment

Experiments are conducted to evaluate some Representative Transfer Learning models across two mainstream research areas:

- Object Recognition
- Text Classification

Dataset and Preprocessing

Three different datasets have been studied in these experiments.

Statistical information of the preprocessed datasets.

Area	Dataset	Domain	Attribute	Total Instances	Tasks
Sentiment Classification	Amazon Reviews	4	5000	27677	12
Text Classification	Reuters-21578	3	4772	6570	3
Object Recognition	Office-31	3	800	4110	6

Amazon Reviews

- Multi-domain sentiment dataset from Amazon.com
- Four domains: Books, Kitchen, Electronics, and DVD's
- Reviews with rating from 0 to 5 with 0 to 2 defined as negative and 3 to 5 defined as positive
- 5K words with highest frequency are selected as attributes of each review
- 1K positive, 1K negative and 5K unlabeled instances in each domain
- Every 2 of the 4 domains are selected to generate 12 tasks

Reuters-21578

- Text categorization dataset with hierarchical structure
- 5 top categories – Exchanges, Orgs, People, Places, Topics
- 3 categories used, Orgs, People, Places – 2 categories are selected at a time across 6 tasks
- Each domain has 1K instances and 4.5K features
- All instances have label so for unlabeled data labels were ignored

Office-31

- Object recognition dataset with 31 categories and 3 domains (Amazon, Webcam, DSLR)
- 2817, 498 and 795 instances across 3 domains respectively
- 2 domains are selected at a time resulting in 6 tasks

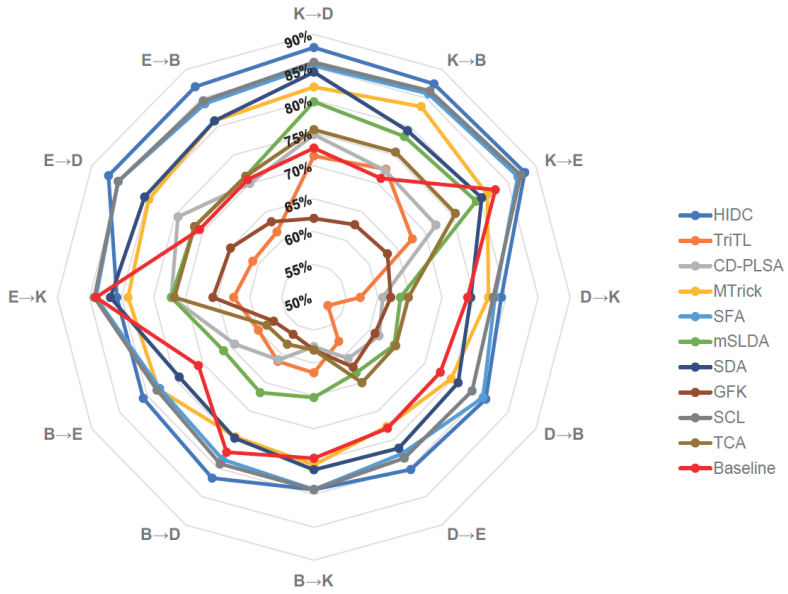
Experiment Setting

- Experiments to compare representative learning tasks
  - 8 experiments on Office-31 dataset
  - 14 experiments on Reuters-21578 dataset
  - 11 experiments on Amazon Reviews
- The classification results are evaluated by accuracy on test set
- For all algorithms (except TrAdaBoost) unlabeled target domain instances are used
- Each algorithm was executed three times and average results are adopted
- Models evaluated include:
  - HIDC
  - TriTL
  - CD-PLSA
  - MTrick
  - SFA
  - mSLDA
  - JDA
  - TrAdaBoost
  - DAN
  - DCORAL
  - CoCC
  - MRAN
  - CDAN
  - DANN
  - JAN
  - CAN
  - TCA
  - SDA
  - SCL
  - GFK

Experiment Result

- Compared over 20 algorithms across 3 datasets
- Parameters of all algorithms were set to recommended or default values mentioned in the original papers

Amazon Reviews



Accuracy performance on the Amazon Reviews of four domains: Kitchen (K), Electronics (E), DVDs (D) and Books (B).

