A Review of Transfer Learning Algorithms

Mohsen Kaboli

A Review of Transfer Learning Algorithms

Mohsen Kaboli

Abstract

This work reviews twenty state-of-the-art papers concerning the topic of visual transfer learning. Special focus lies on algorithms and applications of transfer learning on visual detection and classification. In chapter 1, an overview of transfer learning as well as its applications and general methodology are introduced. Chapter 2 contains brief summaries and comments of each paper.

Contents

Chapter 1

Introduction

In the setting of traditional machine learning, a common assumption is always proposed that the training data and testing data enjoy exactly the same feature space and the same data distributions. However, once a new task arrives, where its data distribution is not identical with the previous one, a new model must be reconstructed from scratch based on the current data. Such methods consume extra efforts and are in most cases very expensive.

Inspired by the fact that human beings are able to intelligently taking advantage of the knowledge being learned in the past when trying to solve a problem they never met before, the idea of transfer learning was raised in order to accelerate the learning process and to obtain better solutions. In contrast to traditional machine learning methods, transfer learning tolerates the difference lying in data distributions and applies the knowledge extracted from other sources to the target task.

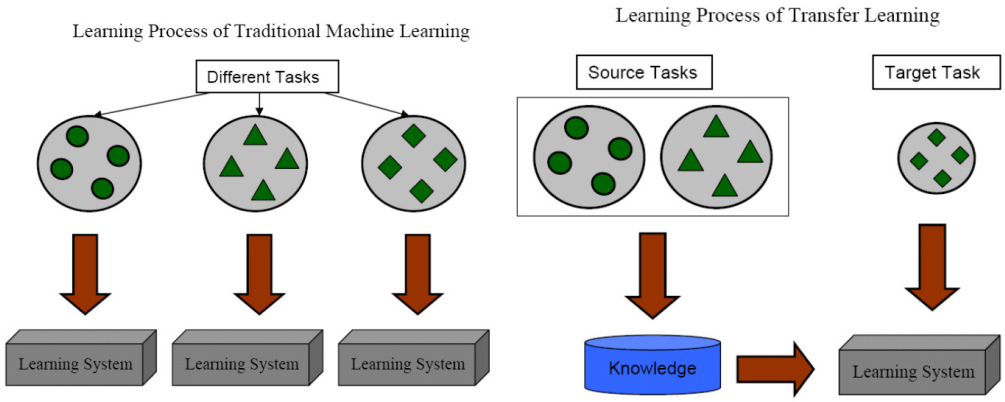


Figure 1.1: Difference between traditional machine learning and transfer learning [11].

1.1 Transfer Learning

Transfer learning (also known as knowledge transfer, learning to learn) refers to a subfield of machine learning. The aim of transfer learning is to learn an objective predictive function for a target task with help of not only the target domain but also from other source domain and source tasks. As shown in Figure 1.1, tasks in traditional machine learning process are independent of each other, and every task should be learned from scratch. However, in transfer learning previous

knowledge extracted from other source tasks can be transferred to the learning process of a new task. Transfer learning can be applied in various scenarios, such as web document classification, indoor WiFi localization problem and sentiment classification etc. [[11]]. In this work the field of visual transfer learning will be focused on.

The research of transfer learning is developed around the following three main problems: what to transfer, how to transfer and when to transfer. Since the source and target data in transfer learning settings can either differ in their tasks or in their domains, there are three different ub-settings of transfer learning: *inductive transfer learning* where the tasks are surly different, regardless of the similarities lying between source and target domain; *transductive transfer learning* where the tasks are same but the feature spaces or marginal probability distributions in both domains are diverse; *unsupervised transfer learning* where there are no labeled data can be utilized for training [[11]].

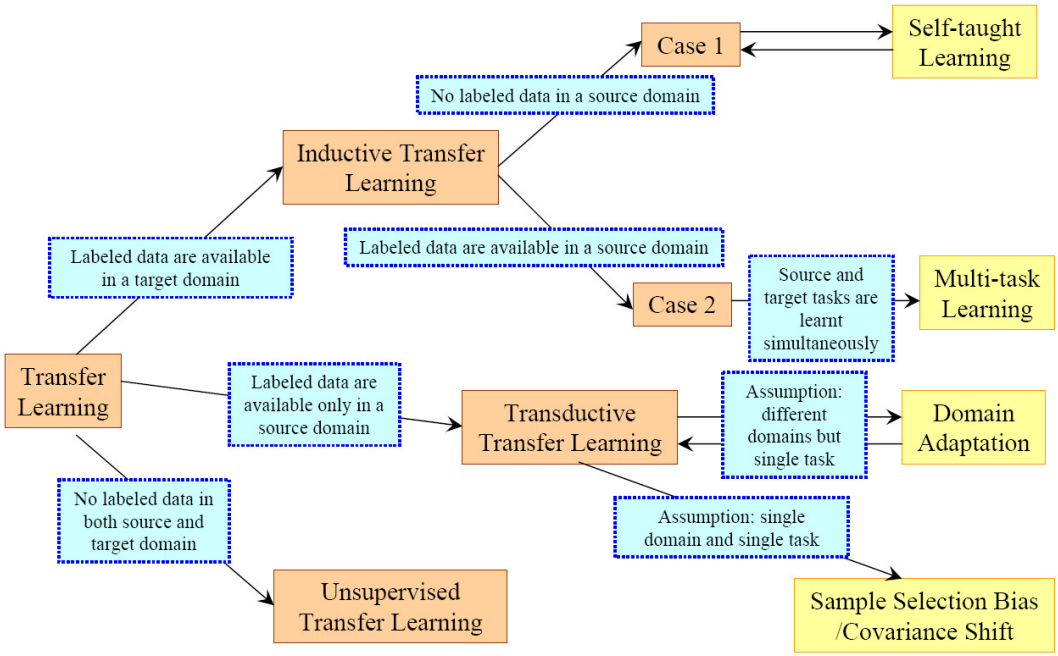


Figure 1.2: The sub-settings of transfer learning [11].

Moreover, as for the categories of knowledge transferred from source domain to target domain, transfer learning approaches can be divided in to 4 types, namely instance-transfer approach, feature-representation-transfer approach, parameter-transfer approach and relational-knowledge-transfer approach.

1.2 Methodology

In order to merge the transfer learning into transfer learning algorithms, there are different methods corresponding to various approaches. Instance-transfer approach and parameter-transfer approach are two most popular approaches applied in the reviewed papers. For instance-transfer approach where the samples or data in source domain are introduced in target domain, an usual method is to re-weight those data before they are reused, so that the effectiveness of the transferred data can be maximized. In the reviewed papers, a LS-SVM based algorithm is frequently proposed for parameter-transfer approach. What the authors commonly do is to rewrite the standard objective

function of LS-SVM by subtracting re-weighted hyperplanes from source domain and by modifying the regularization part.

1.3 Applications

Transfer learning can be applied in various tasks. One of the most investigated applications is classification ([[17]], [[18]], [[5]], [[6]], [[16]], [[19]], [[21]], [[13]], [[1]], [[10]], [[20]], [[14]]). By introducing transfer learning into classification tasks the problem of lacking enough labeled data or data with high quality in target domain can be solved, and the results of classification becomes more reliable. In the rest of reviewed papers, some other practical applications taking advantage of transfer learning are presented, such as feature selection [[8]], [[7]], pedestrian detection [2], improving visual tracking [3] and subtractive bias removal used in medical field [15].

A summary of reviewed papers are listed in Table 1.1.

Paper	**Proposed Algorithms**	**Settings**	**What to Transfer**	**Key Points**	**Applications**
tommasi2009more [17]	*Adapt-W*, *Adapt-2W*	inductive	parameter-transfer	LS-SVM, leave-one-out error, capable of one-shot learning	classification
kuzborskij2015transfer [8]	*GreedyTL*	inductive	parameter-transfer	Hypothesis Transfer Learning (HTL), L2-Regularization	feature selection
kuzborskij2014scalable [7]	*GreedyTL*, *GreedyTL-59*	inducive	parameter-transfer	Hypothesis Transfer Learning (HTL), L2-Regularization, 59-trick	feature selection
gavves2015active [4]	*MCLE*	inductive	parameter-transfer	Query sampling, zero-shot priors	zero-shot classification
tommasi2010safety [18]	*Multi-KT*	inductive	parameter-transfer	LS-SVM, leave-one-out error, learning from multiple models	classification
jie2011multiclass [5]	*MKTL*	inductive	feature-representation-transfer	Multi Kernel Learning (MKL)	classification
kuzborskij2013n [6]	*MULTIpLE*	inductive	parameter-transfer	OVA variant of LS-SVM, leave-one-out error	classification
srivastava2013discriminative [16]	Tree hierarchy	inductive	parameter-transfer	Tree-based priors, Chinese Restaurant Process (CRP)	classification
wang2011dyadic [19]	*DKT*	inductive/ unsupervised	relational-knowledge-transfer/ feature-representation-transfer	Combine both supervised and unsupervised knowledge transfer, nonnegative matrix tri-factorization (NMTF)	classification
zhu2011heterogeneous [21]	*HTLIC*	unsupervised	feature-representation	Two-layer bipartite graph, matrix factorization	classification
patricia2014learning [13]	*H-L2L*	inductive	instance-transfer	Score functions (confidence values)	classification
aytar2011tabula [1]	*A-SVM*, *PMT-SVM*, *DA-SVM*	inductive	parameter-transfer	LS-SVM, projection, deformation, improvement in one-shot learning	classification
long2013transfer [10]	*JDA*	inductive	feature-representation-transfer	Principal Component Analysis (PCA), Maximum Mean Discrepancy (MMD)	classification

wang2015transfer [20]	*TFR*	inductive	feature-representation-transfer	Maximum Mean Discrepancy (MMD), Multi Kernel Learning (MKL)	classification
romera2013transfer [15]	*RMTL*, *MTFL*, *CMTFL*	inductive	relational-knowledge-transfer	Multi Task Learning (MTL)	subtractive bias removal
gao2014transfer [3]	*TGPR*	inductive	instance-transfer	Gaussian Process Regression (GPR)	visual tracking
cao2013transfer [2]	Transfer Learning for Pedestrian Detection	inductive	instance-transfer	Manifold learning, ITLAdaBoost	pedestrian detection
rohrbach2013transfer [14]	*PST*	transductive	feature-representation-transfer	Semi-supervised learning	classification
pan2011transfer [12]	*TCF*	inductive	instance-transfer	Matrix factorization, CMTF, CSVD	zero-shot prediction
lim2011transfer [9]	Borrowing Examples for Multiclass Object Detection	inductive	instance-transfer	Sparse grouped Lasso framework, translation, scaling, affine transformation	object detection

Table 1.1: A brief summary of reviewed papers

Chapter 2

Paper Review

2.1 The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories

Motivation

This paper proposes a SVM-based transfer learning method for visual categorization from only few examples. The algorithms with model adaptation are able to decide automatically from where and how much to transfer. These methods also show a one-shot learning behavior.

Methods

- Basis of proposed methods
  - Least Square-Support Vector Machine (LS-SVM):
    In LS-SVM, the model parameters $(\boldsymbol{w}, b)$ are found by solving

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \quad \textit{subject to} \quad y_i = \boldsymbol{w}\phi(\boldsymbol{x_i}) - b + \xi_i \quad \forall i \in 1, \ldots, l$$

    With help of a criterion error ERR, which is derived from the leave-one-out error, the best learning parameters are those minimizing ERR.
  - Learning a new object category from many samples (*Adapt*):
    The algorithm *Adapt* is based on LS-SVM framework. It takes a known model into consideration and uses a scaling factor β to control the degree to which the new model is close to the old one.

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w} - \beta\boldsymbol{w'}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \quad \textit{subject to} \quad y_i = \boldsymbol{w}\phi(\boldsymbol{x_i}) - b + \xi_i \quad \forall i \in 1, \ldots, l$$

    The resulting model is constructed with pre-trained model scaled by β and the new model built on the new data. β is chosen to be the one producing the lowest ERR.
- Proposed algorithms (*Adapt_W* and *Adapt_2W*)
  - Weighted Error Rate (WERR):
    WERR is extended from the criterion error ERR by introducing the weighting factor ζ_i which is related to the number of positive and negative examples.

– *Adapt_W*:
Adapt_W is obtained by simply substituting ERR with WERR in *Adapt*.

– *Adapt_2W*:
Beside replacing ERR with WERR, the weighting factor ζ_i is also introduced in the model adaptation method:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w} - \beta\boldsymbol{w'}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\zeta_i\xi_i^2 \quad subject\ to \quad y_i = \boldsymbol{w}\phi(\boldsymbol{x_i}) - b + \xi_i \quad \forall i \in 1,\ldots,l$$

– *Adapt_W* and *Adapt_2W* are designed to learn a new object category from few samples.

– LS-SVM and so-called LS-SVM-W can be regarded as the non-adaptive versions of *Adapt_W* and *Adapt_2W* respectively.

Experiments and Results

- Experiments on unrelated categories:

  – Dataset: Caltech-256

  – Objective:
  This set of experiments is designed to observe whether the adaptation model is negatively affected by transferring from unrelated tasks.

  – Results:
  As shown in Figure 2.1, the performance of LS-SVM and *Adapt_W* are nearly identical, as well as LS-SVM-W and *Adapt_2W*, which shows that the adaptation part will not cause negative transfer. Moreover, the performance of *Adapt_2W* is better than that of *Adapt_W*.

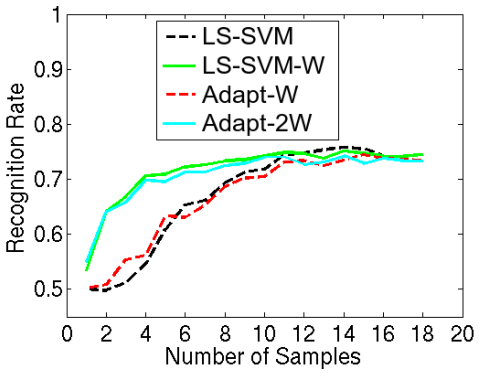


Figure 2.1: Experiments on 3 visually different categories [17].

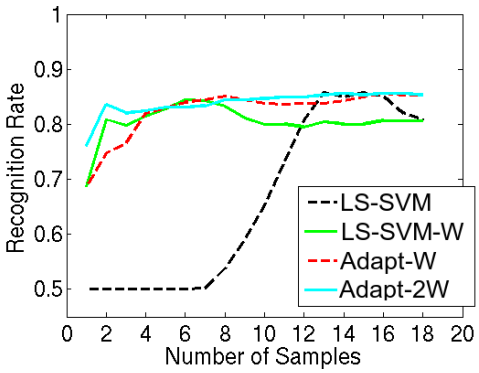


Figure 2.2: Experiments on 3 visually related categories [17].

- Experiments on related categories:

  – Dataset: Caltech-256

  – Results:
  As shown in Figure 2.2, adaptation helps produce better results than starting from scratch.

- Experiments on an increasing number of categories:

- Dataset: IRMA
- Comparison between *Adapt_2W* and LS-SVM-W
- Objective:
  The experiments are designed to observe the one-shot learning performance (how performance varies when the number of known categories grows).
- Results:
  The performance of knowledge transfer method becomes better when the number of known categories increases. Moreover, compared to non-adaptive method, adaptation provides obvious higher recognition rate, especially when the number of samples is low.

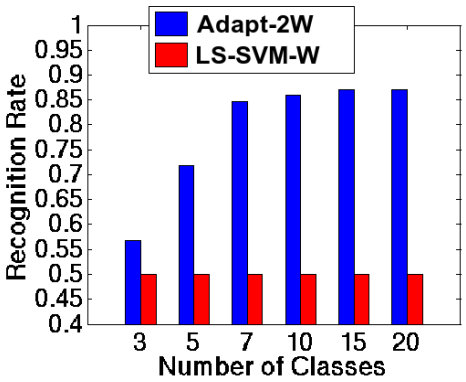


Figure 2.3: One-shot learning performance of the *Adapt_2W* and corresponding LS-SVM-W [17].

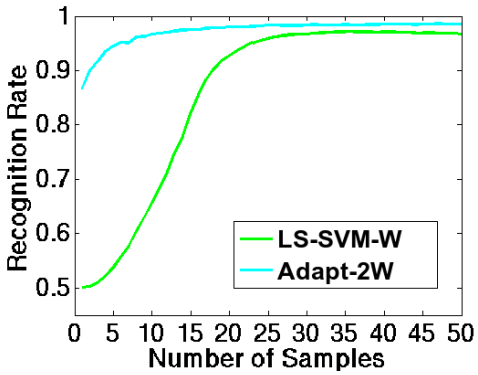


Figure 2.4: Classification performance with respect to the number of training samples (20 categories) [17].

