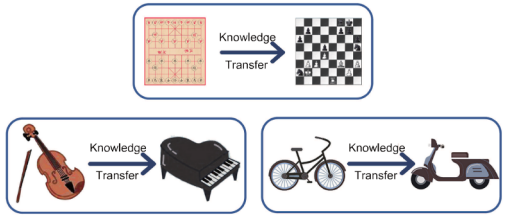
Comprehensive Survey on Transfer Learning

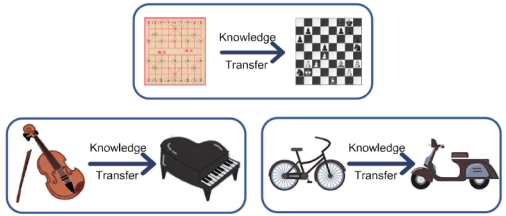
Abnormal Distribution
Eduardo Peynetti, Jessica Wijaya, Rohit Beri, Stuart Neilson

Introduction



- Learning to transfer is the result of the **generalization of experience**
  - There needs to be a **connection** between two learning activities
- **Negative Transfer**:
  - no positive impact on new task
  - no relevance between source vs. target domains and the learner's capacity to find transferable knowledge across domains
  - target learner is negatively affected by the transferred knowledge
  - Example: learning to bike vs play piano, leaning Spanish and French

Introduction



- **Homogenous Transfer**:
  - Domains are in the same feature space
  - Difference is only in marginal distributions
    - only need domain adaptation e.g. sample selection bias or covariate shift
- **Heterogenous Transfer**:
  - Domains have different feature space
    - require feature space adaptation in addition to distribution adaptation

Related Areas

Semi-Supervised Learning (SSL)

- Combines abundant unlabeled instances 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 TL
- 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

Related Areas

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 heterogeneous sensor platform

Related Areas

Multi-Task Learning (MTL)

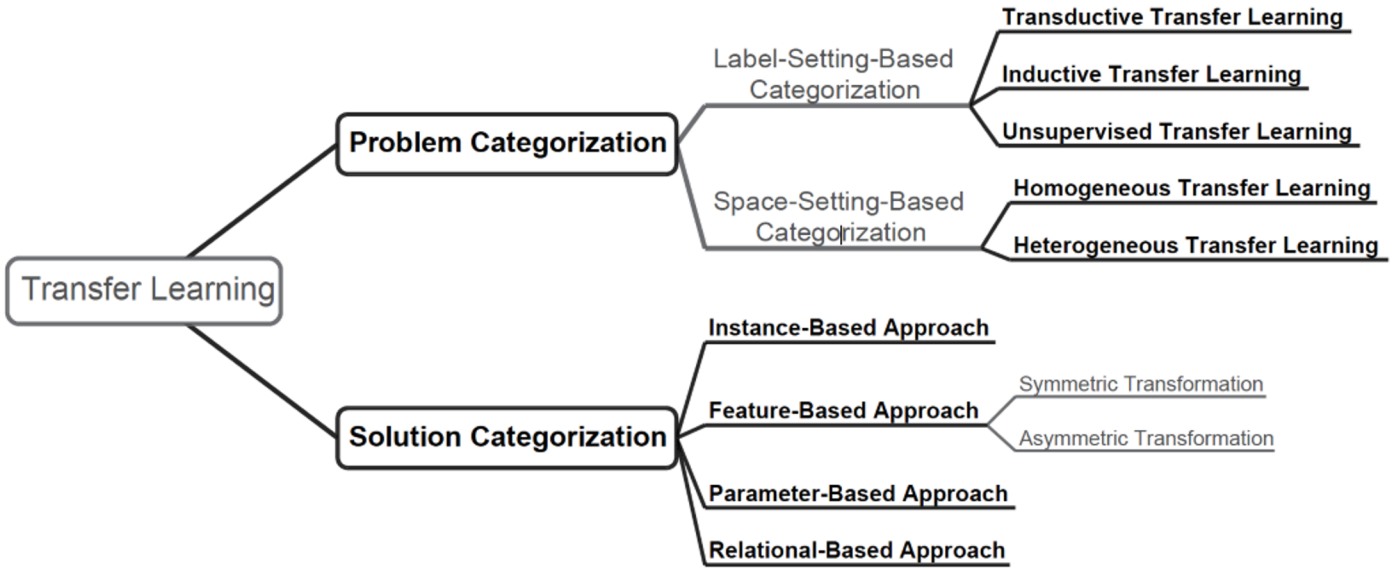
- Jointly learn a group of related tasks
- Reinforces each task by taking advantage of interconnections
- Considers inter-task relevance and inter-task difference – enhances generalization
- MTL vs TL: MTL pays equal attention to each task; TL pays more attention to Target task
- Zhang et al. employs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