Model	K→D	K→B	K→E	D→K	D→B	D→E	B→K	B→D	B→E	E→K	E→D	E→B	Average
HIDC	0.8800	0.8750	0.8800	0.7925	0.8100	0.8025	0.7925	0.8175	0.8075	0.8075	0.8700	0.8700	0.8338
TriTL	0.7150	0.7250	0.6775	0.5725	0.5250	0.5775	0.6150	0.6125	0.6000	0.6250	0.6100	0.6150	0.6225
CD-PLSA	0.7475	0.7225	0.7200	0.6075	0.6175	0.6075	0.5750	0.6100	0.6425	0.7225	0.7450	0.7000	0.6681
MTrick	0.8200	0.8350	0.8125	0.7725	0.7475	0.7275	0.7550	0.7450	0.7800	0.7900	0.7975	0.8100	0.7827
SFA	0.8525	0.8575	0.8675	0.7825	0.8050	0.7750	0.7925	0.7850	0.7775	0.8400	0.8525	0.8400	0.8190
mSLDA	0.7975	0.7825	0.7925	0.6350	0.6450	0.6325	0.6525	0.6675	0.6625	0.7225	0.7150	0.7125	0.7015
SDA	0.8425	0.7925	0.8025	0.7450	0.7600	0.7650	0.7625	0.7475	0.7425	0.8175	0.8050	0.8100	0.7827
GFK	0.6200	0.6275	0.6325	0.6200	0.6100	0.6225	0.5800	0.5650	0.5725	0.6575	0.6500	0.6325	0.6158
SCL	0.8575	0.8625	0.8725	0.7800	0.7850	0.7825	0.7925	0.7925	0.7825	0.8425	0.8525	0.8450	0.8206
TCA	0.7550	0.7550	0.7550	0.6475	0.6475	0.6500	0.5800	0.5825	0.5850	0.7175	0.7150	0.7125	0.6752
Baseline	0.7270	0.7090	0.8270	0.7400	0.7280	0.7300	0.7450	0.7720	0.7080	0.8400	0.7060	0.7070	0.7449

- Most algorithms performed relatively well when the source domain was electronics or kitchen
    - These domains may contain more transferable information
- Performed relatively well in all 12 tasks
    - HIDC (feature reduction – clustering)
    - SCL (feature selection)
    - SFA (feature alignment)
    - MTrick (feature reduction – clustering)
    - SDA (feature encoding)

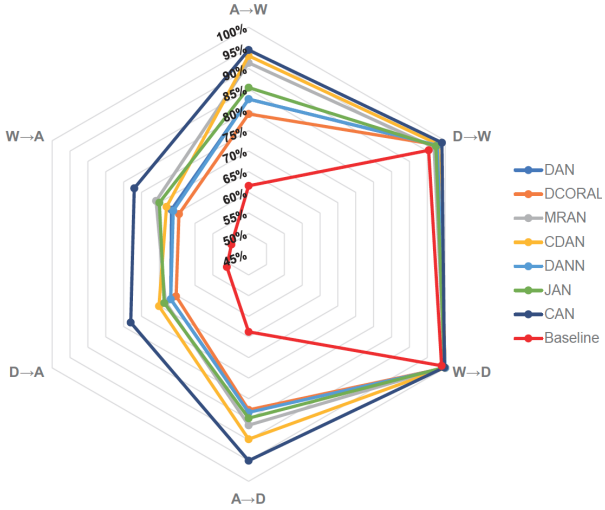
Reuters-21578

Accuracy performance on the Reuters-21578 of three domains: Orgs, People, and Places.

Model	Orgs vs Places	People vs Places	Orgs vs People	Average
HIDC	0.7698	0.6945	0.8375	0.7673
TriTL	0.7338	0.5517	0.7505	0.6787
CD-PLSA	0.5624	0.5749	0.7826	0.6400
MTrick	0.7494	0.6457	0.7930	0.7294
CoCC	0.6704	0.8264	0.7644	0.7537
SFA	0.7468	0.6768	0.7906	0.7381
mSLDA	0.5645	0.6064	0.5289	0.5666
SDA	0.6603	0.5556	0.5992	0.6050
GFK	0.6220	0.5417	0.6446	0.6028
SCL	0.6794	0.5046	0.6694	0.6178
TCA	0.7368	0.6065	0.7562	0.6998
JDA	0.5694	0.6296	0.7424	0.6471
TrAdaBoost	0.7336	0.7052	0.7879	0.7422
Baseline	0.6683	0.5198	0.6696	0.6192