Discussion

- Pros:
  - The adaptation can avoid negative transfer.
  - The adaptation algorithms improve the overall performance in classification.
  - The proposed algorithm is able to perform one-shot learning.
- Cons:
  - The superiority of proposed algorithm exists only in situations where there are few samples of new category. Therefore, for a learning task with sufficient samples of the new category, the proposed algorithm will only lead to higher computational cost.

2.2 Transfer Learning through Greedy Subset Selection

Motivation

In this paper a greedy subset selection algorithm *GreedyTL* is proposed. *GreedyTL*, which is based on Hypothesis Transfer Learning (HTL) framework, is able to select relevant sources and feature dimensions from a large pool. It is assumed that the source data are not directly accessible and that only the source hypotheses trained from those data are available. A L2-regularization variant of the Forward Regression algorithm is applied during k-Source Selection.

Methods

- k-Source Selection:
  - Given the source hypothesis set $\{h_i^{src}\}_{i=1}^n$ and source hypothesis $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, the target hypothesis can be written in the fllowing form:

$$h_{\boldsymbol{w},\boldsymbol{\beta}}^{trg}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \sum_{i=1}^{n} \beta_i h_i^{src}(\boldsymbol{x})$$

  - Considering a binary classification problem, the non-negative loss function $\ell(h(\boldsymbol{x}), y)$ and the empirical risk $\hat{R}(h)$ of a hypothesis h are defined as follows:

$$\ell(h(\boldsymbol{x}), y) = \{(h(\boldsymbol{x}) - y)\}^2$$

$$\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x}), y)$$

  - k-Source Selection:
    Given the source hypothesis set and source hypothesis, k-Source Selection allows to select a subset of size k from n observation variables by solving

$$(\boldsymbol{w}^*, \boldsymbol{\beta}^*) = \underset{\boldsymbol{w},\boldsymbol{\beta}}{\arg\min}\{\hat{R}(h_{\boldsymbol{w},\boldsymbol{\beta}}^{src}) + \lambda\|\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\} \quad s.t. \quad \|\boldsymbol{w}\|_0 + \|\boldsymbol{\beta}\|_0 \leq k\ .$$

    The implementation of L2-regularization improves the generalization ability of empirical risk minimization and the quality of the approximate solution.
  - The k selected elements consist of source hypotheses (selected with $\boldsymbol{\beta}$) and feature dimensions (selected with $\boldsymbol{w}$). Therefore, it is realized that the source hypotheses and feature dimensions can be selected simultaneously.

- Greedy algorithm for k-Source Selection (*GreedTL*):
  - The algorithm *GreedyTL* is derived by extending the Forward Regression algorithm.
  - S is defined to be a set of selected indexes. Initially the set S is empty. During the algorithm, S is populated to size k by selecting the indexes that maximize $\boldsymbol{b}_S^T((\boldsymbol{C} + \lambda\boldsymbol{I})_S^{-1})^T\boldsymbol{b}_S$.
  - The term $\boldsymbol{b}_S^T((\boldsymbol{C} + \lambda\boldsymbol{I})_S^{-1})^T\boldsymbol{b}_S$ is equivalent to the formula presented in k-Source Selection, where $\boldsymbol{C}$ is the covariance matrix of observation random variables and $\boldsymbol{b}$ contains the covariances between predictor random variables and the training data.

Experiments and Results

- Datasets: subsets of Caltech-256, Imagenet and SUN09
- A linear SVM is chosen to train the source classifiers.
- Results:
  As shown in Figure 2.5 and Figure 2.6, it is clear that the performance of proposed algorithm *GreedyTL* is better than other transfer learning and feature selection baselines in most cases, which confirms the effect of L2-regularization as well as the robustness of *GreedyTL*.

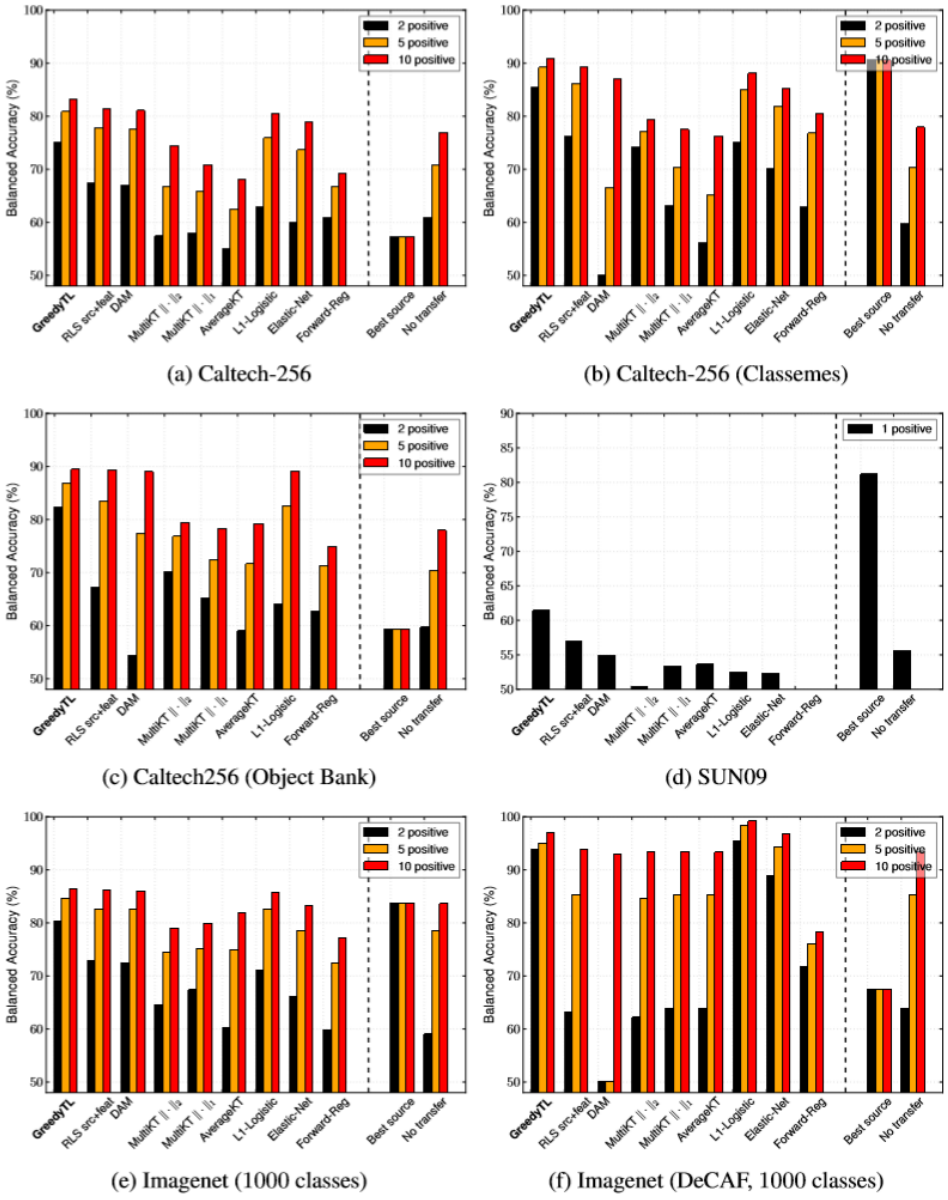


Figure 2.5: Performance of different baselines on Caltech-256, Imagenet and SUN09 [8].

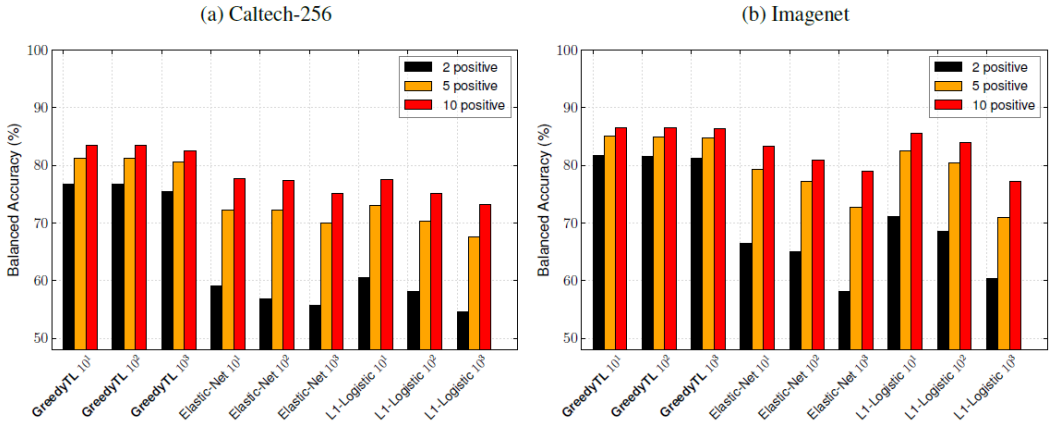


Figure 2.6: Baselines and number of additional noise dimensions sampled from a standard distribution [8].

Discussion

The proposed greedy algorithm aims at selecting the most relevant information from the source hypotheses, especially for large data collections. By introducing the L2-regularization the scalability and capability of the k-Source Selection method.

2.3 Scalable Greedy Algorithms for Transfer Learning

Motivation

The proposed algorithm *GreedyTL* in this paper is developed from Hypothesis Transfer Learning (HTL) algorithm and is based on the assumption that one does not have direct access to the source data, but rather the source hypotheses trained from them. *GredyTL* is able to select relevant source hypotheses and feature dimensions from a large pool at the same time. A randomized variant of this greedy algorithm is also proposed in order to further reduce the computational cost without reducing the performance of*GreedyTL*.

Methods

- k-Source Selection:
  - Given the source hypothesis set $\{h_i^{src}\}_{i=1}^n$ and source hypothesis $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, k-Source Selection allows to select a subset of size k from n observation variables by solving

$$(\boldsymbol{w}^*, \boldsymbol{\beta}^*) = \arg\min_{\boldsymbol{w},\boldsymbol{\beta}} \{\hat{R}(h^{src}_{\boldsymbol{w},\boldsymbol{\beta}}) + \lambda\|\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\} \quad s.t. \quad \|\boldsymbol{w}\|_0 + \|\boldsymbol{\beta}\|_0 \leq k \; .$$

  - In total the number of selected source hypotheses $\|\boldsymbol{w}\|_2^2$ and feature dimensions $\|\boldsymbol{\beta}\|_2^2$ is k. That is, k-Source Selection is able to select source hypotheses and feature dimensions at the same time.

- Greedy algorithm for k-Source Selection (*GreedTL*):
  - The algorithm *GreedyTL* is derived by extending the Forward Regression algorithm.
  - S is defined to be a set of selected indexes. Initially the set S is empty. During the algorithm, S is populated to size k by selecting the indexes that maximize $\boldsymbol{b}_S^T((\boldsymbol{C} + \lambda\boldsymbol{I})_S^{-1})^T\boldsymbol{b}_S$.
  - The term $\boldsymbol{b}_S^T((\boldsymbol{C} + \lambda\boldsymbol{I})_S^{-1})^T\boldsymbol{b}_S$ is equivalent to the formula presented in k-Source Selection, where $\boldsymbol{C}$ is the covariance matrix of observation random variables and $\boldsymbol{b}$ contains the covariances between predictor random variables and the training data.

- Approximated randomized greedy algorithm (*GreedyTL-59*):
  - Objective:
    This approximated algorithm is designed to eliminate the computational cost introduced by the searching process for populating S. This is achieved by approximating this search with a randomized strategy.
  - Theoretical basis (Theorem 1):
    Denote by $M := \{x_1, \ldots, x_m\} \subset \mathbb{R}$ a set of cardinality m, and by $\tilde{M} \subset M$ a random subset of size $\tilde{m}$. Then the probability that max $\tilde{M}$ is greater than or equal to n elements of M is at least $1 - (\frac{n}{m})^{\tilde{m}}$.
  - Application (59-trick):
    If one desires values better than 95% of all other estimates with 1-0.05 probability, then 59 samples are sufficient.
    The searching process becomes a search for the maximum over a random set of size 59.

Experiments and Results

- Datasets: subsets of Caltech-256,Imagenet, SUN09 and SUN-397
- A linear SVM is implemented to train the source classifiers.
- Results of comparing *GreedyTL* to other baselines:
  As shown in Figure 2.5 and Figure 2.6, it is clear that the performance of proposed algorithm *GreedyTL* is better than other transfer learning and feature selection baselines in most cases, which confirms the effect of L2-regularization as well as the robustness of *GreedyTL*.
- Approximated *GreedyTL*:
  - Datasets: Imagenet and SUN-397
  - Results: As shown in Figure 2.7, the performance of approximated algorithm is similar to that of *GreedyTL*, without losing accuracy.

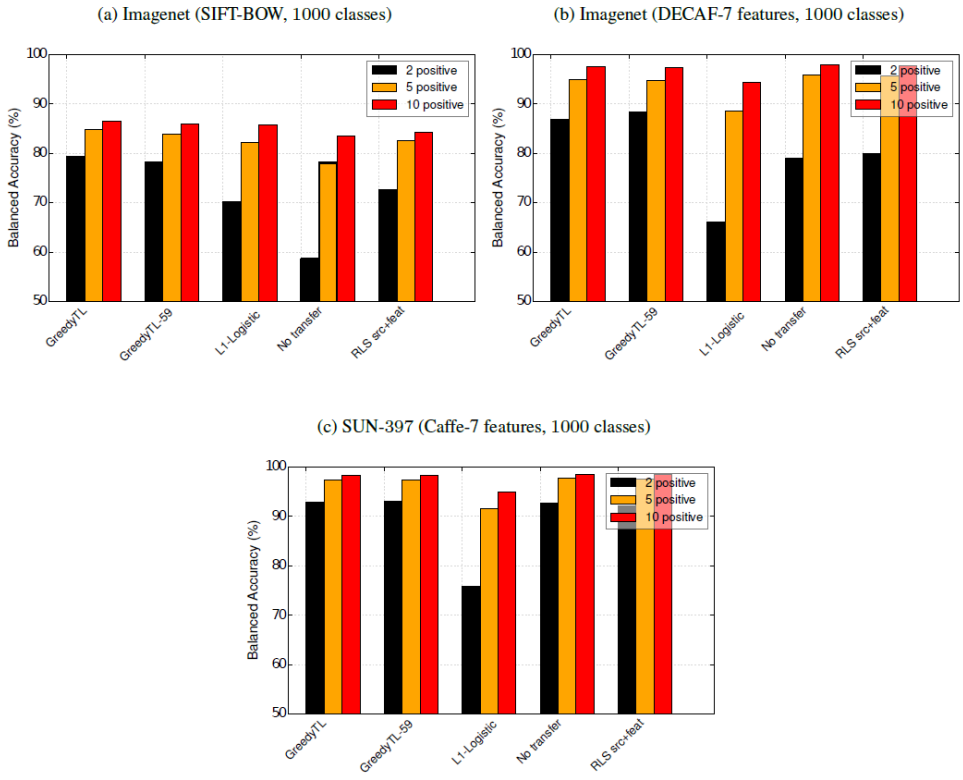


Figure 2.7: *GreedyTL-59* compared to *GreedyTL* and other most powerful algorithms on three datasets [7].

Discussion

The proposed greedy algorithm is useful in the cases where the source data are not directly accessible. The introduction of L2-regularization ensures the generalization ability and quality of approximate solution. The approximated greedy algorithm *GreedyTL-59* applies the 59-trick and reduces the computational cost while keeping the performance unchanged.

2.4 Active Transfer Learning with Zero-Shot Priors: Reusing Past Datasets for Future Tasks

Motivation

In this paper zero-shot classifiers are used to provide priors for active learning even when the known datasets and the new tasks are unrelated. For the query sampling procedure two conditions (maximum conflict and label equality) are proposed to achieve efficient and optimal sampling.

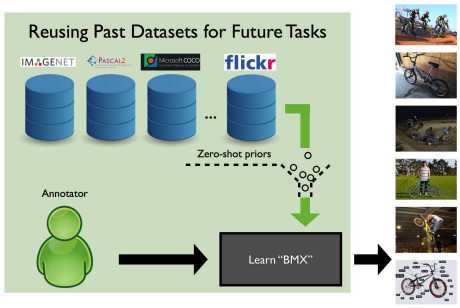


Figure 2.8: An example for learning a new classifier for a category that is absent from the existing dataset [4].