Transfer Learning Overview

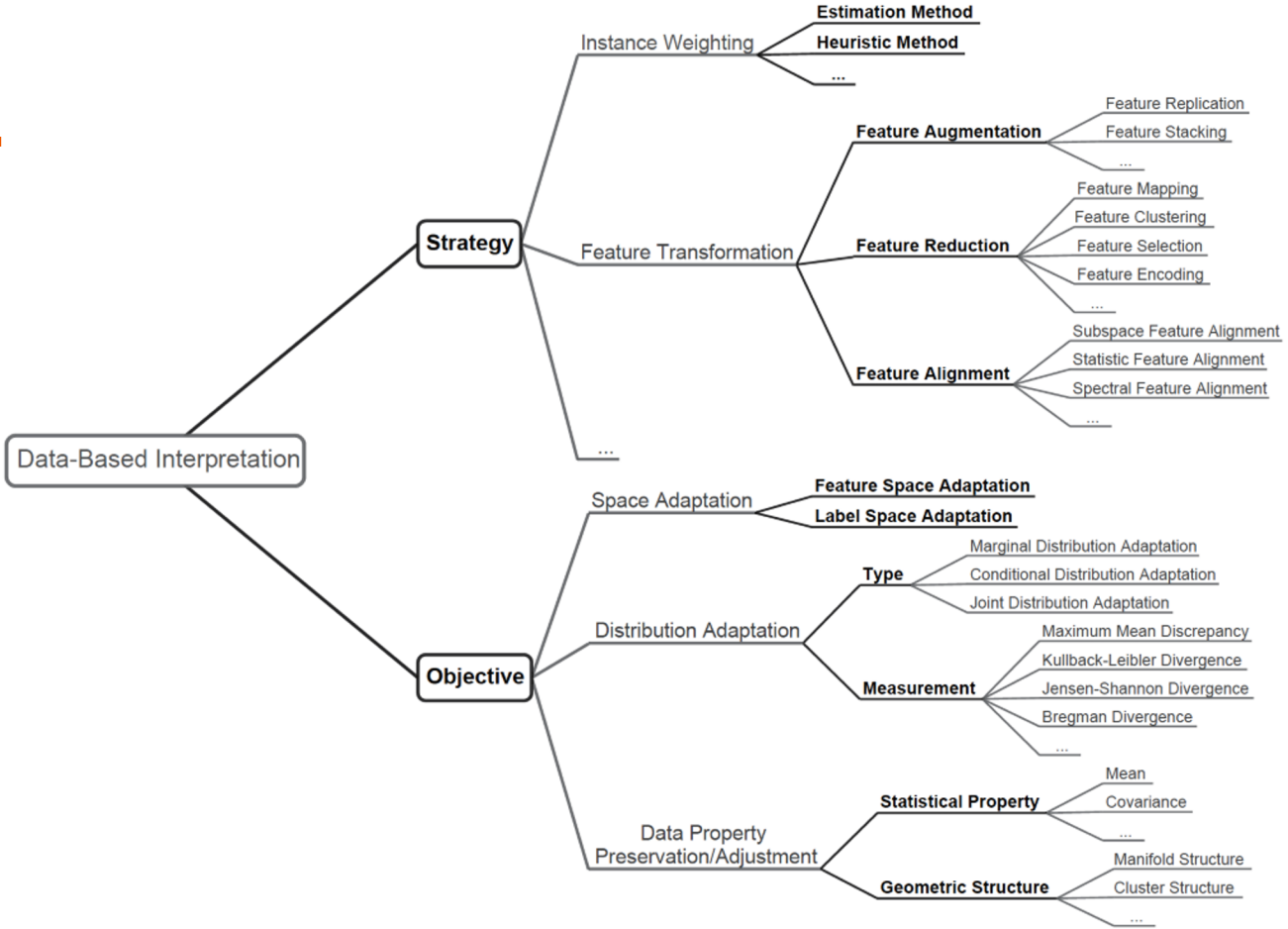
Definitions:

- **Domain**: comprises of feature space ***X*** and a marginal distribution ***P(X)***
- **Task**: consists of label space ***Y*** and a decision function ***f*** to be learned from the data
- **Transfer Learning** will utilize knowledge implied in source domain to improve performance of the learned decision function f^T on the target domain
- **Domain Adaptation**: process of adapting 1 (or more) source domains to transfer knowledge and improve the performance of the target learner

Transfer Learning Categorization



Data Based Interpretation



Data Based Interpretation

Instance Weighting Strategy (Overview)

- Large number of labeled source and a limited number of target domain instances
- Domains differ in only marginal distributions i.e. $P^s(X) \neq P^t(X)$ but $P^s(Y|X) = P^t(Y|X)$
  - Adaptation : assigning weights to the source instances in the loss function
- Weighting Strategy:

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

- Objective:

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

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

Data Based Interpretation

Instance Weighting Strategy (Overview)

- Large number of labeled source and a limited number of target domain instances
- Domains differ in only marginal distributions i.e. $P^s(X) \neq P^t(X)$ but $P^s(Y|X) = P^t(Y|X)$
  - Adaptation : assigning weights to the source instances in the loss function
- Weighting Strategy:

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

- Objective:

difficult to compute

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

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

Data Based Interpretation

Instance Weighting Strategy (estimates for β_i)

Solution #1: Kernel Mean Matching (KMM)

- Estimates β_i's by matching means between the source and target domain instances in a Reproducing Kernel Hilbert Space (RKHS).

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

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

- δ is a small parameter, and B is a parameter for constraint

Data Based Interpretation

Instance Weighting Strategy (estimates for β_i)

Solution #2: Kullback-Leibler Importance Estimation Procedure (KLIEP)

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

Data Based Interpretation

Instance Weighting Strategy (estimates for β_i)

Solution #3: 2-Stage Weighting Framework for Multi-Source Domain Adaptation (2SW-MDA)

- Step 1: Instance Weighting (similar to KMM)
- Step 2: Domain Weighting (weights are assigned to each domain for reducing conditional distribution difference based on smoothness assumption)
- Then, source domain instances are reweighted using instance weights and domain weights. The reweighted instances along with labeled target-domain instances are used to train the classifier

Data Based Interpretation

Instance Weighting Strategy (estimates for β_i)

Solution #4: TrAdaBoost

- Adjust weights iteratively:
  - decrease the weights of the instances that have negative effects on the target learner
- Labeled source-domain and labeled target-domain instances are combined to train the weak classifier, but the weighting operations are different for the source-domain and the target-domain instances

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

Data Based Interpretation

Instance Weighting Strategy (estimates for β_i)

Other solutions:

- 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

Data Based Interpretation

Feature Transformation Strategy (Overview)

- Example: cross domain text classification problem
  - Find latent features (e.g. latent topics) through transformation use them as bridge for knowledge transfer
- Objective of constructing latent space:
  - minimizing marginal and conditional distribution difference **(primary objective)**
  - preserving properties/potential structures of the data,
  - finding correspondence between features
- Operations of feature transformation:
  - **Feature Augmentation** (Feature Replication, Stacking)
  - **Feature Reduction** (Feature Mapping, Clustering, Selection, Encoding)
  - **Feature Alignment**