- Most algorithms performed relatively well for Orgs vs People and Orgs vs Places but poor for People vs Places
    - Difference between People and Places may be relatively large
- Consistent Performance across 3 tasks
    - HIDC
    - SFA
    - MTrick
- Top Performers for People vs Places
    - TrAdaBoost
    - CoCC

Office-31



Accuracy performance on Office-31 of three domains: Amazon (A), Webcam (W), and DSLR (D).

Model	A → W	D→W	W→D	A→D	D → A	W→A	Average
DAN	0.826	0.977	1.00	0.831	0.668	0.666	0.828
DCORAL	0.790	0.980	1.00	0.827	0.653	0.645	0.816
MRAN	0.914	0.969	0.998	0.864	0.683	0.709	0.856
CDAN	0.931	0.982	1.00	0.898	0.701	0.680	0.865
DANN	0.826	0.978	1.00	0.833	0.668	0.661	0.828
JAN	0.854	0.974	0.998	0.847	0.686	0.700	0.843
CAN	0.945	0.991	0.998	0.950	0.780	0.770	0.906
Baseline	0.616	0.954	0.990	0.638	0.511	0.498	0.701

- All 7 algorithms have excellent performance on tasks D→W and W→D
    - Consistent with the fact that WebCam and DSLR are more similar to each other than Amazon
- CAN outperforms other 6 algorithms and DAN and DANN do a good job as well
    - Adversarial learning is effective

Conclusion

- Performance of some algorithms is less than ideal
- Parameters selected may not be suitable for the datasets
- Suitability of algorithms is likely domain dependent
- Selection of Transfer Learning model is a complex issue and an important research topic

Future Areas of Exploration

- Application of transfer learning to wider range of applications
- User privacy protection
- How to avoid negative transfers?
- Measuring transferability across domains
- Interpretability of transfer learning
- Theoretical support for transfer learning
- Heterogenous transfer learning
- Reinforcement transfer learning

Cross-Domain Sentiment Classification via Spectral Feature Alignment

Authors: Sinno Jialin Pany, Xiaochuan Niz, Jian-Tao Sunz, Qiang Yangy and Zheng Chen

Harvard AC295/CS115

SUMMARY BY: EDUARDO PEYNETTI, JESSICA WIJAYA, ROHIT BERI, STUART NEILSON

Abstract

Spectral Feature Alignment (SFA) algorithm to align domain-specific words from different domains into unified clusters with the help of domain-independent words as a bridge by simultaneously co-clustering them in a common latent space.

Introduction

Due to mismatch between domain-specific words, a sentiment classifier trained on one domain may not work well when directly applied to other domains. Cross domain sentiment classification algorithms can, however, reduce domain dependency and manual labeling cost.

Spectral Feature Alignment (SFA) algorithm helps find new representation for cross-domain sentiment data to reduce the gap between the domains. Spectral clustering algorithm, based on graph spectral theory, is adapted on the bipartite graph to co-align domain-specific and domain-independent words into a set of feature-clusters. SFA seems to be performing better than the state-of-the-art Structural Correspondence Learning (SCL) algorithm in experiments discussed ahead.

Problem Setting

Given two specific domains Source-domian ($\boldsymbol{D_s}$) and Target-domain ($\boldsymbol{D_t}$), and labeled-data $\boldsymbol{\{X_s, Y_s\}}$ for $\boldsymbol{D_s}$ and unlabeled-data for $\boldsymbol{D_t}$, the task is to learn an accurate classifier to predict the sentiment of the unseen data in $\boldsymbol{D_t}$.