Methods

- Auxiliary zero-shot active learning
  - Objective:
    To find out the most informative instances by querying an oracle and use them to build a classification model with limited time and annotation budget.
  - Maximum Conflict - Label Equality
    * The dual objective function at time t can be written as:

$$\max_{\alpha^t,\gamma^t} \sum_i \gamma_i^t \lambda_i^t \alpha_i^t - \frac{1}{2} \sum_{i,j} \alpha_i^t \alpha_j^t \gamma_i^t \gamma_j^t y_i y_j \boldsymbol{x}_i \boldsymbol{x}_j$$
$$s.t. \quad \sum_i \gamma_i^t \alpha_i^t y_i = 0$$
$$0 \leq \alpha_i^t \leq, \ \forall i$$
$$\gamma_i^t \geq \gamma_i^{t-1}, \ \forall i$$
$$\sum_i \gamma_i^t = \sum_i \gamma_i^{t-1} + B$$

      where $\gamma_i^t \in \{0,1\}$ indicates whether at time step t the label y_i has been queried; α_i^t represents the Lagrange multipliers; B restricts the maximum annotation budgt per iteration.
    * Maximum Conflict:
      The first condition of maximum conflict requires that the sample $i^\star$ should be queried such that its label $y_{i^\star}$ has an opposite sign from its classification score ai $(t-1)$. That is, the Lagrange multipliers α_i should be set large if the model $\boldsymbol{x}_i$ is misclassified; Otherwise α_i can be close or equal to 0 for corectly classified models.

    * Label Equality:
      In order to respect the constraint $\sum_i \gamma_i^t \alpha_i^t y_i = 0$, the number of positive and negative examples in the training set should be balanced.
    * These two conditions can be applied in querying procedures in order to obtain an optimal sampling.
  - Zero-shot priors:
    The zero-shot prediction of linear classification models can be written as $f^{zs}(\boldsymbol{x}) = \sum_{k \in \mathcal{K}} \beta_{ck} \boldsymbol{w}_k \boldsymbol{x}_i$, where β_{ck} chooses the known learning parameters $\boldsymbol{w}_k$ and transfers them to the new model with different weightings. Moreover, the prediction score of active learning can be modified to

$$f^t(\boldsymbol{x}) = \eta^t f^{zs}(\boldsymbol{x}) + \boldsymbol{w}^t \boldsymbol{x}.$$

- Query sampling procedure:
  - Sampling from different feature space zones:
    The sampling regions for SVM classifiers can be divided into 3 zones, namely negative outer margin zone $\mathcal{F}_-$ ($f^{t-1}(\boldsymbol{x}) < -1$), margin-hyperplane zone $\mathcal{F}_0$ ($-1 < f^{t-1}(\boldsymbol{x}) < 1$) and positive outer margin zone $\mathcal{F}_+$ ($f^{t-1}(\boldsymbol{x}) > 1$).
    Theoretical analysis shows that sampling from the positive outer margin zone results in the fastest learning in the first rounds.
  - Maximum conflict - label equality sampling:
    The proposed MCLE sampling relies on the likelihoods of sampling from $\mathcal{F}_+$ and $\mathcal{F}_0$, in order to satisfy the two conditions at the same time. Instances are sampled from $\mathcal{F}_+^{t-1}$ if there are too many negative ones, otherwise examples are sampled from $\mathcal{F}_0^{t-1}$.

Experiments and Results

- Datasets: Hierarchical SUN (HSUN) dataset and Microsoft COCO (MCOCO) dataset
- Zero-shot priors for active learning:
  After comparing different prior strategies and zero-shot models, the constant prior where $\eta^t = 1,\ \forall t$ is the fastest learner. Besides, the COSTA and Image search priors perform the best for HSUN and MSCOCO, respectively. These strategies and models are used in the following experiments.
- Maximum conflict - label equality:
  As show in Figure 2.9, comparing to the methods sampling only from one zone ($\mathcal{F}_+$ or $\mathcal{F}_0$), the proposed MCLE sampling can adaptively sample from the two zones and presents better or equal performance than its competitors

Discussion

- Pros:
  - It is creative to combine the zero-shot learning with active learning and use the zero-shot classifiers as priors to guide the learning.
  - The MCLE conditions provide the requirements for optimal sampling procedure, which introduce flexibility and efficiency to the sampling.
- Cons:

– If the system is very sensitive to additional computational costs, the introduction of this adaptive sampling strategy may be evaluated in advance.

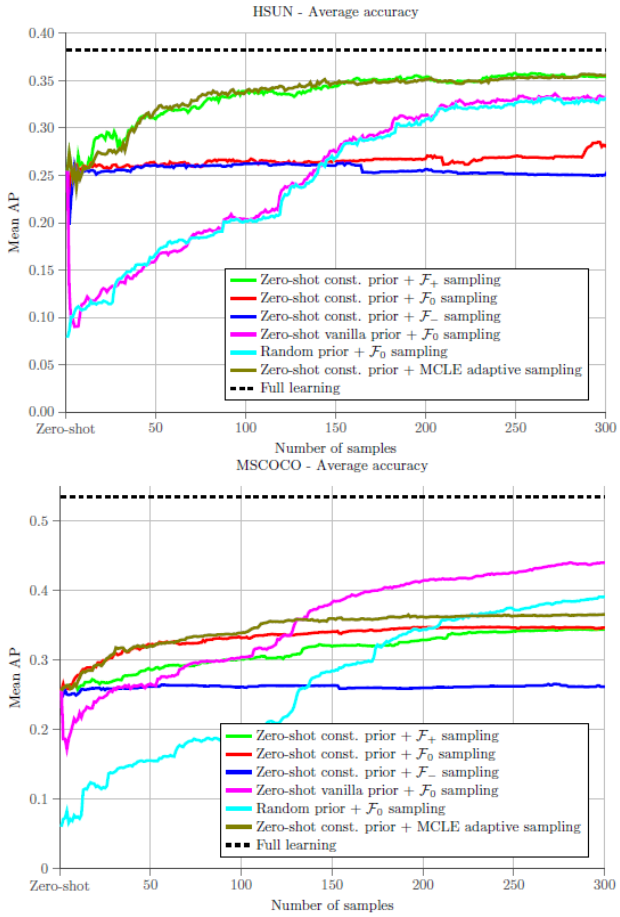


Figure 2.9: The adaptive MCLE sampling strategy compared to other strategies on HSUN and MSCOCO [4].

2.5 Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer

Motivation

In this paper another SVM-based transfer learning algorithm *Multi-KT* is proposed, which is able to appropriately transfer prior knowledge from multiple learned categories. Moreover, when the prior knowledge grows, *Multi-KT* presents a one-shot learning behavior.

Methods

- Basis of the proposed algorithm:
  - *Multi-KT* is based on the weighted adaptation transfer algorithm illustrated in [17], where the model parameters $(\boldsymbol{w}, b)$ are obtained by solving

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w} - \beta\boldsymbol{w'}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\zeta_i[y_i - \boldsymbol{w}\phi(\boldsymbol{x}_i) - b]^2.$$

  - This algorithm *Single-KT* is able to transfer the parameters of an old model, but it can choose only one model.
- Multi model knowledge transfer (*Multi-KT*):
  - Objective: To design a transfer learning algorithm which is able to select more than one learned models and transfer them properly to the learning of a new category.
  - It is extended from the previous adaptation model by substituting β with a vector $\boldsymbol{\beta}$ of size k:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\left\|\boldsymbol{w} - \sum_{j=1}^{k}\beta_j\boldsymbol{w}'_j\right\|^2 + \frac{C}{2}\sum_{i=1}^{l}\zeta_i[y_i - \boldsymbol{w}\phi(\boldsymbol{x}_i) - b]^2$$

  - The optimal solution of $\boldsymbol{w}$ consists of a new model built on the incoming data and a weighted combination of old models.

Experiments and Results

- Datasets: subsets of Caltech-256
- Experiments on related/unrelated prior knowledge:
  - Objective:
    This set of experiments is designed to study how *Multi-KT* chooses the reliable prior knowledge and its impact on performance.
  - Results:
    The results are shown in Figure 2.10 and Figure 2.11.
    - For related classes:
      There is no obvious difference when the proposed algorithm is compared to other two knowledge transfer methods. However, their performances are much better than learning from scratch.
    - For mixed classes:
      In the given mixed categories parts of the data are related while others are unrelated. It can be concluded from Figure 2.11 that *Mult-KT* performs better than *Average-KT* and *Single-KT*.

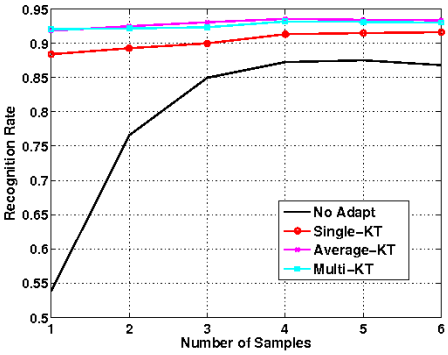


Figure 2.10: Experiments on related categories [18].

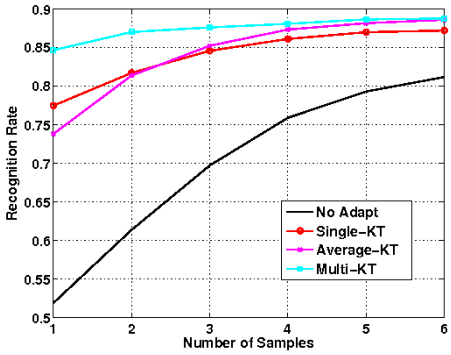


Figure 2.11: Experiments on mixed categories [18].

- Experiments on increasing prior knowledge:
  - Objective:
    The experiments are designed to study how the performance varies when the number of known categories grows.
  - Results:
    As show in Figure 2.12, when start learning from only few classes, *Multi-KT* outperforms its competitors.

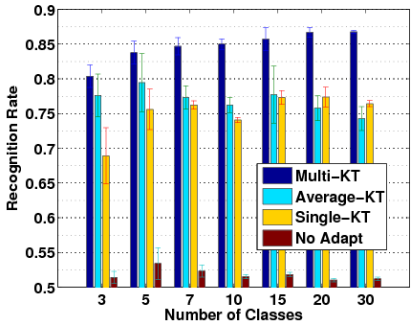


Figure 2.12: Experiments on increasing number of categories to show the ability of one-shot learning [18].

Discussion

The proposed algorithm *Multi-KT* can take advantage of multiple prior models and can automatically decides from where and how much to transfer. Some key points of it are listed as follows:

- How to transfer:
  Multi-KT is based on LS-SVM and learns the new classes through adaptation.
- What to transfer:
  The transferred knowledge is the hyperplanes of the classifiers of known classes.
- When to transfer:
  If the transferred knowledge would result in negative transfer, the transfer might be disregarded completely.

2.6 Multiclass Transfer Learning from Unconstrained Priors

Motivation

The multiclass transfer learning algorithm *MKTL* is based on Multi Kernel Learning algorithm and combines it with transfer learning techniques. This algorithm is able to use different types of feature representations and learning methods as prior knowledge.

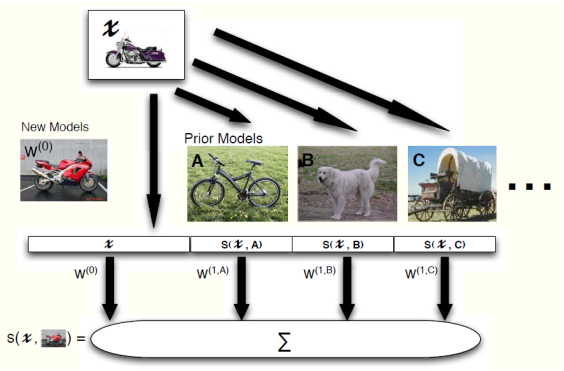


Figure 2.13: Graphical illustration of using the outputs from the prior models as auxiliary features [5].

Methods

- Problem definition:
  - Task description:
    The main purpose is to learn a new classifier for F' categories, given F categories already known.
  - Score function:
    The score function $s(\boldsymbol{x}, y)$ represents the value or confidence of an instance $\boldsymbol{x}$ to be assigned to class y. The general form of a score function can be written as:

$$s(\boldsymbol{x}, y) = \boldsymbol{w}^{(0)}\phi^{(0)}(\boldsymbol{x}, y) + \sum_{z=1}^{F} \boldsymbol{w}^{(y,z)}\phi^{(y,z)}(s_p(\boldsymbol{x}, z), y)$$

    where
    $\boldsymbol{w}^{(\cdot)}$ is a hyperplane;
    $\bar{\boldsymbol{w}}$ represents the concatenation of various $\boldsymbol{w}^{(\cdot)}$;
    $\boldsymbol{w}^{(y,z)}$ represents contribution of the z-th prior model in predicting that $\boldsymbol{x}$ is assigned to class y;
    $\phi^{(\cdot)}(\cdot, \cdot)$ maps the samples into space with higher dimensions.
  - Objective function:
    The objective function is made up of a regularization of the combination of models $\bar{\boldsymbol{w}}$ as well as a convex loss function. With help of objective function the optimal $\bar{\boldsymbol{w}}$ can be obtained.

$$\min_{\bar{\boldsymbol{w}}} \Omega(\bar{\boldsymbol{w}}) + C \sum_{i=1}^{N} \ell(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i)$$

- Multi kernel transfer learning (*MKTL*):

  – Multi kernel learning:
  The multi kernel learning algorithm is extended from then objective function by using a l_p norm regularization for $\bar{\boldsymbol{w}}$.

$$\min_{\bar{\boldsymbol{w}}} \|\bar{\boldsymbol{w}}\|_{2,p}^2 + C \sum_{i=1}^{N} \ell(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i)$$

  For binary cases, the loss function is chosen to be $\ell^{HL}(\bar{\boldsymbol{w}}, \boldsymbol{x}, y) = \mid 1 - y\bar{\boldsymbol{w}} \cdot \bar{\phi}(\boldsymbol{x}) \mid_+$.
  For multiclass cases, the loss function is chosen to be $\ell^{MC}(\bar{\boldsymbol{w}}, \boldsymbol{x}, y) = \max_{y' \neq y} \mid 1 - \bar{\boldsymbol{w}} \cdot (\bar{\phi}(\boldsymbol{x}, y) - \bar{\phi}(\boldsymbol{x} - y')) \mid_+$.

  – Multi kernel transfer learning:
  In order to realize the transfer learning, the forms of $\bar{\boldsymbol{w}}$ and $\bar{\phi}(\boldsymbol{x}, y)$ are chosen as follows:

$$\bar{\boldsymbol{w}} = [\boldsymbol{w}^{(0)}, \boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(y,z)}, \ldots, \boldsymbol{w}^{(F',F)}]$$

$$\bar{\phi}(\boldsymbol{x}, y) = [\phi^{(0)}, \phi^{(1,1)}(s_p(\boldsymbol{x}, 1), y), \ldots, \phi^{(y,z)}(s_p(\boldsymbol{x}, z), y), \ldots, \phi^{(F',F)}(s_p(\boldsymbol{x}, F), y)]$$

  The *MKTL* problem can be solved with the off-the-shelf OBSCURE framework.

Experiments and Results

- Datasets: subsets of the Caltech-256 and the Animals with Attributes (AwA) dataset

- Experiments are conducted for both binary and multiclass transfer learning.

- Results:
  The results for the binary and multiclass cases are shown in Figure 2.14 and Figure 2.15 respectively. It is clear to conclude that the performance of proposed *MKTL* is better than previously proposed transfer learning algorithms, especially for the weighted *MKTL* which takes into account the unbalance of positive and negative samples.

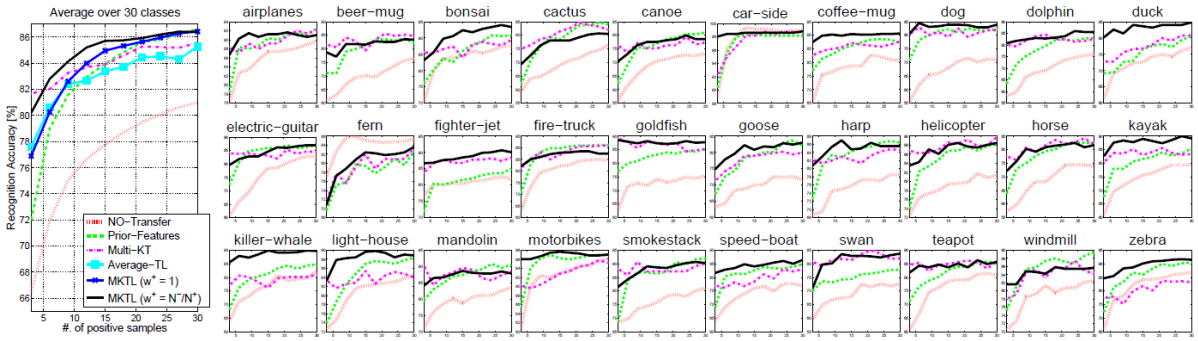


Figure 2.14: Experiments on binary cases with the average behavior of the 30 categories on the left [5].