Data Based Interpretation

Feature Transformation Strategy (Metric)

To measure distribution difference/similarity:

- Maximum Mean Discrepancy (MMD)

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

- Kullback-Leibler Divergence
- Jensen-Shannon Divergence
- Bregman Divergence
- Hisbert-Schmidt Independence Criterion
- And many others

Data Based Interpretation

Feature Augmentation

- **Feature Replication** - Feature Augmentation Method (FAM)
  - Feature space augmented 3x: general, source-specific, & target-specific features
  - For transformed source-domain, target-specific features are set to 0 & vice-versa

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

where Φ_S and Φ_T denote the mappings to the new feature

  - Then, the final classifier is trained on transformed labeled instances
- **Feature Stacking**
  - Problem with FAM: Padding 0 vectors and directly replicating features is less effective when source and target domains have different feature representations
  - Heterogeneous Feature Augmentation (HFA) maps original features into a common features space and then performs a feature stacking operation

Data Based Interpretation

Feature Reduction (with Feature Mapping)

- Traditional ML (PCA, kernel PCA, etc.) focus on data variance, not distribution variances
- **Objective**: find mapping for feature extraction when there is little difference in condition

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

- Find Φ(·) that minimizes the numerator (distance) & maximizes denominator (variance)
- How?
  - First optimize the objective of the numerator and then denominator

Data Based Interpretation

Feature Reduction (with Feature Mapping)

Goal: Find $\Phi(\cdot)$ that minimizes the numerator (distance) & maximizes denominator (variance). 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. Then, use PCA 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
  - solvable in certain conditions – e.g. when mapping is restricted to linear.

Data Based Interpretation

Feature Reduction (with Feature Mapping)

Other approach

- Match the conditional distribution and preserve the structures of the data
- Modified objective function (with additional terms and constraints):

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

- Mapping techniques:
  - Maximum Mean Discrepancy Embedding (MMDE)
  - Transfer Component Analysis (TCA)
  - etc.

Data Based Interpretation

Feature Reduction (with Feature Clustering)

Reduce the feature by finding more abstract representations of original features

Co-Clustering Based Classification (CoCC)

- Source and target document-to-word matrix is co-clustered
- Minimize the joint loss in mutual-information with 2-steps single iteration:
  - Document Clustering – reorder document-to-word matrix for target document
  - Word Clustering – adjust word clusters to minimize joint mutual-information loss

Data Based Interpretation

Feature Reduction (with Feature Clustering)

Reduce the feature by finding more abstract representations of original features

Self-Taught Cluster (STC) - unsupervised

- Does not need label information but the domains should share the same feature clusters in their common feature space
- Minimizes mutual-information loss with 2 steps/iteration:
  - Instance Clustering: clustering updated to minimize respective loss
  - Feature Clustering: feature clusters are updated to minimize the joint loss

Data Based Interpretation

Feature Reduction (with Feature Clustering)

Reduce the feature by finding more abstract representations of original features

Approaches for **Concept-based Transfer Learning Approaches**

- *Latent Semantic Analysis (LSA)*:
  - maps document-to-word matrix to latent space using SVD
- *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 $\mathbf{z}^{d}$ & $\mathbf{z}^{w}$ reflecting concepts behind documents and words

Data Based Interpretation

Feature Reduction (with Feature Clustering)

Reduce the feature by finding more abstract representations of original features

Approaches for **Concept-based Transfer Learning Approaches**

- *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

Data Based Interpretation

Feature Reduction (with Feature Selection)

Reduce the feature by extracting pivot features that behave the same way in different domains. Stability of pivot features helps in serving as bridge for knowledge transfer.