The proposed framework targets to achieve the following sub-tasks:

- Identify domain-independent features
    - Occur frequently and act similarly across domains
    - Used as a bridge to make knowledge transfer possible across domains
- Align domain-specific features
    - Domain-specific features act as complement to independent features
    - Align specific features from both domains into "***k***" predefined feature clusters
    - Reduces the difference between domain-specific features in the new representation

A Motivating Example

Let's consider an example of sentiment classification on the below data. Assume the classifier is a linear function that can be written as:

$$y^* = f(x) = \mathbf{sgn}(x \cdot w^T)$$

Table 1: Cross-domain sentiment classification examples: reviews of *electronics* and *video games* products. Boldfaces are domain-specific words, which are much more frequent in one domain than in the other one. Italic words are some domain-independent words, which occur frequently in both domains. "+" denotes positive sentiment, and "-" denotes negative sentiment.

	electronics	*video games*
+	**Compact**; easy to operate; very *good* picture quality; looks **sharp**!	A very *good* game! It is action packed and full of *excitement*. I am very much **hooked** on this game.
+	I purchased this unit from Circuit City and I was very *excited* about the quality of the picture. It is really *nice* and **sharp**.	Very **realistic** shooting action and *good* plots. We played this and were **hooked**.
-	It is also quite **blurry** in very dark settings. I will *never buy* HP again.	The game is so **boring**. I am extremely unhappy and will probably *never buy* UbiSoft again.

Table 2: Bag-of-words representations of *electronics* (E) and *video games* (V) reviews. Only domain-specific features are considered. "..." denotes all other words.

		...	compact	sharp	blurry	hooked	realistic	boring
E	+	...	1	1	0	0	0	0
	+	...	0	1	0	0	0	0
	-	...	0	0	1	0	0	0
V	+	...	0	0	0	1	0	0
	+	...	0	0	0	1	1	0
	-	...	0	0	0	0	0	1

Table 3: Ideal representations of domain-specific words.

		...	sharp_hooked	compact_realistic	blurry_boring
E	+	...	1	1	0
	+	...	1	0	0
	-	...	0	0	1
V	+	...	1	0	0
	+	...	1	1	0
	-	...	0	0	1

Table 4: A co-occurrence matrix of domain-specific and domain-independent words.

	compact	realistic	sharp	hooked	blurry	boring
good	1	1	1	1	0	0
exciting	0	0	1	1	0	0
never_buy	0	0	0	0	1	1

Spectral Domain-Specific Feature Alignment

Algorithm to adapt spectral clustering techniques to align domain-specific features.

Domain-Independent Feature Selection

Three strategies to identify domain-independent features:

- Based on their frequency in both domains
  - Given the number ***"L"*** of domain-independent features to be selected, we choose features that occur more than ***"K"*** times in both source and target domains
  - ***"K"*** is set be the largest number such that we can get at-least ***"L"*** such features
- Based on the mutual dependence between features and labels on the source domain data
  - Mutual information is applied on source domain labeled data to select features as pivots
  - Pivots are recognized as domain-independent features
  - There is no guarantee that such selected features act similarly in both domains
- Modified mutual-information criterion
  - Motivated by supervised feature selection criteria
  - Features with high mutual information are domain-specific
  - Requirement of independent features to occur frequently
  - Modified Mutual Information:

$$I(X^i; D) = \sum_{d \in D} \sum_{x \in X^i, x \neq 0} p(x, d) log_2 \left(\frac{p(x, d)}{p(x)p(d)} \right), \quad (1)$$

where D is a domain variable and we only sum over non-zero values of a specific feature X^i. The smaller $I(X^i; D)$ is, the more likely that X^i can be treated as a domain-independent feature.

Bipartite Feature Graph Construction

Give the domain-independent features (***V_{DI}***) and domain-specific features (***V_{DS}***), construct a bipartite graph.

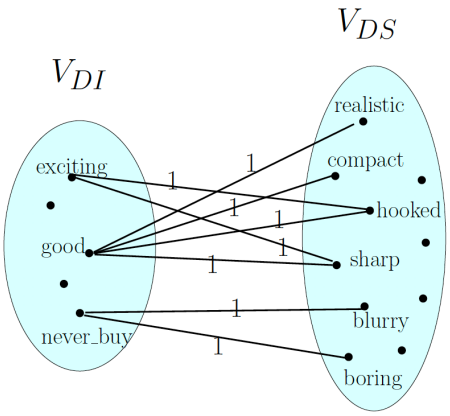


Figure 1: A bipartite graph example of domain-specific and domain-independent features based on Table 4.