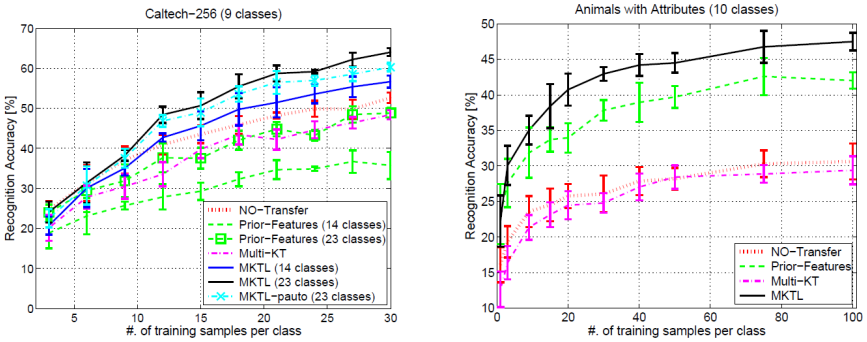


Figure 2.15: Experiments on multiclass cases [5].

Discussion

- Pros:
  - The proposed algorithm *MKTL* is more flexible because it can learn from different features and different learning methods.
  - More than only one prior models can be transferred to the new model.
- Cons:
  - Although the computational cost is under control of the *MKL* solver, the procedure to construct the *MKTL* model is more complex than other algorithms.

2.7 From N to N+1: Multiclass Transfer Incremental Learning

Motivation

This paper propose a multiclass transfer learning method *MULTIpLE* based on Least Squares-Support Vector Machine. During the design not only the building for a new model with help of transferred knowledge is considered, but the preserving of the performance of the existing hyperplanes is also focused on. One-Versus-All (OVA) variant of LS-SVM instead of minimizing the Leave-One-Out error is used to optimize the transfer coefficient $\boldsymbol{\beta}$.

Problem Setting and Definitions

- Definitions:
  Each column of matrix $\boldsymbol{W} = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_N]$, namely $\boldsymbol{W}_n$, represents a hyperplane of LS-SVM;
  $\boldsymbol{Y}$ is a label matrix where its label $\boldsymbol{Y}_{in} = 1$ if $y_i = 1$, and -1 otherwise.

- General form of a multiclass LS-SVM objective function:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{2}\|\boldsymbol{W}\|_F^2 + \frac{C}{2}\sum_{i=1}^{M}\sum_{n=1}^{N}\left(\boldsymbol{W}_n^T\boldsymbol{x}_i + b_n - Y_{in}\right)^2$$

Methods

- Objectives:
  - Transfer learning problem:
    Transfer learning can be realized by adding the term $\|\boldsymbol{W}_{N+1} - \boldsymbol{W}'\boldsymbol{\beta}\|^2$ into the objective function.
  - Avoiding degradation of $\boldsymbol{W}'$:
    By introducing the term $\|\boldsymbol{W} - \boldsymbol{W}'\|^2$ to the objective function can help keep the new hyperplanes $\boldsymbol{W}$ close to the old ones.

- Objective function for *MULTIpLE*:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{W}'\|_F^2 + \frac{1}{2}\|\boldsymbol{W}_{N+1} - \boldsymbol{W}'\boldsymbol{\beta}\|_F^2 + \frac{C}{2}\sum_{i=1}^{M}\sum_{n=1}^{N}\left(\boldsymbol{W}_n^T\boldsymbol{x}_i + b_n - Y_{in}\right)^2$$

Experiments and Results

- Datasets: subsets of the Caltech-256 and the Animals with Attributes (AwA) dataset

- Selected algorithms are divided into two groups: no transfer baselines and transfer baselines

- Results:
  The results for experiments conducted on Caltech-256 is shown in Figure 2.16, and experiments on the AwA dataset provide similar results. *MULTIpLE* outperforms other transfer baselines.

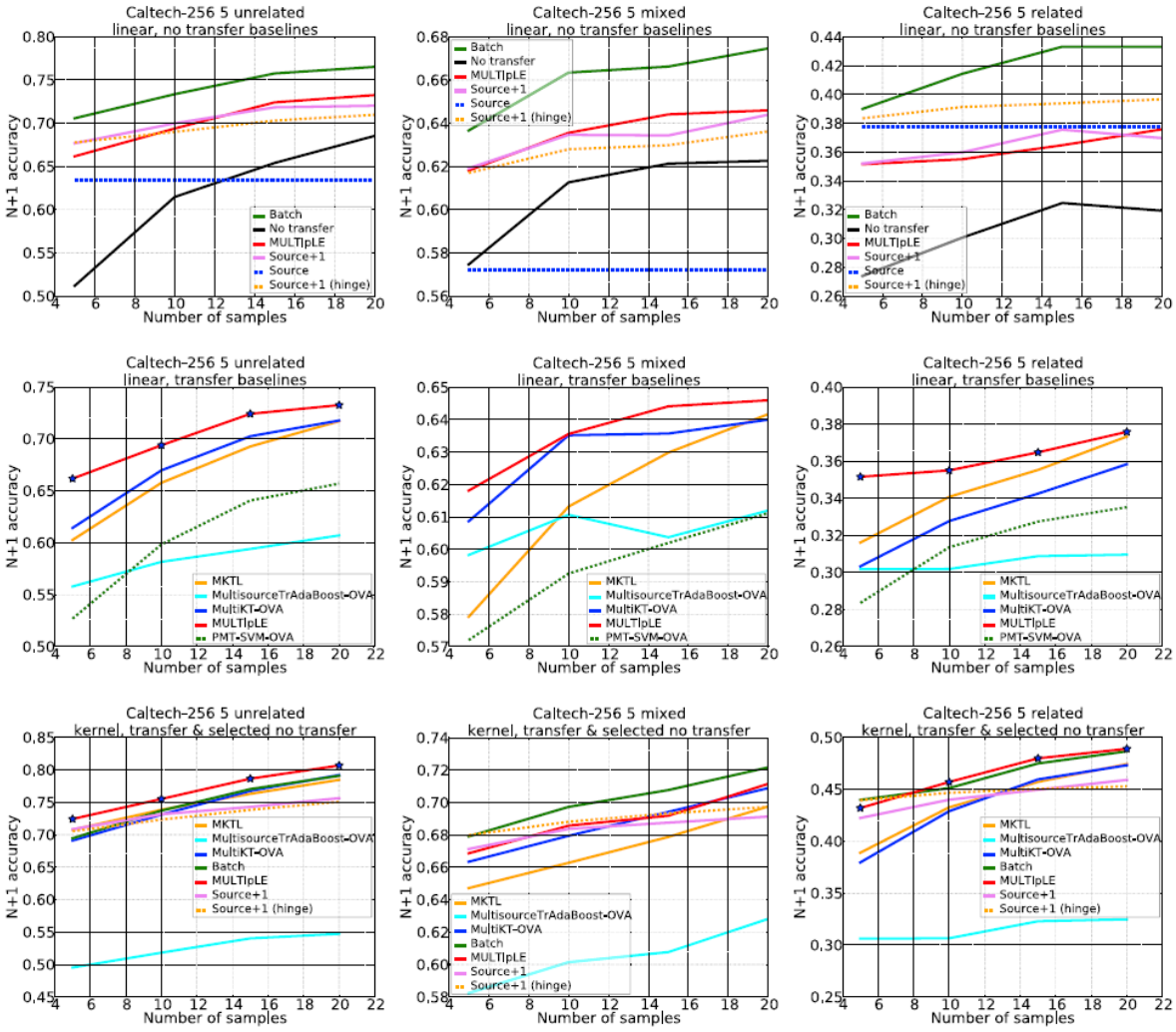


Figure 2.16: Experiments on Caltech-256 for each group of baselines and with unrelated, mixed and related categories, respectively [6].

Discussion

One of the advantages of *MULTIpLE* is that it also consider the influence on the performances of the old models, which ensures the overall high performance of the models. The ability to transfer multiple models provides higher probability for optimal transfer learning.

2.8 Discriminative Transfer Learning with Tree-based Priors

Motivation

In this paper there are two main contributions. Firstly, a method is proposed in order to combine the deep neural network with some tree-based priors. Moreover, the algorithm for learning the tree structure is also proposed.

Methods

- Model description:
  As shown in Figure 2.17(a), the system is modeled as a multi-layer neural network. $\boldsymbol{w}$ represents the set of all parameters of the network except the top-level weights, which are denoted by $\beta \in \mathbb{R}^{D\times K}$ where D is the number of hidden units in the last hidden layer and K is the number of labels.

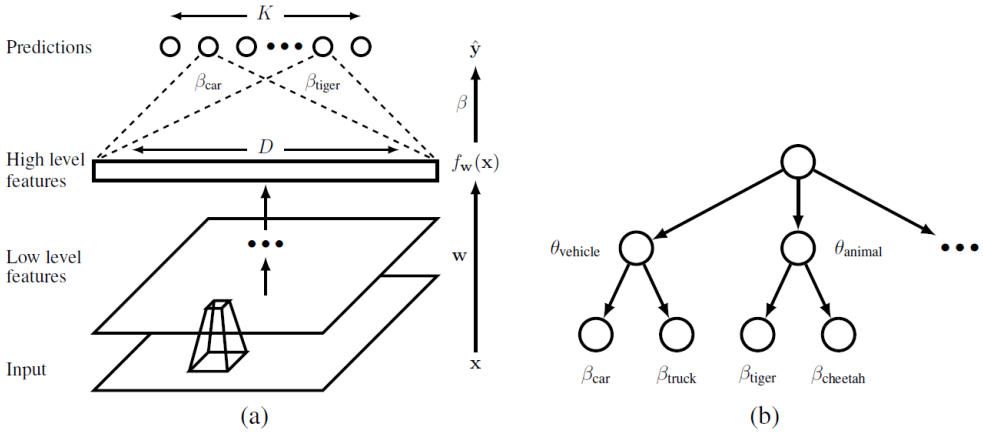


Figure 2.17: Graphic model of a deep neural network and a tree hierarchy [16].

- Learning with a fixed tree hierarchy:
  - Assumption:
    In this part it is assumed that there is already a available fixed tree hierarchy organized by the classes.
  - Example: a two-layer hierarchy as shown in Figure 2.17(b)
    Each leaf k on the lowest layer is associated with a weight vector $\beta_k \in \mathbb{R}^D$.
    Each siper-class node s is associated with a vector $\theta_s \in \mathbb{R}^D$.
    $\theta_s \in \mathbb{R}^D$ and $\beta_k \in \mathbb{R}^D$ are modeled as normal distributions:

$$\theta_s \sim \mathcal{N}(0, \frac{1}{\lambda_1} I_D), \quad \beta_k \sim \mathcal{N}(\theta_{parent(k)}, \frac{1}{\lambda_2} I_D)$$

  - Loss function:

$$L(\boldsymbol{w}, \beta, \theta) = -\log P(\mathcal{Y}|\mathcal{X}, \boldsymbol{w}, \beta) + \frac{\lambda^2}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda_2}{2}\sum_{k=1}^{K}\left\|\beta_k - \theta_{parent(k)}\right\|^2 + \frac{\lambda_1}{2}\|\theta\|^2$$

  - The loss function can be minimized with the following 2-step iteration:

1. Optimize $\boldsymbol{w}$ and β with θ fixed;
2. Optimize θ with β fixed.

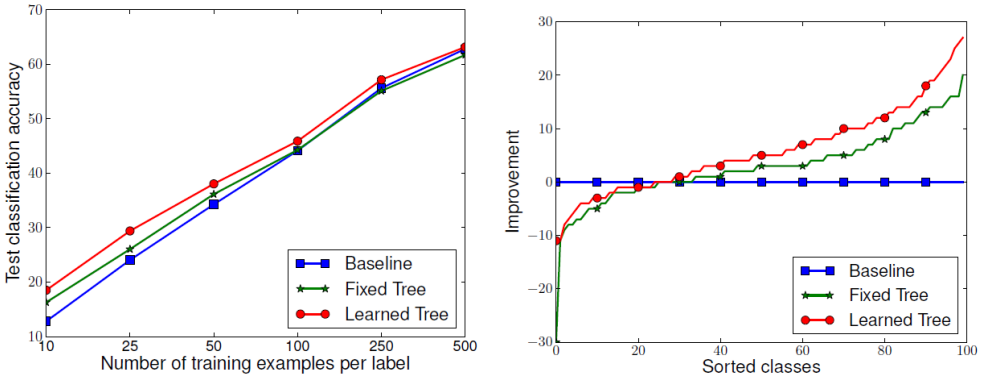
- Learning the tree hierarchy:
  - When learning the structure of the tree a Chinese Restaurant Process (CRP) is used to determine whether a new incoming class belongs to a existing superclass or to a new superclass.
  - Optimization problem:

$$\max_{\boldsymbol{w},\beta,\theta,\boldsymbol{z}} \log P(\mathcal{Y}|\mathcal{X},\boldsymbol{w},\beta) + \log P(\boldsymbol{w}) + \log P(\beta|\theta,\boldsymbol{z}) + \log P(\theta) + \log P(\boldsymbol{z})$$

    where $\boldsymbol{z}$ is a vector indicating the connections of each class and their superclass.

Experiments and Results

- Experiments on the CIFAR-100 dataset:
  - Characteristic of CIFAR-100:
    CIFAR-100 has a large number of classes but only a few samples in each class.
  - Experiments with few examples per class:
    As shown in Figure 2.18, the learned tree outperforms the original baseline and the fixed tree, and the classification for most of the classes are improved.



(a) (b)

Figure 2.18: Results of experiments with few examples per class [16].

  - Experiments with few examples for one class:
    For the situation where there are many examples for different classes but only few for one particular class. The experiments are designed to observe the behavior of taking advantage of learned classes to learn a new class with few examples. The results shown in Figure 2.19 shows that the classification accuracy of the learned tree is higher than others and the results are similar for other classes other than dolphin.

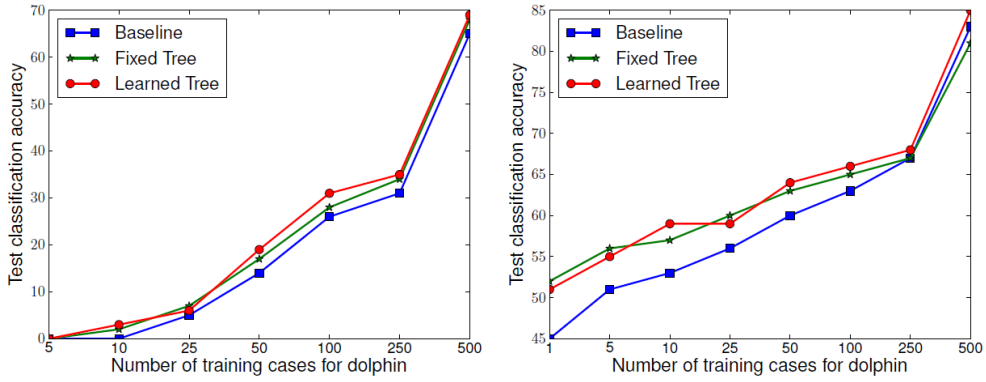


Figure 2.19: Results of experiments with few examples for one class [16].

- Experiments on the Multimedia Information Retrieval (MIR) Flickr dataset:
  The performance of the learned tree compared to baseline and the fixed tree is similar to the results on CIFAR-100.

Discussion

Modeling the classes into a tree structure makes the relations among each class clearer. The proposed algorithm constructs the tree hierarchy adaptively, which provides flexibility as well as improvement in performance.

2.9 Dyadic Transfer Learning for Cross-Domain Image Classification

Motivation

In this paper a Dyadic Knowledge Transfer (DKT) approach is proposed for image classification. DKT approach is based on the nonnegative matrix tri-factorization and is able to transfer cross-domain knowledge, both unsupervised and supervised information, from source data to target data. Moreover, an efficient approach for solving the objective of DKT approach is also proposed.