Structural Correspondence Learning (SCL) consists of the following steps:

- Feature Selection:
  - 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

Data Based Interpretation

Feature Reduction (with Feature Encoding)

Use Autoencoder to produce more abstract representation of the input

Stacked Denoising Autoencoder (SDA):

- 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:
  - Training Autoencoder
  - Feature Encoding & Stacking
  - Learner Training
- Challenges: High computational and parameter estimation costs

Data Based Interpretation

Feature Reduction (with Feature Encoding)

Suggested Approaches:

Marginalized Stacked Linear Denoising Autoencoder (mSLDA)

- Adopts linear autoencoder and marginalizes the randomly corrupting step in a closed form to shorter the training time
- Basic architecture: 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]:

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

Data Based Interpretation

Feature Alignment

- While feature augmentation and feature reduction mainly focus on explicit features, feature alignment focus on implicit features
  - explicit features can be aligned to generate new representation
  - implicit features can be aligned to construct a satisfied feature transformation
- Implicit Features:
  - Subspace Features
  - Static Features
  - Spectral Features

Data Based Interpretation

Feature Alignment (Subspace Feature)

Subspace feature alignment process in general:

- *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

Data Based Interpretation

Feature Alignment

1. **Subspace Alignment (SA)**
   - Subspaces generated using PCA; bases obtained by selecting leading eigenvectors
   - Transformation matrix ***W*** is learned to align subspaces
2. **Subspace Distribution Alignment between Two Subspaces (SDA-TS)**
   - Aligns both subspaces and the distributions
3. **Geodesic Flow Kernel (GFK)**
   - based on Geodesic Flow Subspaces integrates infinite number of subspaces located on geodesic curve from the source to the target subspace
4. **Co-Relation Alignment (CORAL)**
   - constructs transformation matrix by aligning the second-order statistic features i.e. covariance matrices

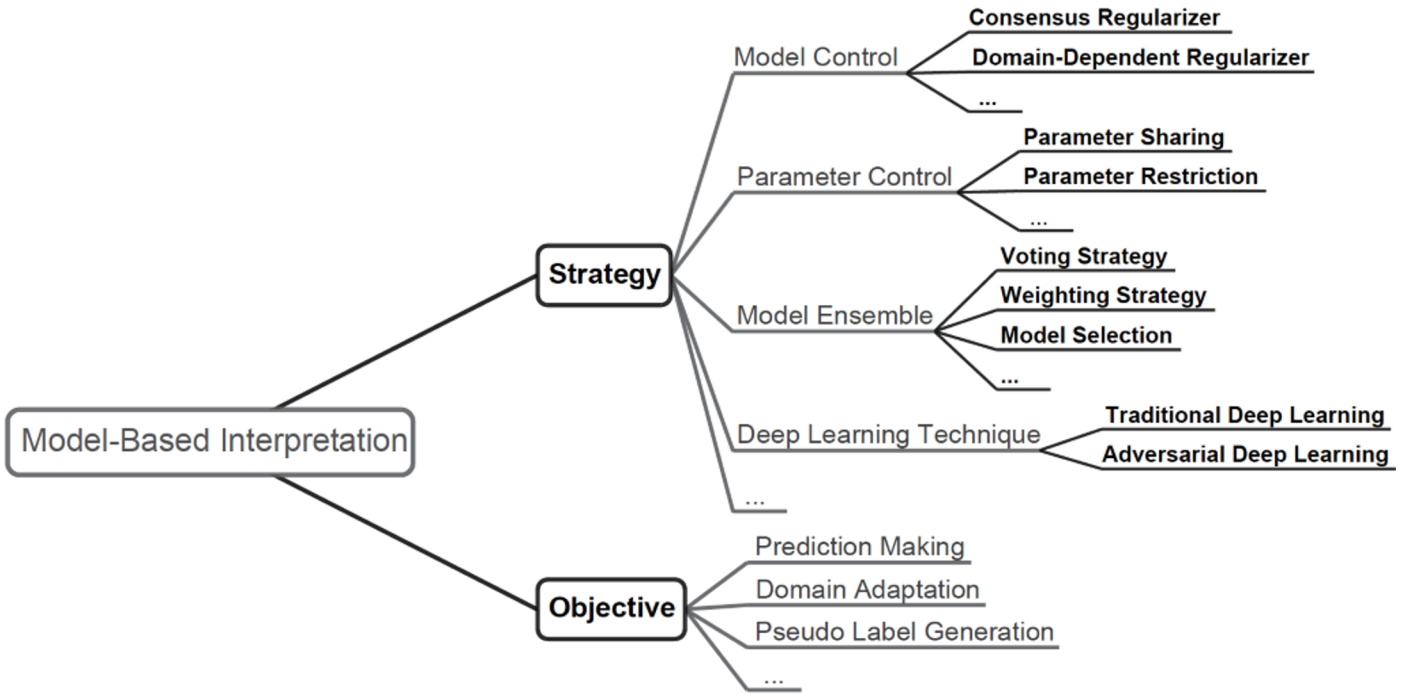
$$W = \arg\min_{W} ||M_S W - M_T||_F^2 = M_S^{\mathrm{T}} M_T,$$