Spectral Feature Clustering

Assumptions underlying the SFC:

- Two domain-specific features are connected to many common domain-independent features
  - Then they tend to be very related
  - Will be aligned to same cluster with high probability
- Two domain-independent features are connected to many common domain-specific features
  - Then they tend to be very related
  - Will be aligned to same cluster with high probability
- We can find more compact and meaningful representation of domain-specific features which can reduce the gap between domains

Feature Augmentation

Selected domain-independent and aligned-domain-specific features can be combined as feature augmentation. A tradeoff parameter $\boldsymbol{\gamma}$ is used to balance the effect of original and new features.

$$\widetilde{x}_i = [x_i,\ \gamma\varphi(\phi_{DS}(x_i))]$$

SAF Algorithm

Algorithm 1 Spectral Domain-Specific Feature Alignment for Cross-Domain Sentiment Classification

Input: labeled source domain data $\mathcal{D}_{src} = \{(x_{src_i}, y_{src_i})\}_{i=1}^{n_{src}}$, unlabeled target domain data $\mathcal{D}_{tar} = \{x_{tar_j}\}_{j=1}^{n_{tar}}$, the number of clusters K and the number of domain-independent features m.

Output: adaptive classifier $f : X \to Y$.

1: Apply the criteria mentioned in Section 4.1 on $\mathcal{D}_{src}$ and $\mathcal{D}_{tar}$ to select l domain-independent features. The remaining $m - l$ features are treated as domain-specific features.
$\Phi_{DI} = \begin{bmatrix} \phi_{DI}(x_{src}) \\ \phi_{DI}(x_{tar}) \end{bmatrix}$ and $\Phi_{DS} = \begin{bmatrix} \phi_{DS}(x_{src}) \\ \phi_{DS}(x_{tar}) \end{bmatrix}$.
2: By using Φ_{DI} and Φ_{DS}, calculate (*DI-word*)-(*DS-word*) co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{(m-l)\times l}$.
3: Construct matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$,
where $\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{0} \end{bmatrix}$.
4: Find the K largest eigenvectors of $\mathbf{L}$, $u_1, u_2, ..., u_K$, and form the matrix $\mathbf{U} = [u_1 u_2 ... u_K] \in \mathbb{R}^{m\times K}$.
Let mapping $\varphi(x_i) = x_i\mathbf{U}_{[1:m-l,:]}$, where $x_i \in \mathbb{R}^{m-l}$
5: Return a classifier f, trained on
$\{([x_{src_i}\ \gamma\varphi(\phi_{DS}(x_{src_i}))], y_{src_i})\}_{i=1}^{n_{src}}$

Experiments

Experiments are conduction on two real world datasets to determine the effectiveness of the SFA algorithm for cross-domain sentiment classification

Datasets

- Product Reviews from Amazon across 4 domains – books, dvd's, electronics, & kitchen appliances
- Web-scraping of reviews from Amazon (video games, electronics, software) , Yelp (hotel), and Citysearch (hotel)

Table 5: Summary of Datasets Used for Evaluation.

Dataset	Domain	# Reviews	# Pos	# Neg	# Features
RevDat	dvds	2,000	1,000	1,000	473,856
	kitchen	2,000	1,000	1,000	
	electronics	2,000	1,000	1,000	
	books	2,000	1,000	1,000	
SentDat	video game	3,000	1,500	1,500	287,504
	hotel	3,000	1,500	1,500	
	software	3,000	1,500	1,500	
	electronics	3,000	1,500	1,500	

Baselines

Following baseline models were used for comparison:

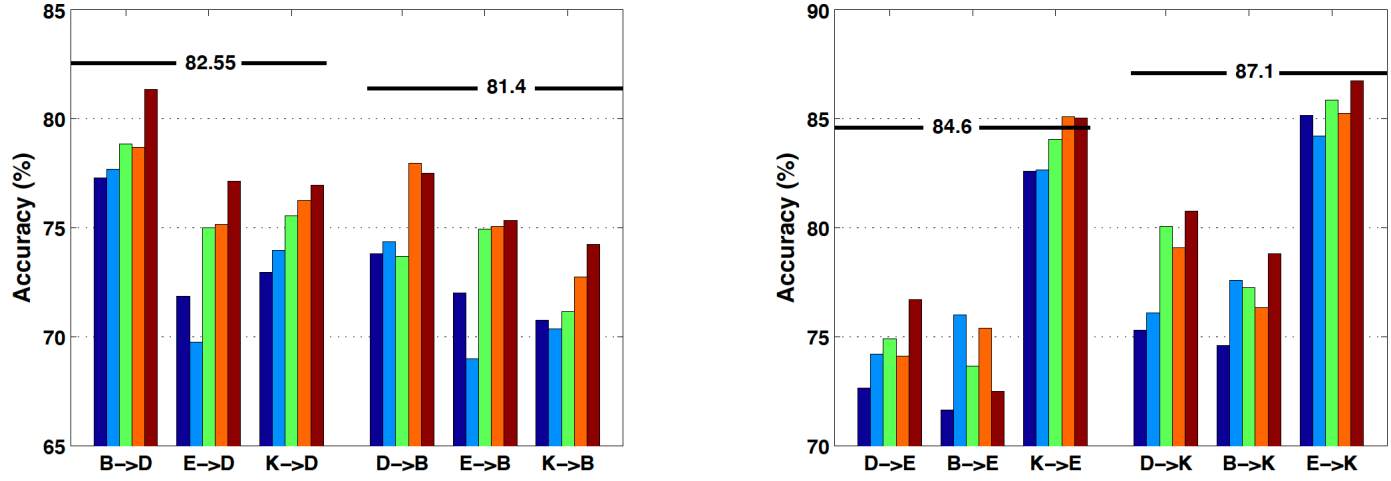
- NoTransf – Classifier trained on source domain
- upperBound – Classifier trained on labeled data from target domain
- FALSA – Classifier trained on augmented representation generated using LSA
- LSA – Classifier trained on latent representation from Latent Semantic Analysis
- SCL – Structured Correspondence Learning

Parameter Settings & Evaluation Criteria

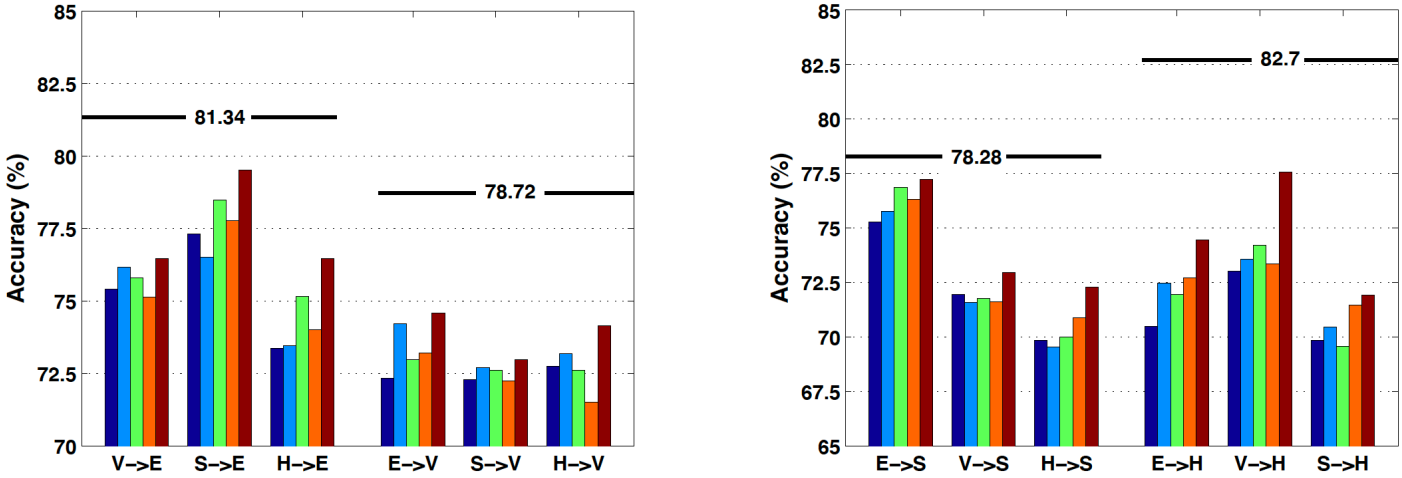
- Logistic regression as base sentiment classifier for all the models including SFA
- $\boldsymbol{\lambda}$ is set to 0.0001 i.e. $\boldsymbol{C}$ = 10,000
- Accuracy is defined as below:

$$Accuracy = \frac{|\{x|x \in \mathcal{D}_{tst} \cap f(x) = y\}|}{|\{x|x \in \mathcal{D}_{tst}\}|}$$

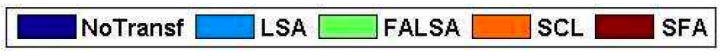
Overall Comparison Result



(a) Comparison Results on $RevDat$.