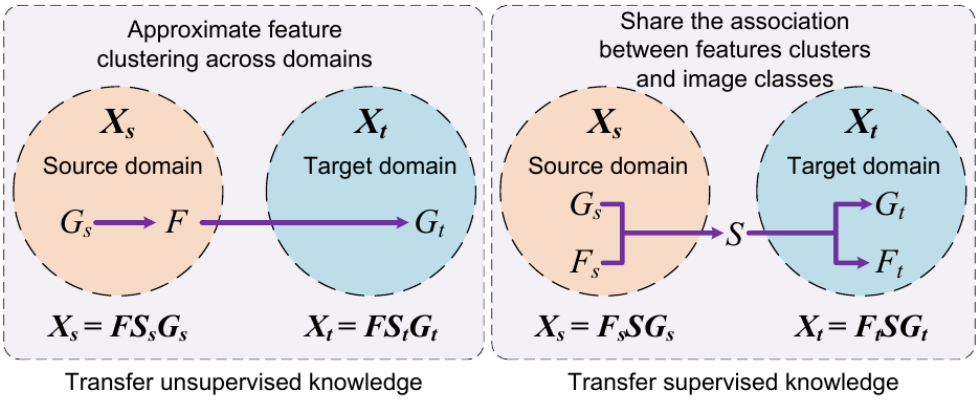


Figure 2.20: Graphic illustration of Dyadic Knowledge Transfer approach for transferring unsupervised and supervised knowledge[19].

Methods

- Basis of proposed approach (NMTF)
  - The goal of nonnegative matrix tri-factorization is to estimate a nonnegative matirx with 3 nonnegative factor matrices.
  - The general form of NMTF is

$$\min_{F\geq 0, S\geq 0, G\geq 0} \|X - FSG\|^2$$

    where X is the matrix to be approximated, F contains the unsupervised information (native structural information) of X, S contains the supervised information (annotations) of X, and each row of G is the soft clustering indication of a data point.

- Objective of the DKT approach
  - The objective function is written as follow:

$$\begin{aligned}\min_{F\geq 0, S\geq 0, G_s\geq 0, G_t\geq 0} J =& \left\|X_s - FSG_s^T\right\|^2 + \left\|X_t - FSG_t^T\right\|^2 \\ &+ \alpha \boldsymbol{tr}[Q_s(G_s - Y_s)^T C_s(G_s - Y_s) \\ &+ Q_t(G_t - Y_t)^T C_t(G_t - Y_t)]\end{aligned}$$

  - F and S in the objective represent the shared unsupervised and supervised knowledge between the source data and the target data, respectively.

  - The two terms of trance $\boldsymbol{tr}[Q_s(G_s - Y_s)^T C_s(G_s - Y_s)]$ and $\boldsymbol{tr}[Q_t(G_t - Y_t)^T C_t(G_t - Y_t)]$ pick out which images are annotated by labels Y, and the matrix Q is used to enforce the available label information in source or target domain.

- A novel optimization algorithm:

  - Idea:
    The basic idea of this new optimization algorithm is to update the Matrices F, S, G_s and G_t iteratively.

  - First, the objective function is expanded and the constant parts are then discarded. For every following iteration step, F, S, G_s and G_t are computed by introducing a Lagrangian multiplier, respectively.

Experiments and Results

- Datasets: the TRECVID 2005 dataset and the MSRC dataset

- In the designed experiments, the performance of proposed DKT approach is compared to its non-transfer version. The results are shown in Figure 2.21 and Figure 2.22. The classification precision of the DKT approach is higher for all categories, and the classification precision is improved even for the unshared semantic concepts.

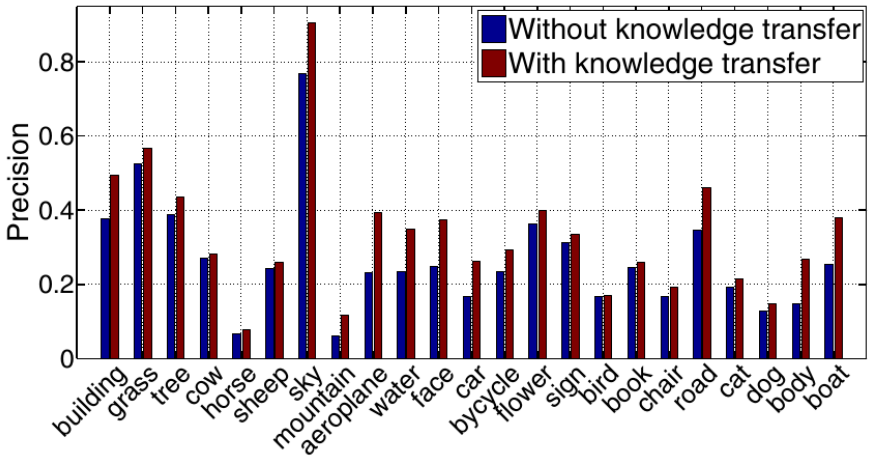


Figure 2.21: Precision of classification of DKT approach compared to its non-transfer version[19].

Discussion

It is creative for the proposed knowledge transfer algorithm to be able to transfer both the unsupervised and the supervised information from source domain to target domain. This performance is realized by utilizing the NMTF, where the matrices F and S used for approximating X represent the unlabeled and labeled information respectively.

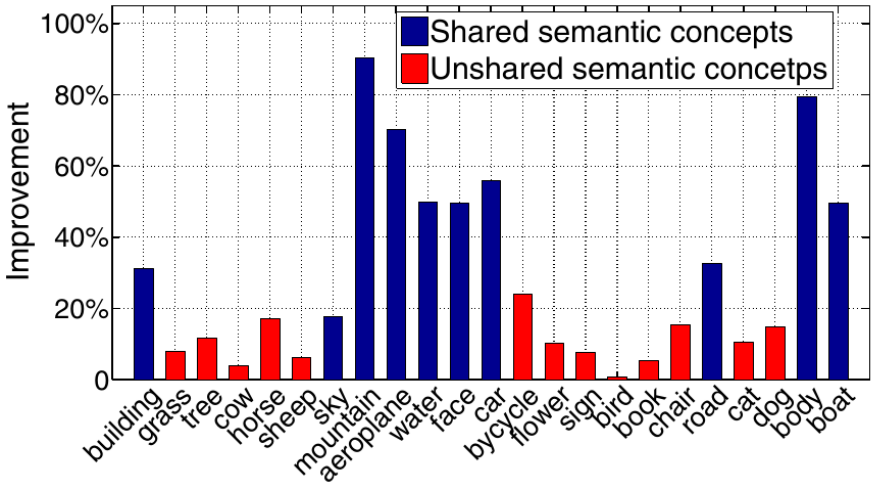


Figure 2.22: Corresponding improvement in classification with help of knowledge transfer[19].

2.10 Heterogeneous Transfer Learning for Image Classification

Motivation

In this paper a heterogeneous transfer learning based algorithm *HTLIC* is proposed for image classification. The creativity lies in the choice of source data, which include unlabeled semantic concepts. Those auxiliary texts can connect images with semantic level representations and are used to help improve the learning process.

Methods

- Auxiliary source data:
  - unlabeled annotated images: $\boldsymbol{I} = \{\boldsymbol{z}_i, \boldsymbol{t}_i\}_{i=1}^{l}$
    where $\boldsymbol{z}_i$ represents auxiliary images, $\boldsymbol{t}_i$ represents the corresponding tags.
  - unlabeled text documents: $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{k}$
    where $\boldsymbol{d}_i$ represents the documents which is a vector of bag-of-words.

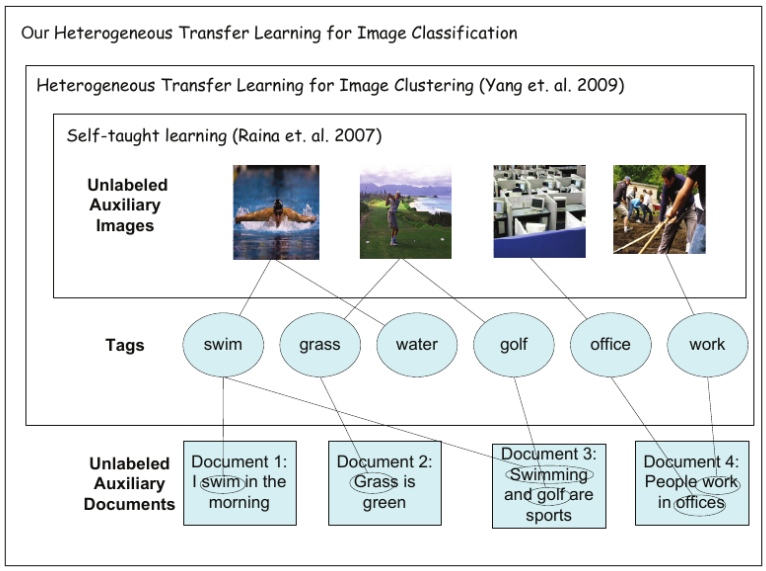


Figure 2.23: Graphic illustration of the two-layer bipartite graph[21].

- Bridging images and text:
  - In this stage a two-layer bipartite graph is introduced to represent the connections among images, their tags and the text documents, which is shown as Figure 2.23.
  - The top layer consists of the relationship between images and their corresponding tags, which is represented by the matrix $\boldsymbol{G} = \boldsymbol{Z}^T\boldsymbol{T}$.
  - The bottom layer illustrates the relationship between the tags and text documents, which is represented by the matrix F.
- Learning semantic features for images:

- Latent Semantic Analysis (LSA) is used for decomposition of matrices $\boldsymbol{F}$ and $\boldsymbol{G}$, in the form of $\boldsymbol{G} = \boldsymbol{U} \cdot \boldsymbol{V}_1^T$ and $\boldsymbol{F} = \boldsymbol{W} \cdot \boldsymbol{V}_2^T$.
  Matrices $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ comprise vectors of latent semantic representations of the tags in top and bottom layers.
  $\boldsymbol{U}$ is made up of latent semantic representations of the auxiliary images, while the matrix $\boldsymbol{W}$ provides the latent semantic representations of text documents.
- In order to describe the images in latent semantic representations more precisely, the matrix $\boldsymbol{U}$ should be precise enough. The decomposition of $\boldsymbol{F}$ is used to improve the same proess of $\boldsymbol{G}$. It is effective to choose $\boldsymbol{V}_1 = \boldsymbol{V}_2 = \boldsymbol{V}$.
- Objective:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}} = \lambda \|\boldsymbol{G} - \boldsymbol{U}\boldsymbol{V}\|_F^2 + (1-\lambda)\|\boldsymbol{F} - \boldsymbol{W}\boldsymbol{V}\|_F^2 + R(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$$

  where $R(U, V, W)$ is a regularization function.

- Constructing new representations: To unify the representation of images and text documents and apply the results of the learning process, it is necessary to map the target images $\boldsymbol{X}$ to semantic feature space by $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i \boldsymbol{U}$.

Experiments and Results

- Datasets:
  For source domain, annotated images and text documents are selected from Flickr and Google respectively.
  For target domain, subsets of Caltech-256 are used.

- Evaluation criterion:

$$ACC(f, \boldsymbol{X}^\star, \boldsymbol{Y}^\star) = \frac{\sum_{\boldsymbol{x}_i^\star \in \boldsymbol{X}^\star} \boldsymbol{I}[f(\boldsymbol{x}_i^\star) = y_i^\star]}{|\boldsymbol{X}^\star|}$$

- Proposed algorithm *HTLIC* compared to other baselines:
  As shown in Figure 2.24, the 4 classification tasks with highest improvement and 3 tasks with lowest improvement among the total 171 tasks are listed in the table. Averagely it can be concluded that the proposed algorithm outperforms other baselines in image classification.

Tasks	**Orig**	**PCA**	**Tag**	**HTLIC**
watermelon vs *sheet-music*	64.66 ± 9.99	70.28 ± 11.33	78.13 ± 14.40	85.29 ± 11.94
fried-egg vs *american-flag*	59.19 ± 7.80	60.54 ± 9.28	63.70 ± 12.54	78.80 ± 12.21
fried-egg vs *school-bus*	65.42 ± 10.72	66.73 ± 11.01	75.58 ± 14.56	83.74 ± 11.88
zebra vs *motorbikes*	69.95 ± 11.74	70.55 ± 12.37	85.74 ± 13.72	86.66 ± 12.32
minaret vs *lighthouse*	53.67 ± 7.62	53.61 ± 6.18	52.71 ± 7.03	53.32 ± 6.38
llama vs *greyhound*	51.48 ± 7.11	52.65 ± 5.58	50.79 ± 5.53	51.94 ± 5.40
cd vs *cake*	62.85 ± 10.45	65.20 ± 11.87	54.98 ± 5.33	57.71 ± 8.35
Average	63.1925	67.0312	66.3192	71.5493

Figure 2.24: Comparison with baselines[21].

Discussion

It is effective and also creative to take advantage of not only the images but also the text documents for image classification. With help of the auxiliary text as source data, the connections of the images in target domain can be extended to semantic level, which is able to discover deeper relationships of the images.
Moreover, the auxiliary text documents are also easy to get from the Internet.

2.11 Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective

Motivation

Previous learning to learn frameworks are not able to be compatible with different problem settings such as domain adaptation and transfer learning, due to the different assumptions of each algorithm. In this paper a new learning to learn framework is proposed to learn from the source data without considering the distribution mismatch between the source and target domain. The confidence values calculated from the source data, rather than the source data, are used as extra features and are then combined with the features from the target domain.

Methods

- Typical learning to learn frameworks:
  The goal of learning to learn frameworks is to apply the few labeled samples together with many source sets to improve the learning process. Previous learning to learn frameworks are specified to different situations.
  - Domain adaptation is designed for the case where the source domain and target domain have the same labels ($Z_S = Z_T$) but the data distributions are different ($P_S(X) \neq P_T(X)$).
  - In contrast, transfer learning assumes that the labels used in source and target domain are not identical ($Z_S \neq Z_T$), but the data distributions in these two fields are related to each other ($P_S(X) \sim P_T(X)$).
- The high-level learning to learn framework (*H-L2L*):
  - The above mentioned confidence values of a sample $\boldsymbol{x}$ with respect to a category z is conveyed by a score function:

$$s(\boldsymbol{x}, z) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}, z)$$

  - The high-level integration assigns different weights β to the confidence values and then combine them together:

$$s(\boldsymbol{x}, z) = \sum_{j=1}^{F} \beta_z^j s^j(\boldsymbol{x}) = \sum_{j=1}^{F} \beta_z^j \boldsymbol{w}_z^j \phi^j(\boldsymbol{x})$$

    where F denotes the number of features describing one sample.
- The *H-L2L* framework:
  - The *H-L2L* framework is extended from the high-level integration scheme.
  - Score function:

$$s(\boldsymbol{x}, z) = \beta^{(0)} \boldsymbol{w}^{(0)}(\boldsymbol{x}, z_T) + \sum_{z=1}^{F_S} \beta^{(z_T, z)} \boldsymbol{w}^{(z_T, z)} \phi^{(z_T, z)}(s_S(\boldsymbol{x}, z), z_T)$$

  - The first term of the score function corresponds to the original training samples in target domain. The second term represents the confidence scores predicted by the source data, which are transferred into the learning in target data. Thw graphic illustration of this framework is shown in Figure 2.25.

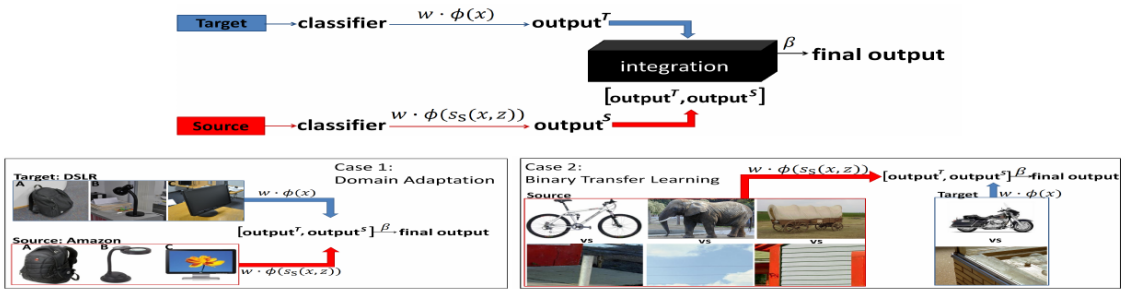


Figure 2.25: Graphic illustration for high-level learning to learn framework[13].