Data Based Interpretation

Feature Alignment

5. **Spectral Feature Alignment (SFA):**
   - Algorithm for sentiment classification based on *spectral clustering* and *feature alignment*
   - SFA generally consists of following 5-steps:
     - **Feature Selection**: Select domain independent/pivot features
     - **Similarity Matrix Construction**: Construct a bipartite graph
     - **Spectral Feature Alignment:** Adapting spectral clustering algorithm
     - **Feature Stacking:** Original features and low dimensional features
     - **Learner Training**: On final representation using labeled instances

Model Based Interpretation



Model Based Interpretation

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) - for multi-source transfer learning
  - Goal: construct a classifier for the target domain with the help of some pre-obtained base classifiers that are respectively trained on multiple source domains
  - Objective:

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

Model Based Interpretation

Parameter Control Strategy

This strategy focuses on the parameters of models.

Parameter Sharing: directly share the parameters of the source learner to the target learner

- Network based parameter sharing:
  - if we have a neural network for the source task, we can freeze (or say, share) 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.
  - Triplex Transfer Learning (TriTL) by Zhuang et al. (an extension of MTrick)

Model Based Interpretation

Parameter Control Strategy

Parameter Restriction

- Unlike *parameter sharing strategy* that enforces the models to share some parameters, *parameter restriction strategy* only requires the parameters of the source and the target models to be similar
- Proposed Approaches:
  - Single-Model Knowledge Transfer (SMKL) by Tommasi et al.
  - Multi-Model Knowledge Transfer (MMKL) by Tommasi et al. (extension of SMKL)

Model Based Interpretation

Model Ensemble Strategy

Aims to combine a number of weak classifiers to make the final predictions. Approaches:

- **TaskTrAdaBoost** (extension of TrAdaBoost for multi-source scenarios):
  - Candidate Classifier Construction
  - Classifier Selection and Ensemble
- **Locally Weighted Ensemble (LWE)**
  - 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
- **Ensemble Framework of Anchor Adapters (ENCHOR)**
  - Constructs a group of weak learners via using different representations of the instances produced by anchors.

Model Based Interpretation

Deep Learning Strategy

- SDA and mSLDA approaches utilize deep learning techniques
- 2 types of approaches
  - **Traditional Deep Learning (non-adversarial)**
  - **Adversarial Deep Learning**

Model Based Interpretation

Traditional Deep Learning

In additional to SDA and mSLDA, there are other reconstruction-based TL approaches:

- Transfer Learning with Deep Autoencoders (TLDA)
  - Adopts 2-autoencoders for the source and target domain (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)

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

Model Based Interpretation

Traditional Deep Learning

- Deep Adaptation Network (DAN)
  - 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