(b) Comparison Results on $SentDat$.

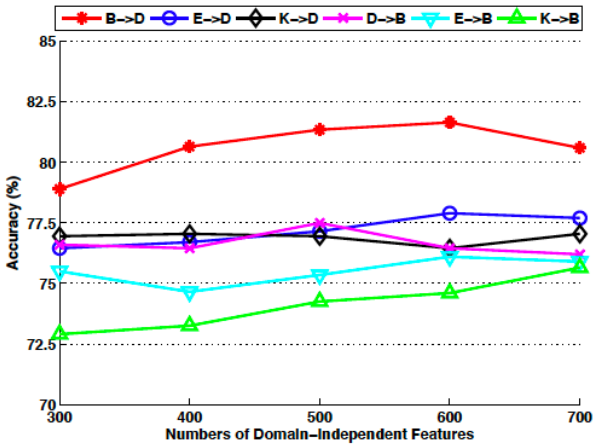


Effect of Domain Independent Features

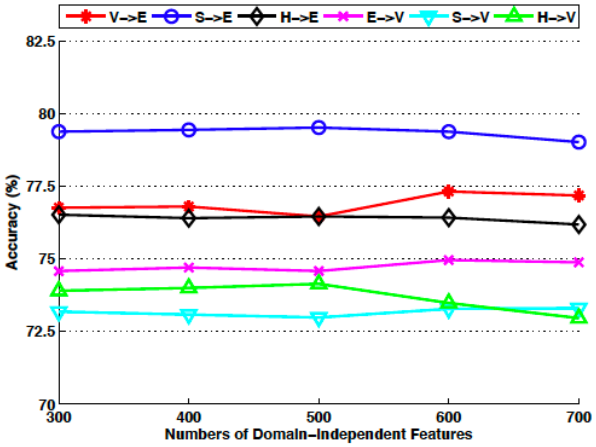
Table 6: Experiments with Different Domain-Independent Feature Selection Methods. Numbers in the table are accuracies in percentage.

RevDat												
	B→D	**E→D**	**K→D**	**D→B**	**E→B**	**K→B**	**D→E**	**B→E**	**K→E**	**D→K**	**B→K**	**E→K**
SFA_{DI}	81.35	77.15	76.95	77.5	75.65	74.8	76.7	72.5	85.05	80.75	78.8	86.75
SFA_{FQ}	81.25	77	76.6	78.25	75.35	74.25	76.05	73.45	84.9	80.6	79.05	85.8
SFA_{MI}	80.1	70.4	78.45	79.8	78.25	75.15	70.85	73	82.05	78.9	78.8	86.75
SentDat												
	V→E	**S→E**	**H→E**	**E→V**	**S→V**	**H→V**	**E→S**	**V→S**	**H→S**	**E→H**	**V→H**	**S→H**
SFA_{DI}	76.64	79.52	76.46	74.58	72.98	74.14	77.22	72.94	72.3	74.44	77.58	71.92
SFA_{FQ}	76.62	79.5	76.64	74.38	73.16	74.54	77.5	72.96	72.22	75	77.46	71.62
SFA_{MI}	76.96	79.08	76.46	75.06	73.86	74.88	77.48	73.22	72.38	75.98	77.08	72.46