Experiments and Results

- The experiments are designed to verify the effectiveness of the proposed framework on domain adaptation, binary and multi-class transfer learning.
- Two instantiations of *H-L2L* used in the experiments:
  - *H-L2L(SVM-DAS)*: The feature representation is constructed by simply augmenting the confidence values of source and target domain.
  - *H-L2L(LP-β)*: The parameters β are learned through a boosting approach.

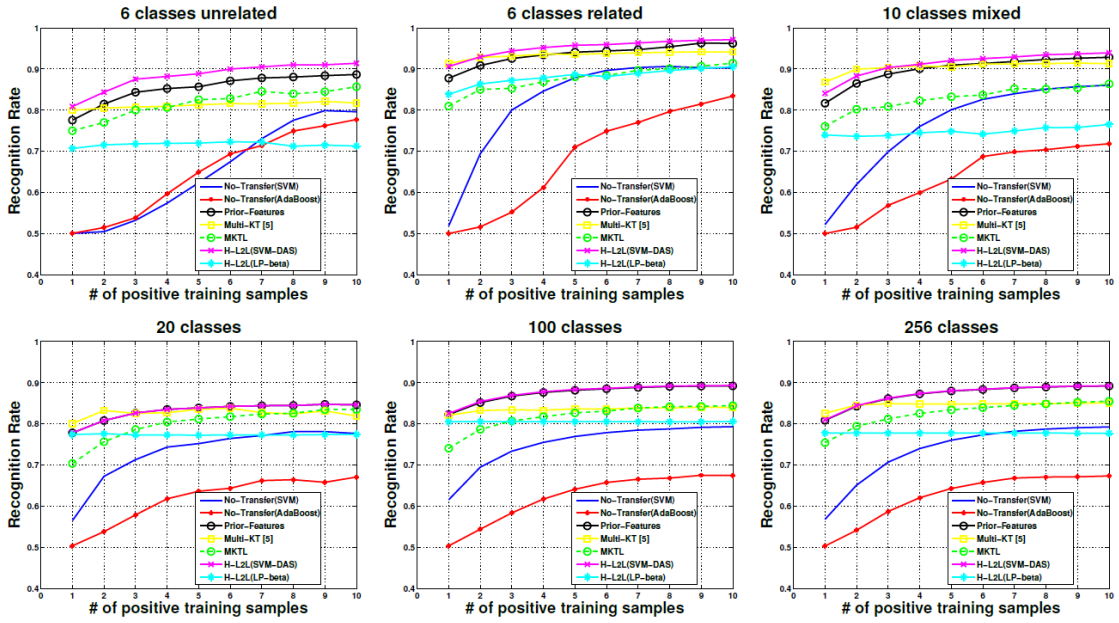


Figure 2.26: Experiments on binary transfer learning with small number of classes and increasing number of classes [13].

- Results:
  For the three different situations, the framework *H-L2L(SVM-DAS)* always presents top-level performance.
  However, the performance of *H-L2L(LP-β)* is unstable. It outperforms all other algorithms

in domain adaptation, but in transfer learning, its performance is not that outstanding (As shown in Figure 2.26), showing a sign of overfitting.

Discussion

The proposed framework shows its effectiveness in dealing with different kinds of problems by using the predicted confidence values as experts and combining them with the target samples, regardless of the causes of the distribution mismatch. Unfortunately, not all of the instantiations can adapt to all these cases.

2.12 Tabula Rasa: Model Transfer for Object Category Detection

Motivation

In this paper three different but related SVM based transfer learning algorithms are proposed. The transfer learning is conducted on the HOG (histogram of oriented gradient) template models, and the models from source domain are transferred to help improve the learning of a new category.

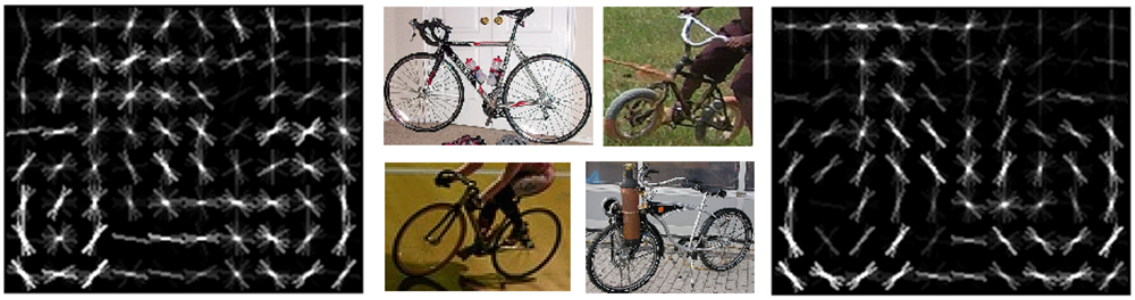


Figure 2.27: Transferring the already learned model for "motorbike" (left) to learn a new category "bicycle" (right)[1].

Methods

- An example of the HOG template models are shown in Figure 2.27, which has 8×10 cells. Each cell of the HOG model is represented by a vector of 32 dimensions.

- Adaptive SVM (*A-SVM*):
  This algorithm is a normal and classic one that minimizes the distance between the target model and the source model. Its objective function is shown as follows:

$$L_A = \min_{\boldsymbol{w},b} \left\| \boldsymbol{w} - \Gamma \boldsymbol{w}^S \right\|^2 + C \sum_i^N l(\boldsymbol{x}_i, y_i; \boldsymbol{w}, b)$$

  The parameters Γ and C control the weights of the source model $\boldsymbol{w}^S$ and the loss function $l(\boldsymbol{x}_i, y_i; \boldsymbol{w}, b)$ respectively.

- Projective Model Transfer SVM (*PMT-SVM*):
  The basic idea of this method is to minimize the projection of target model $\boldsymbol{w}$ onto the separating hyperplane orthogonal to source model $\boldsymbol{w}^S$. The objective function is shown as follows:

$$L_{PMT} = \min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + \Gamma \|Pw\|^2 + C \sum_i^N l(\boldsymbol{x}_i, y_i; \boldsymbol{w}, b)$$

  In this objective function, the term $\|Pw\|^2 = \|\boldsymbol{w}\|^2 \sin^2\theta$ is the projection of $\boldsymbol{w}$ onto the orthogonal hyperplane of $\boldsymbol{w}^S$.

- Deformable Adaptive SVM (*DA-SVM*):

  – The ability of the source template to be able to change itself slightly can help improve the fitting between the source and target models.

– This deformation can be represented by the following formulation:

$$\tau(\boldsymbol{w}^S) = \sum_{i}^{M} f_{ij} \boldsymbol{w}_j^S$$

where $\boldsymbol{w}_j^S$ denotes the jth cell of the source template, f_{ij} is the corresponding weight for the transfer of that cell.

– Therefore, the objective function of *DA-SVM* can be written as follows:

$$L_{DA} = \min_{f,\boldsymbol{w},b} \left\| \boldsymbol{w} - \Gamma\tau(\boldsymbol{w}^S) \right\|^2 + C \sum_{i}^{N} l(\boldsymbol{x}_i, y_i; \boldsymbol{w}, b) + \lambda \left(\sum_{i \neq j}^{M,M} {f_{ij}}^2 d_{ij} + \sum_{i}^{M} (1 - f_{ij})^2 d \right)$$

The last term $\lambda \left(\sum_{i \neq j}^{M,M} {f_{ij}}^2 d_{ij} + \sum_{i}^{M} (1 - f_{ij})^2 d \right)$ conveys the amount of the deformation. If the value of λ is chosen to be small, the deformation will become more flexible and obvious.

Experiments and Results

- Dataset: PASCAL VOC 2007 dataset
- The algorithms are evaluated in two main aspects, namely one shot learning and multiple shot learning. Moreover, two groups of related categories are taken into consideration (horse-cow and motorbike-bicycle).

Ranks	Base. SVM	A-SVM	DA-SVM	PMT-SVM
01-15	40.5 ± 07.2	**53.9 ± 04.2**	53.7 ± 04.3	53.5 ± 05.7
16-30	33.0 ± 13.5	52.5 ± 08.3	51.9 ± 08.8	**54.7 ± 05.7**
31-45	26.4 ± 13.3	47.1 ± 07.3	47.1 ± 07.6	**48.5 ± 08.7**
46-60	14.0 ± 09.3	42.4 ± 03.7	**42.5 ± 04.2**	27.8 ± 11.3

Source: motorbike(**44.7%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

Figure 2.28: Performance of one shot learning on learning bicycle when given the motorbike classifier[1].

- Results:
  The result for one shot learning (motorbike-bicycle) is shown in Figure 2.28. The ranks of source samples are obtained by source classifier, in which lower rank means worse resolution or undesired angle of view. The proposed algorithms perform much better than baseline SVM.
  As shown in Figure 2.29, the results conducted for multiple shot learning again confirm the better performance of proposed algorithms. In the last figure, *PMT-SVM* outperforms others when there is negative transfer.

Discussion

- Pros:
  – The representation of source models with HOG is creative and effective.

  - The construction of *PMT-SVM* and *DA-SVM* provides some new views on establishing the objective function, and they work well in the experiments.

- Cons:
  - Although the three proposed algorithms outperform the baseline SVM, no one of them shows top-level performance in all the experiments.

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	05.2 ± 05.6	23.6 ± 02.9	**24.1 ± 03.2**	22.7 ± 12.1
2	09.0 ± 07.7	32.3 ± 03.9	32.4 ± 04.9	**34.3 ± 07.3**
3	18.9 ± 10.9	34.7 ± 05.5	35.0 ± 04.7	**36.0 ± 10.5**
4	24.7 ± 12.2	37.1 ± 04.5	**37.7 ± 04.0**	35.0 ± 05.6
5	28.7 ± 09.5	37.7 ± 06.6	**37.9 ± 06.4**	34.8 ± 06.9

Source: cow(**26.1%**), **Target:** horse(**60.2%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(a)

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	26.9 ± 11.2	51.3 ± 04.5	49.9 ± 05.1	**54.9 ± 04.0**
2	48.4 ± 05.0	**55.5 ± 05.8**	55.2 ± 05.1	55.4 ± 06.0
3	46.9 ± 11.0	54.2 ± 07.1	54.1 ± 06.7	**56.4 ± 06.9**
4	48.2 ± 09.5	**56.0 ± 08.5**	55.4 ± 07.3	54.2 ± 06.0
5	52.5 ± 09.1	58.1 ± 06.5	**58.7 ± 05.6**	56.8 ± 06.4

Source: motorbike(**44.7%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(b)

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	26.9 ± 11.2	06.0 ± 09.4	06.2 ± 09.3	**27.8 ± 08.1**
2	48.4 ± 05.0	26.4 ± 05.0	27.8 ± 05.4	**50.0 ± 06.0**
3	46.9 ± 11.0	33.3 ± 12.7	33.5 ± 12.5	**51.4 ± 12.9**
4	48.2 ± 09.5	36.3 ± 14.7	36.0 ± 14.3	**51.1 ± 12.7**
5	52.5 ± 09.1	45.4 ± 13.0	45.5 ± 13.4	**56.0 ± 09.3**

Source: horse(**00.9%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(c)

Figure 2.29: Transferring the already learned model for "motorbike" (left) to learn a new category "bicycle" (right)[1].

2.13 Transfer Feature Learning with Joint Distribution Adaptation

Motivation

In this paper a new transfer learning approach named Joint Distribution Adaptation (*JDA*) is proposed to solve the cross-domain adaptation problems. *JDA* is able to reduce the differences in marginal distributions as well as conditional distributions between the source and target domain. It provides new feature representations for the samples and ensures its robustness in different situations.

Methods

- What to transfer: Feature representations
- How to transfer : Joint Distribution Adaptation (*JDA*)
  - The proposed algorithm *JDA* can be divided into two parts: property preservation via Principal Component Analysis (*PCA*) and distribution adaptation based on Maximum Mean Discrepancy (*MMD*).
  - Main process of *JDA*:

  step 1 Apply *PCA* to reduce the dimension of original data to a smaller value k, which is represented by a matrix A ($Z = AX$);

  step 2 Use *MMD* for the first time to initialize matrix M in order to reduce the differences in marginal distributions between source and target domains;

  step 3 Assign pseudo labels to the unlabeled data in target domain with help of some base classifiers (for example, *SVM*), use *MMD* again to reduce the differences lying in conditional distributions. A new representation of the target data (A) is then obtained;

  step 4 Repeat step 3 and update the parameters until convergence.
  The classifier for target data is trained on the resulting labeled target data.

Experiments and Results

- Datasets: Totally 6 datasets are grouped into 4 types, namely USPS + MNIST, COIL20, PIE, and Office + Caltech-256.

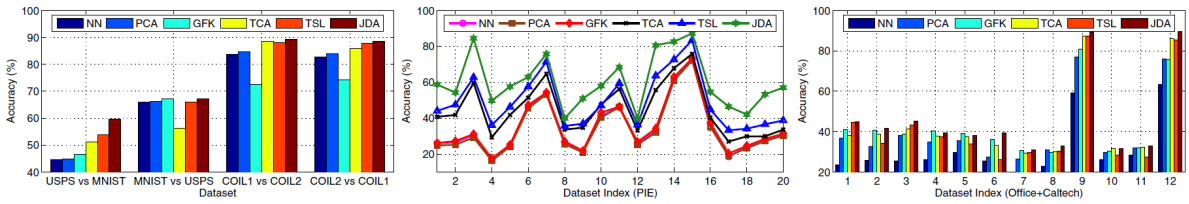


Figure 2.30: Classification accuracy on 36 cross-domain image datasets of *JDA* compared to other baselines. The values of x-axis of the figures in the middle and on the right stand for the indexes of comparisons within the datasets. That is, there are 20 and 12 comparisons between different subsets of PIE and Office + Caltech-256, respectively.[10].

- Results:
  As shown in Figure 2.30, the classification accuracy of proposed *JDA* stays in the highest

level when compared to other competitors. In further experiments, it is also shown that the classification accuracy can converge within 10 iterations.

Discussion

- Pros:
  - *JDA* is able to find out the conditional distributions by itself with help of dimensionality reduction and a base classifier.
  - It is effective to assign pseudo labels to the unlabeled target data and use these pseudo labels for further learning process.
- Cons:
  - The time complexity of this algorithm may become quite large if the number of features and examples are chosen to be large.

2.14 Transfer Feature Representation via Multiple Kernel Learning

Motivation

In this paper a transfer learning algorithm Transfer Feature Representation (*TFR*) is proposed. *TFR* transfers the feature representations from source domain, and it is able to learn the weights of a convex combination of classical kernels and a linear transformation at te same time. In this algorithm a differentiable cost function is introduced so that it can be easily solved with help of reduced gradient.

Methods

- Basis of *TFR*:
  - Maximum Mean Discrepancy (*MMD*):
    MMD is used to measure the distance between to distributions by equivalently measuring the distance between their data means in Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The normal form of *MMD* is:

$$\left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(\boldsymbol{x}_i) - \frac{1}{n_2} \sum_{1}^{n_2} \phi(\boldsymbol{z}_i) \right\|_{\mathcal{H}}$$

  - Multiple Kernel Learning (*MKL*):
    In *MKL* it is assumed that a learned kernel function is a convex combination of multiple classical or basis kernels.

$$\boldsymbol{K}(x_i, x_j) = \sum_{m=1}^{M} d_m \boldsymbol{K}_m(x_i, x_j), \quad d_m \geq 0, \ \sum_{m=1}^{M} d_m = 1$$

    where $\boldsymbol{K}_m$ represents the classical kernels.

- Three main conditions of *TFR*:
  - Minimizing the distribution difference between source and target domain:
    It is realized by combining *MMD* and *MKL* together, in order to get better performance.
  - Preserving the geometry of the data in target domain;
  - Preserving the valuable information of source data:
    The last two conditions can be fulfilled by introducing two diffusion kernels $\boldsymbol{K}_T$ and $\boldsymbol{K}_S$ respectively.

- The traditional cost function is improved to be differentiable, which can be simply minimized by reduced gradient.

- The learning algorithm of *TFR* is an iterative process which iteratively update the weights of each classical kernel as well as the linear transformation. The iteration ends until it reaches convergence.

Experiments and Results

- Datasets:
  FERET and YALE for face classification;
  Reuters-21578 and 20-Newsgroups for text classification.