Model Based Interpretation

Traditional Deep Learning

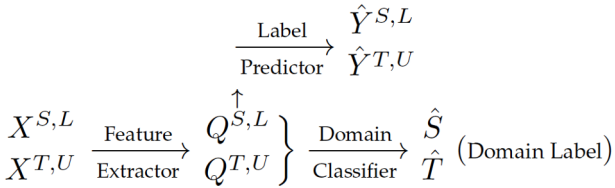
- Multiple Feature Spaces Adaptation Network (MSFAN) by Zhu et al.
  - Deep learning technique for multi-source transfer learning
  - Architecture consists of: Common feature extractor, Domain specific feature extractor, Domain specific classifier
  - Step in each iteration: Common Feature Extraction, Specific Feature Extraction, Data Classification, Parameter Updating
  - 3 main objectives: Classification error minimization, Distribution Adaptation, Consensus Regularization

Model Based Interpretation

Adversarial Deep Learning

Many TL approaches, motivated by GAN, aassumes that a good feature representation contains almost no discriminative information about the instance's original domain

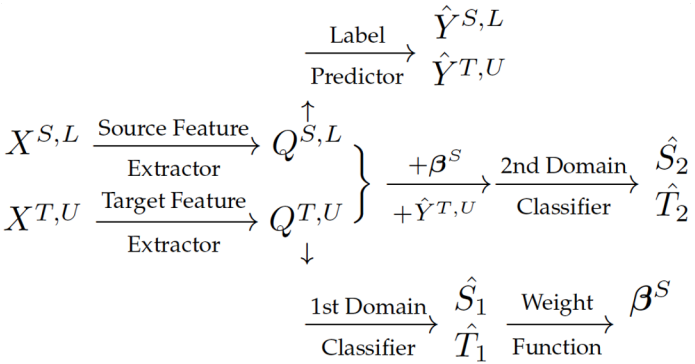
- Deep Adversarial Neural Network (DANN)
  - Assumes no labeled target instances
  - Architecture consists of:
    - feature extractor: acts like generator
    - label predictor: label prediction of the instances
    - domain classifier: acts like discriminator



Model Based Interpretation

Adversarial Deep Learning

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



Application

- **Medical**
  - ML and computer aided diagnosis for medical imaging
- **BioInformatics**
  - Understanding of some organisms can be transferred to other organisms
- **Transportation**
  - Understanding Traffic Image Scenes that suffers from variations due to weather and light conditions
- **Recommender System**
  - Helps in cases where data is sparse (e.g. when data doesn't exist for new users)
- **Communication**
- **Urban Computing**
  - help deal with data scarcity on traffic monitoring, health care, social security, etc

Experiment

2 mainstream research areas:

1. Object Recognition
2. Text Classification

Statistical information of the preprocessed datasets.

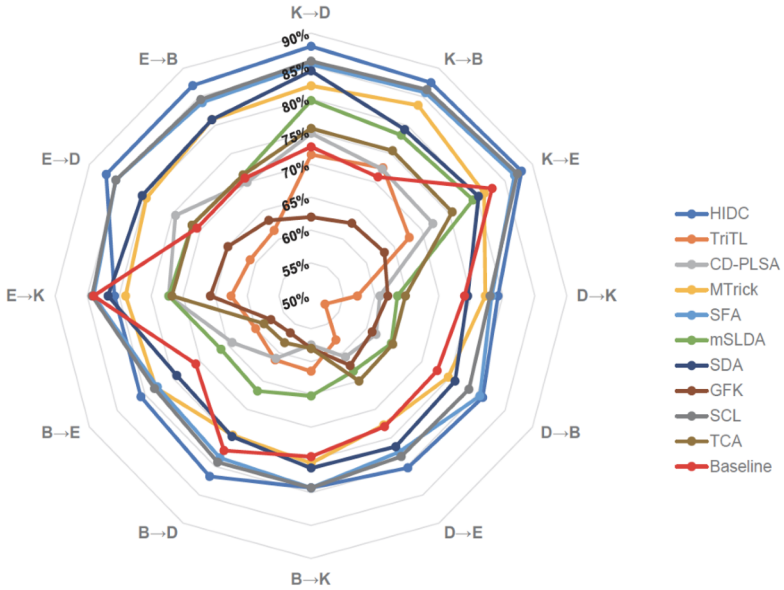
Area	Dataset	Domain	Attribute	Total Instances	Tasks
Sentiment Classification	Amazon Reviews	4	5000	27677	12
Text Classification	Reuters-21578	3	4772	6570	3
Object Recognition	Office-31	3	800	4110	6