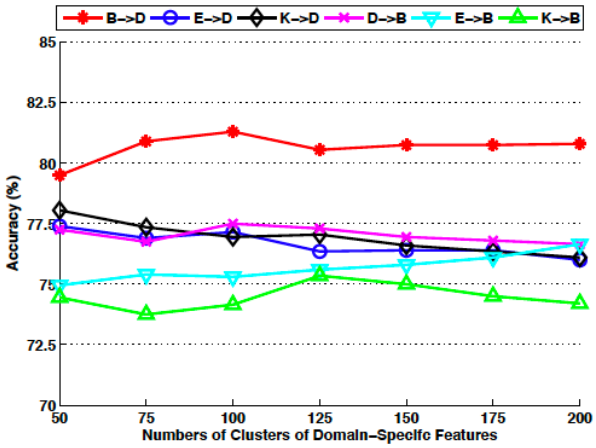
Parameter Sensitivity



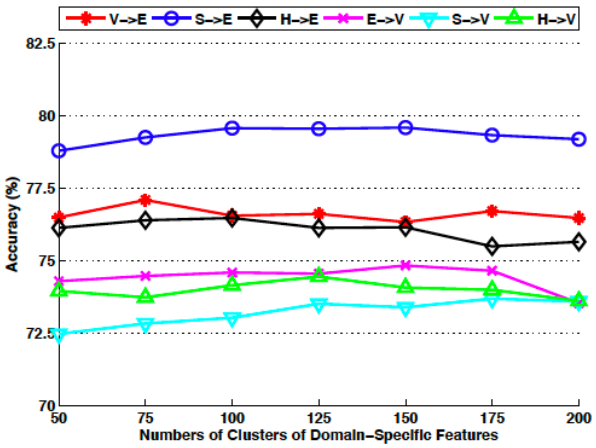
(a) Results on $RevDat$ under Varying Numbers of Domain-Independent Features.



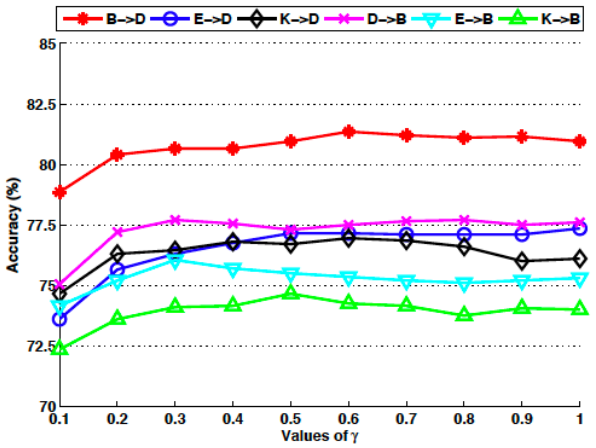
(b) Results on $SentDat$ under Varying Numbers of Domain-Independent Features.



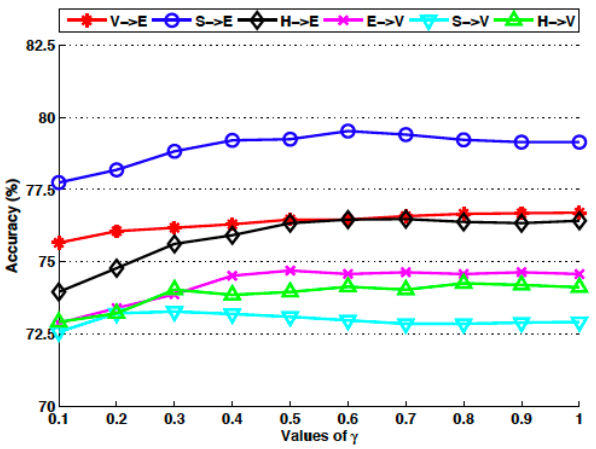
(c) Results on $RevDat$ under Varying Numbers of Feature-Clusters.



(d) Results on $SentDat$ under Varying Numbers of Feature-Clusters.



(e) Results on $RevDat$ under Varying Values of γ.



(f) Results on $SentDat$ under Varying Values of γ.

Related Work

Structured Correspondence Learning (SCL) by Blitzer et al. exploits domain adaptation techniques for sentiment classification:

- Motivated by multi-task learning algorithm, Alternating Structural Optimization by Ando and Zhang
- Tries to construct set of related tasks to model relationship between pivot-features and non-pivot features

- Non-pivot features with similar weights among tasks tend to be close with each other in low-dimensional latent space

Conclusion

Experimental results on both document-level and sentence-level sentiment classification tasks demonstrate the effectiveness of SFA