- Results of face classification:

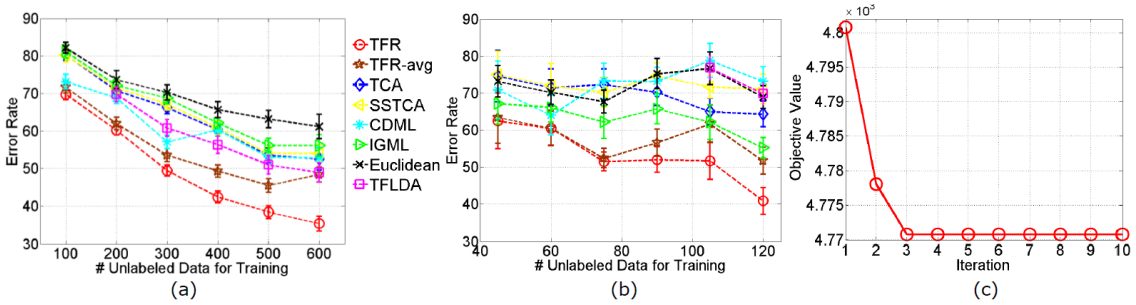


Figure 2.31: (a) Classification error rates when using YALE as source dataset and FERET as target domain set (denoted as YALE vs FERET); (b) Classification error rates when using FERET as source domain set and YALE as target domain set (denoted as FERET vs YALE); (c) Convergence evaluation of *TFR*.[20].

- Results for text classification:

Data Set	*orgs vs people*					*orgs vs place*					*people vs place*				
Size (%)	50	60	70	80	90	50	60	70	80	90	50	60	70	80	90
Eucliden	50.50	51.16	51.10	52.24	53.33	48.91	49.32	50.65	50.50	49.27	50.56	51.04	51.68	53.02	57.41
TCA	48.51	**43.58**	46.48	47.93	50.07	51.44	49.16	50.80	53.59	51.92	47.40	50.12	50.80	53.59	54.41
SSTCA	48.84	45.96	49.54	**42.15**	49.24	**42.99**	**42.93**	46.05	47.45	44.23	**43.63**	45.71	**45.05**	45.45	45.37
CDML	46.34	46.07	46.49	47.24	49.33	47.22	47.32	46.67	45.50	42.69	47.04	45.48	46.67	47.50	43.52
JDA	47.52	46.17	48.90	48.24	**46.28**	48.56	47.80	52.40	50.24	54.81	48.80	48.33	52.40	50.24	47.22
IGML	47.68	47.62	46.41	47.63	50.71	47.50	49.12	48.37	48.67	45.31	48.03	48.72	49.65	50.60	51.85
TFR	**43.71**	44.93	**45.07**	47.93	50.89	44.72	45.24	**44.60**	**44.37**	**42.31**	49.19	**45.03**	45.60	**44.37**	**43.52**
TFR-avg	47.85	45.13	46.38	48.17	52.37	46.83	49.40	45.73	47.85	50.00	50.44	45.71	48.61	52.92	44.64

Data Set	*comp vs rec*			*comp vs sci*			*comp vs talk*			*rec vs sci*			*rec vs talk*			*sci vs talk*		
Size (%)	30	40	50	30	40	50	30	40	50	30	40	50	30	40	50	30	40	50
Eucliden	52.40	53.71	56.20	61.41	59.62	62.97	55.24	58.47	54.23	57.06	55.33	56.29	45.10	43.33	46.73	52.75	53.08	55.79
TCA	50.64	49.66	**49.54**	50.35	49.72	50.70	**43.76**	44.90	45.23	58.86	**49.18**	49.87	43.74	44.02	43.87	50.56	50.50	50.61
SSTCA	51.92	51.11	51.54	50.64	50.75	50.41	44.83	45.21	43.35	49.26	51.41	50.83	43.21	44.79	43.35	48.41	**47.75**	48.95
CDML	51.27	51.90	51.54	49.14	**48.27**	51.59	46.52	49.06	43.17	50.52	50.87	**49.54**	45.72	51.44	46.25	53.52	49.59	52.17
JDA	52.61	51.28	52.87	48.88	48.70	**47.85**	46.14	47.55	45.39	53.29	52.49	49.67	47.85	48.43	48.69	50.92	50.37	51.02
IGML	55.21	52.04	53.85	53.35	50.56	53.12	47.97	49.72	50.99	56.08	52.59	56.09	**43.12**	**42.02**	**43.18**	53.39	50.63	53.85
TFR	**49.36**	**48.98**	50.00	**48.37**	49.03	48.21	46.32	**44.41**	**42.45**	**48.79**	49.39	50.09	45.32	46.87	43.45	**48.32**	47.95	**48.51**
TFR-avg	51.37	50.09	50.26	48.92	50.32	49.38	47.09	48.36	43.10	50.09	49.67	51.23	48.74	51.38	48.29	50.09	49.87	50.29

Figure 2.32: The chart on the top provides the classification errors of the algorithms on Reuters-21578 with increasing number of target data; The chart on the bottom shows the classification errors of the algorithms on 20-Newsgroups with increasing number of target data.[20].

Discussion

- Pros:
  - *TFR* converges faster than previous iterative learning algorithms, such as *JDA*[10].
  - The modification of cost function into differentiable form effectively improve the reliability of the learning process.
- Cons:
  - The performance of *TFR* in text classification can be further improved.

2.15 Transfer Learning to Account for Idiosyncrasy in Face and Body Expressions

Motivation

This paper provides an application of transfer learning in real-world face and motion recognition, which can be further applied in clinical environments. The introduction of transfer learning techniques can help overcome the individual idiosyncrasy of different patients and only make use of the commonalities among them.

Methods

- The learning process can be separated into 2 stages, namely transfer stage and calibration stage.
  In transfer stage the desired information is selected from transfer subjects (source data) and form a supervised learning model. In the calibration stage, the obtained model is then modified with some labeled data in target domain.

- A normal formulation of a Multi Task Learning (*MTL*) is:

$$\min_{W,C} \sum_{t=1}^{T} \left\| X_t^T w_t - Y_t \right\|_2^2 + \gamma \Omega(W, C)$$

  where $\Omega(W, C)$ is the regularization part, W consists of T weight vectors as columns, C represents the common information among the learning tasks.
  This paper implements the following three variants of *MTL*.

- Regularized MTL (*RMTL*):
  RMTL concentrates on the commonalities among the tasks by simply change the regularization part as

$$\Omega(W, w_0) = \frac{1}{T} \sum_{t=1}^{T} \|w_t - w_0\|_2^2 + \lambda \|w_0\|_2^2$$

- Multi-Task Feature Learning (*MTFL*):
  In *MTFL* it is assumed that there exists a common feature representation of the data in lower dimension. The optimization problem can be described as

$$\min_{W,D} \sum_{t=1}^{T} \left\| X_t^T w_t - Y_t \right\|_2^2 + \gamma \sum_{t=1}^{T} w_t^T D^{-1} w_t$$

  The matrix D transforms the similarities extracted from the transfer stage.

- Composite Multi-Task Feature Learning (*CMTFL*):
  CMTFL can be regarded as the combination of *RMTL* and *MTFL*. Its optimization problem is shown as follows:

$$\min_{W,D} \sum_{t=1}^{T} \left\| X_t^T w_t - Y_t \right\|_2^2 + \gamma \sum_{t=1}^{T} (w_t - w_0)^T D^{-1} (w_t - w_0) + \|w_0\|_2^2$$

Experiments and Results

- Datasets:
  UNBC-McMaster Shoulder Pain Expression Archive (facial expressions);
  Multi-Modal Chronic Lower Back Pain Dataset (motion data of human body).

- Results for facial expressions:
  As shown in Figure 2.33 and Figure 2.34, *RMTL* always outperforms other competitors.

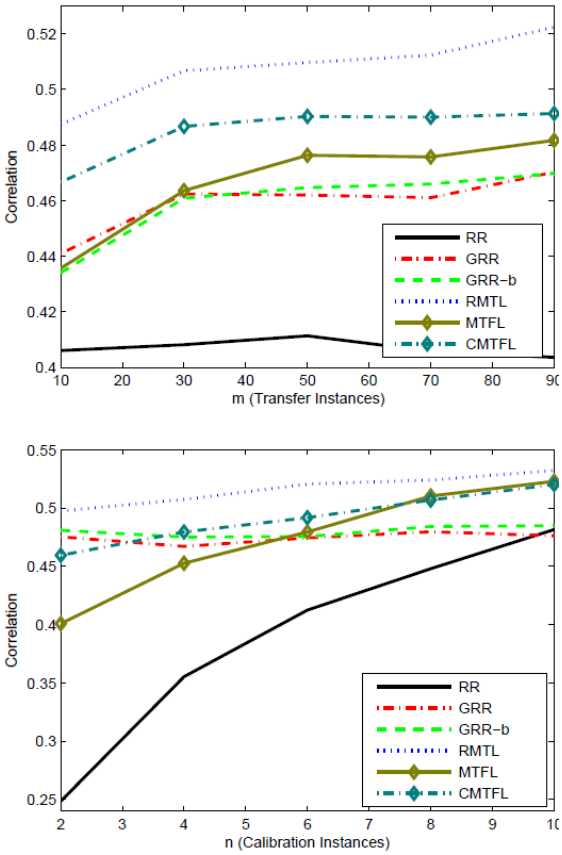


Figure 2.33: Experiments on facial expressions: The upper figure shows the results with increasing number of transfer subjects and 6 calibration samples; The lower one shows the results with fixed number of transfer subjects (90 samples) and an increasing number of calibration samples. [15].

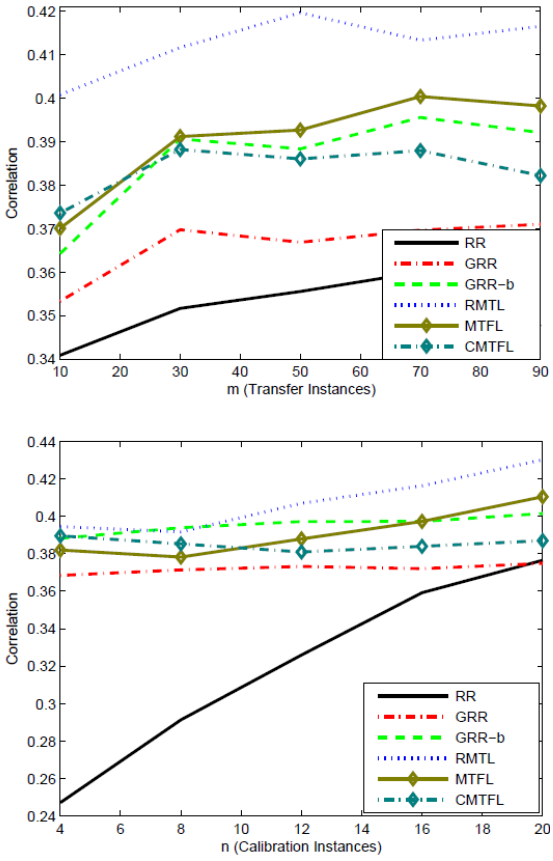


Figure 2.34: Experiments on body motion data: The upper figure shows the results with increasing number of transfer subjects and 6 calibration samples, while the other one represents the results with fixed number of transfer subjects (90 samples) and an increasing number of calibration samples. [15].

Discussion

This paper provides a possibility of applying transfer learning in practical situations. According to the experimental results, the transfer learning technique *RMTL* illustrates the best performance and it can be further utilized to eliminate the idiosyncrasy among the individuals.

2.16 Transfer Learning Based Visual Tracking with Gaussian Processes Regression

Motivation

In this paper the idea of transfer learning is applied into the visual tracking framework, where the models built on previous re-weighted auxiliary samples are transferred to the target decisions. The proposed visual tracking algorithm is *TGPR* (Gaussian Processes Regression with Transfer Learning), and the attention of this review is mainly laid on the transfer learning part.

Methods

- The objective of the visual tracking algorithm is to maximize the observation model

$$Pr(\boldsymbol{X}_t^i|\boldsymbol{\ell}_t^i) \propto Pr(y_i = +1|\boldsymbol{X}_t^i)$$

where $\boldsymbol{X}_t^i$ denotes the ith observation at time t, $\boldsymbol{\ell}_t^i$ the tracking candidates, and $y_i = +1$ means that the ith observation is the same as the tracking object.
As shown in Figure 2.35, the auxiliary samples are the samples acquired in the past while target samples are those captured in the recent frames.

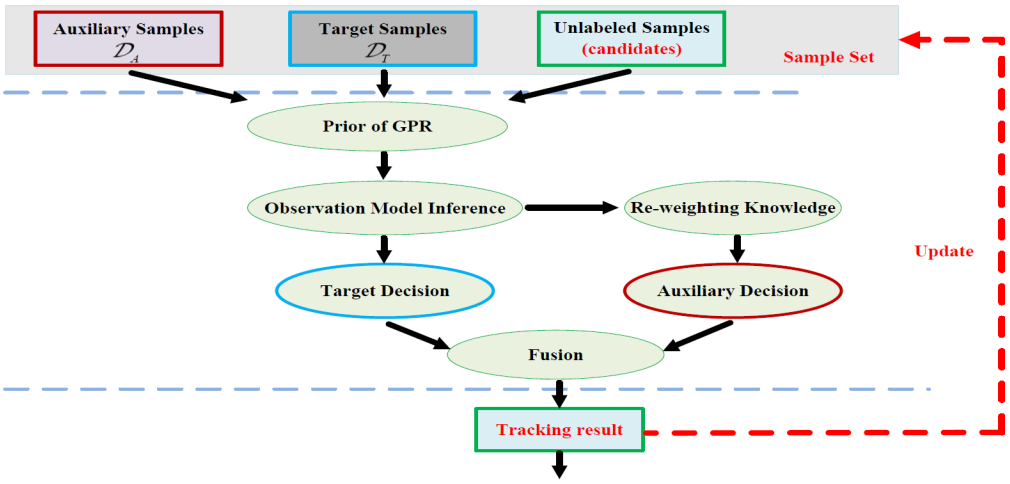


Figure 2.35: Graphic illustration of the proposed *TGPR* tracker.[3].

- What to transfer:
  The models built on the previous auxiliary samples.
- How to transfer:
  In the final tracking stage, two separate tracking decisions are constructed based on auxiliary data and target data respectively.
  After this a coincidence degree between the two candidate sets obtained by those two decisions is calculated. Based on this degree, the amount of auxiliary decisions being transferred to target decisions is determined.
- How much to transfer:
  If the coincidence degree is high, each one of the decisions can be applied for visual tracking. If this degree is quite small, then the target decision has higher reliability. However, if there is no coincidence between these decisions, auxiliary decisions should be more considered for recovery.

Experiments and Results

- The proposed tracking algorithm is evaluated on three benchmarks, namely CVPR2013 Visual Tracker Benchmark, Princeton Tracking Benchmark and VOT2013 Challenge Benchmark.
- Results:
  Take the results on Princeton Tracking Benchmark as an example. As shown in Figure 2.36, the tracking performance of *TGPR* is better than the performance of other algorithms for almost all the time. Its powerful tracking ability is also verified on the other two benchmarks.

Alg.	**Avg.**	target type			target size		movement		occlusion		motion type	
	Rank	human	animal	rigid	large	small	slow	fast	yes	no	passive	active
TGPR	**1.09**	**0.46(1)**	0.49(2)	**0.67(1)**	**0.56(1)**	**0.53(1)**	**0.66(1)**	**0.50(1)**	**0.44(1)**	**0.69(1)**	**0.67(1)**	**0.50(1)**
Struck	2.82	0.35(2)	0.47(3)	0.53(4)	0.45(2)	0.44(4)	0.58(2)	0.39(2)	0.30(4)	0.64(2)	0.54(4)	0.41(2)
VTD	3.18	0.31(5)	**0.49(1)**	0.54(3)	0.39(4)	0.46(2)	0.57(3)	0.37(3)	0.28(5)	0.63(3)	0.55(3)	0.38(3)
RGBdet	4.36	0.27(7)	0.41(5)	0.55(2)	0.32(7)	0.46(3)	0.51(5)	0.36(4)	0.35(2)	0.47(6)	0.56(2)	0.34(5)
CT	5.36	0.31(4)	0.47(4)	0.37(7)	0.39(3)	0.34(7)	0.49(6)	0.31(5)	0.23(8)	0.54(4)	0.42(7)	0.34(4)
TLD	5.64	0.29(6)	0.35(7)	0.44(5)	0.32(6)	0.38(5)	0.52(4)	0.30(7)	0.34(3)	0.39(7)	0.50(5)	0.31(7)
MIL	5.82	0.32(3)	0.37(6)	0.38(6)	0.37(5)	0.35(6)	0.46(7)	0.31(6)	0.26(6)	0.49(5)	0.40(8)	0.34(6)
SemiB	7.73	0.22(8)	0.33(8)	0.33(8)	0.24(8)	0.32(8)	0.38(8)	0.24(8)	0.25(7)	0.33(8)	0.42(6)	0.23(8)
OF	9.00	0.18(9)	0.11(9)	0.23(9)	0.20(9)	0.17(9)	0.18(9)	0.19(9)	0.16(9)	0.22(9)	0.23(9)	0.17(9)

Figure 2.36: Results of experiments on Princeton Tracking Benchmark, where the red numbers represent the best results and the blue ones are the second best.[3].