Dataset & PreProcessing:

1. **Amazon Reviews:** Multi-domain sentiment dataset from Amazon.com in 4 domains (Books, Kitchen, Electronics, and DVD's)
2. **Reuters-21578** : Text categorization dataset with hierarchical structure
3. **Office-31**: Object recognition dataset with 31 categories and 3 domains (Amazon, Webcam, DSLR)

Experiment Result (Amazon Reviews)

- Most algorithms performed relatively well when the source domain is *electronics / 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)



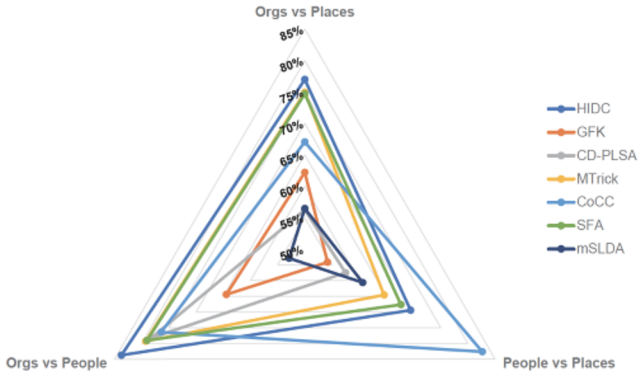
Experiment Result (Amazon Reviews)

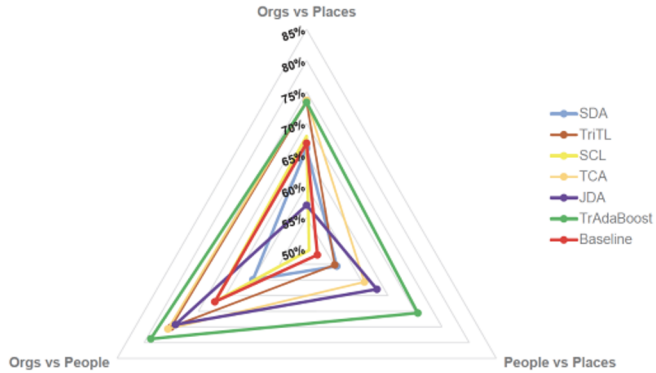
Accuracy performance on the Amazon Reviews of four domains: Kitchen (K), Electronics (E), DVDs (D) and Books (B).

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

Experiment Result (Reuters-21578)

- Most algorithms performed relatively well for Orgs vs People and Orgs vs Places but poor for People vs Places
- Consistent Performance across 3 tasks: HIDC, SFA, MTrick
- Top Performers for People vs Places: TrAdaBoost, CoCC





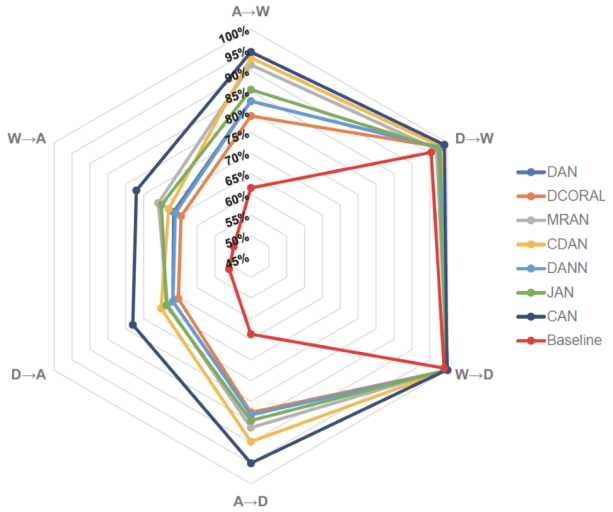
Experiment Result (Reuters-21578)

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

Experiment Result (Office-31)

- 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



Experiment Result (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

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