Discussion

In visual tracking tasks it is proved that considering and taking advantage of the previous data can obviously improve the ability of a visual tracker. The way of fusing two separate decisions together, which is utilized in this paper, can also effectively control the amount of transferred information in the decision making.

2.17 Transfer Learning for Pedestrian Detection

Motivation

Pedestrian detection aims to figure out the human in a scene of street view. In this paper a transfer learning based pedestrian detection method is proposed, which can be split in to two main parts, namely sample screening (*Isomap* algorithm) and classification (*ITLAdaBoost*).

Methods

- For pedestrian detection tasks, usually there are many labeled data in training scenes D_a, while the unseen scenes D_s consist of only limited number of labeled data and the test dataset D_t.
- The method is designed to expand D_s in unseen scenes with useful and related data points in D_a, so that the performance of the detection will be improved.
- Sample screening based on manifold learning:
  - Manifold learning is applied in order to conduct the feature dimensionality reduction as well as the data visualization.
  - As shown in Figure 2.37, there are always similar samples existing in training scenes and unseen scenes. Therefore, it is meaningful to make use of the data in D_a to enhance the detection ability.
  - This part of work include two main steps: First, the distances between data points are estimated; Then the related or similar points are picked out.

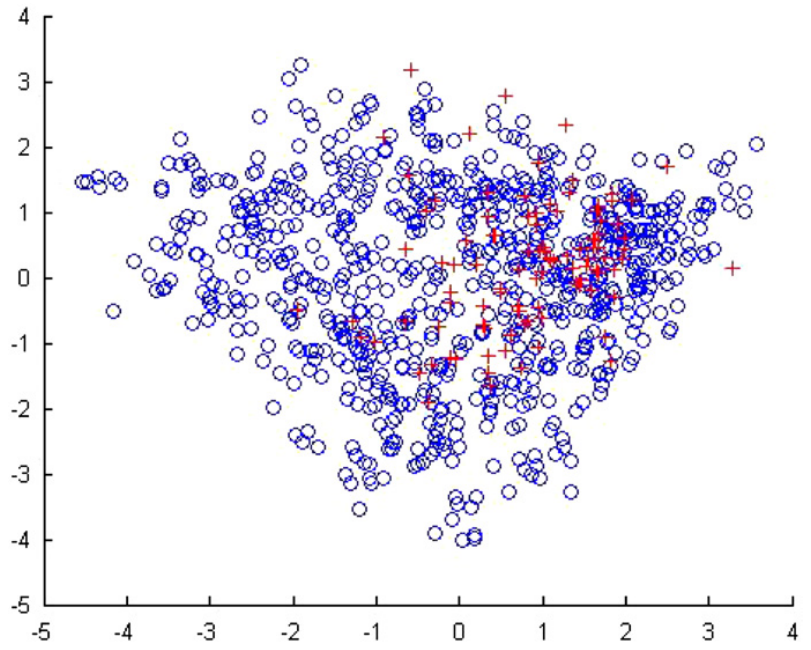


Figure 2.37: Data visualization of data points in unseen dataset including D_s and D_t (blue circles) and D_a (red crosses).[2].

- Classification based on transfer learning (*ITLAdaBoost*):

  - What to transfer:
    The samples in training scenes D_a are transferred to the unseen scenes.
  - How to transfer:
    The transfer learning is introduced by iteratively evaluate and update the weights of selected data points in D_a and D_s in their own domains separately. The final classifier is learned on the training dataset merged from D_a and D_s.

Experiments and Results

- Datasets: DC and NICTA
- Results:
  As shown in Figure 2.38, when compared to other three methods, the performance of proposed method is always the best. It is shown that the detection rate of proposed method becomes steady and reliable after 50 iterations.

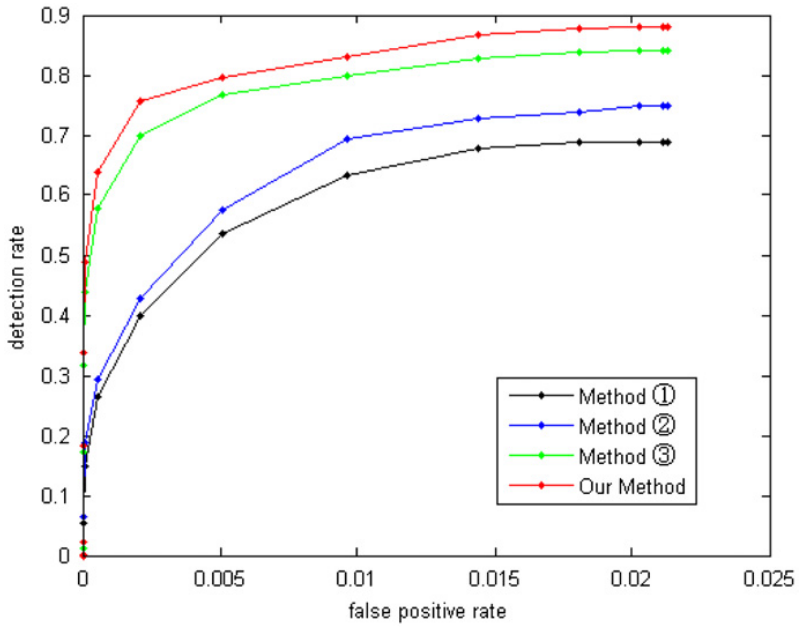


Figure 2.38: Results of proposed method compared to other three methods. Method 1: Classifier is trained with *AdaBoost* only on D_a; Method 2: Classifier is trained with *AdaBosst* on D_a and D_s; Method 3: Classifier is trained with *ITLAdaBoost* on D_a and D_s.[2].

Discussion

- Pros:
  - Using auxiliary data points from training scenes can improve the performance of pedestrian detection;
  - It is also important to select the similar or useful data in training scenes before they are transferred.
- Cons:
  - This method still requires some labeled data in the unseen scenes.

2.18 Transfer Learning in a Transductive Setting

Motivation

In this paper a new transfer learning algorithm Propagated Semantic Transfer (PST) is proposed. The proposed method applies a semi-supervised learning fashion which includes the knowledge transfer from the know categories as well as the visual similarities of the unlabeled samples. The feature representations of source data are modified as intermediate object-based or attribute-based representations.

Methods

- What to transfer:
  The feature representations of known object classes are transferred to the learning for classifiers of new categories.
- How to transfer:
  - For N new categories, the probability of a class z_n given an instance x can be represented in following two ways:
    1. With M intermediate attribute classifiers $p(a_m|x)$;
    2. With U most similar known categories y_u as a predictor for the new class $p(y_u|x)$.
  - $p(z_n|x)$ is utilized to build the label assignment of the instance x, while taking into account the possibly existing labels of some instances.
  - Evaluate the distances between two instances in the target domain with respect to their different representations:

    $$d(x_i, x_j) = \sum_{m=1}^{M} |p(a_m|x_i) - p(a_m|x_j)|$$

    or

    $$d(x_i, x_j) = \sum_{k=1}^{K} |p(a_k|x_i) - p(a_k|x_j)|$$

    Then construct a k-NN graph based on the distances calculated for the instances.
  - Iteratively update the labels assigned to each x.
- The graphic illustration of PST is shown in Figure 2.39.

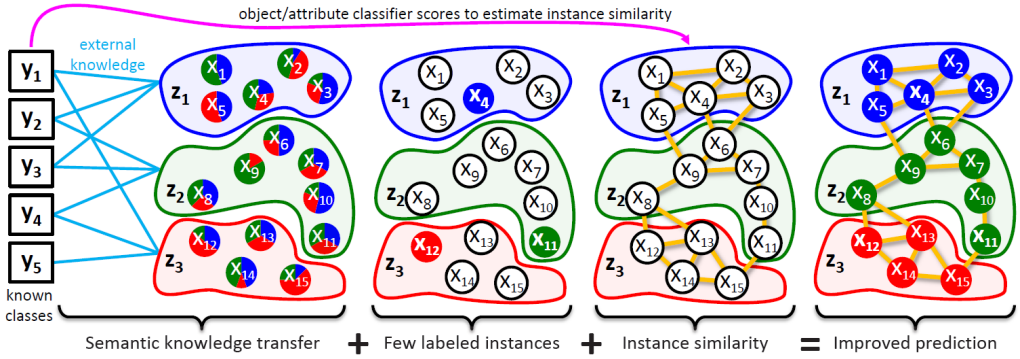


Figure 2.39: The classifier is learned with help of known categories, trained models, similarities within the data and some labels of the instances (if available).[14].

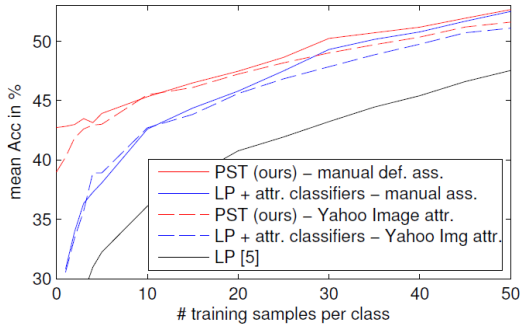
Experiments and Results

- Datasets:
  The Animals with Attributes dataset (AwA), ImageNet, and MPII Composite Cooking Activities together with some external knowledges related to these datasets

- Results:
  The results for experiments on AwA and ImageNet are shown in Figure 2.40 and Figure 2.41. The focus is taken on the ability of *PST* on zero-shot or few-shot learning.

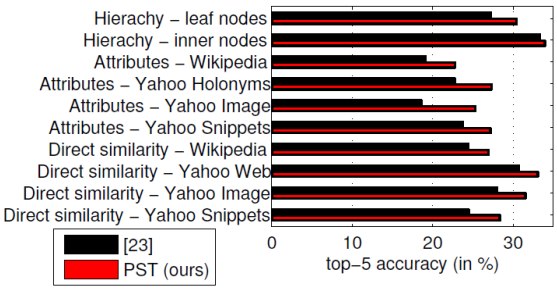
Approach	Performance AUC	Acc.
DAP [11]	81.4	41.4
IAP [11]	80.0	42.2
Zero-Shot Learning [9]	n/a	41.3
PST (ours)		
on image descriptors	81.2	40.5
on attributes	83.7	42.7

(a) Zero-Shot. Predictions with attributes and manual defined associations, in %.

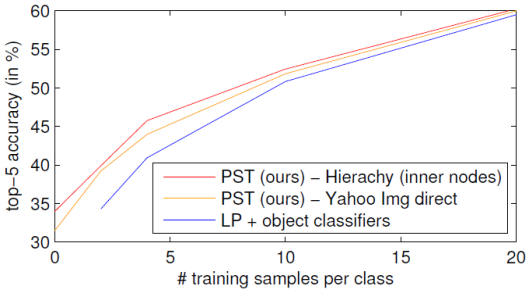


(b) Few-Shot

Figure 2.40: Results of *PST* on AwA dataset compared to label propagation (LP) baselines.[14].



(a) Zero-Shot.



(b) Few-Shot.

Figure 2.41: Results of *PST* on ImageNet compared to LP baselines. [14].

Discussion

The proposed transfer learning algorithm *PST* combines several different sources together and this combination does improve the performance of the classifier, which even shows impressive ability of zero-shot.

2.19 Transfer Learning to Predict Missing Ratings via Heterogeneous User Feedbacks

Motivation

The proposed algorithm Transfer by Collective Factorization (*TCF*) is designed to solve the problem of data sparsity in recommender systems. The idea is to take advantage of the binary data (like/dislike) and use them as auxiliary data to help predict the unobserved data in the target domain.

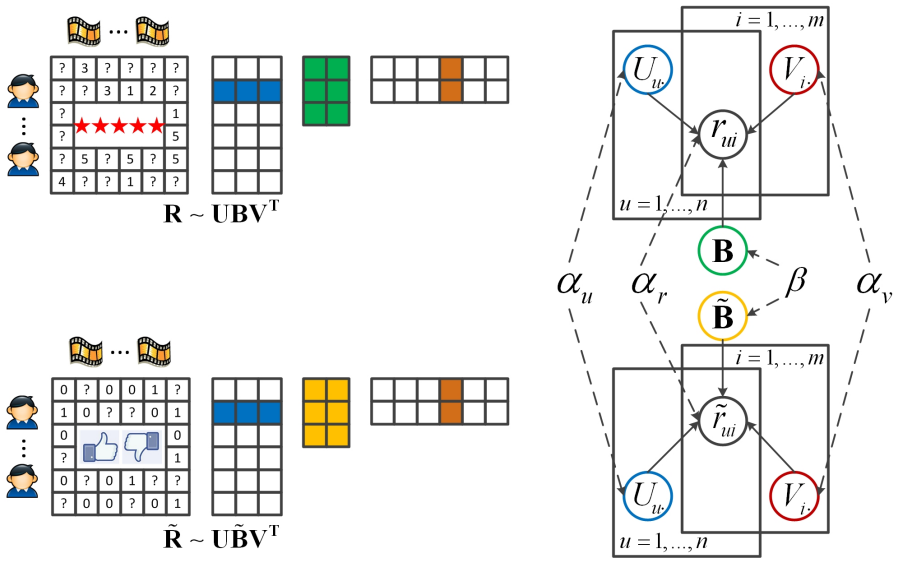


Figure 2.42: Graphic illustration of *TCF* [12].

Methods

- What to transfer:
  The latent features of auxiliary data are transferred to help construct the target data.
- How to transfer:
  - The optimization problem of *TCF* is

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{B},\tilde{\boldsymbol{B}}} \mathcal{F}(\boldsymbol{R} \sim \boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^T) + \lambda \mathcal{F}(\tilde{\boldsymbol{R}} \sim \boldsymbol{U}\tilde{\boldsymbol{B}}\boldsymbol{V}^T)$$

    where λ controls the amount of auxiliary data being transferred into target data; $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ are data observation matrices of target ratings and auxiliary binary ratings respectively; They are tri-factorized into $\boldsymbol{U}$, $\boldsymbol{B}$ ($\tilde{\boldsymbol{B}}$) and $\boldsymbol{V}$.
  - Initialize $\boldsymbol{U}$ and $\boldsymbol{V}$, then $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$ can be estimated separately but in the same manner with (taking $\boldsymbol{B}$ as example)

$$\min \frac{1}{2}\|\boldsymbol{r} - \boldsymbol{X}\boldsymbol{w}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{w}\|_F^2$$

    where $\boldsymbol{r}$ represents the corresponding observations of $\boldsymbol{B}$; $\boldsymbol{X}$ is the data matrix, and $\boldsymbol{w}$ a vector constructed by concatenation of the columns in $\boldsymbol{B}$.

  - Next, $\boldsymbol{U}$ and $\boldsymbol{V}$ can be calculated by solving the partial differential of the optimization problem.
  - Repeat the last two steps until the algorithm reaches convergence.

- The graphic model of *TCF* is shown as Figure 2.42.

Experiments and Results

- Datasets: Moviepilot Data, and Metflix Data
- Results:
  When computing $\boldsymbol{U}$ and $\boldsymbol{V}$, there are two different solutions, namely CMTF and CSVD. In the experiments, they are taken into consideration separately and compared with each other.

Data set	Sparsity of $\mathbf{R}$ (*Observed* tr. #, val. #)	Without transfer		With transfer		
		AF	PMF	CMF-link	CMTF (TCF)	CSVD (TCF)
Moviepilot	0.2% (tr. 3, val. 1)	0.7942±0.0047	0.8118±0.0014	0.9956±0.0149	*0.7415*±0.0018	**0.7087**±0.0035
	0.4% (tr. 7, val. 1)	0.7259±0.0022	0.7794±0.0009	0.7632±0.0005	*0.7021*±0.002	**0.6860**±0.0023
	0.6% (tr. 11, val. 1)	0.6956±0.0017	0.7602±0.0009	0.7121±0.0007	*0.6871*±0.0013	**0.6743**±0.0048
	0.8% (tr. 15, val. 1)	0.6798±0.0010	0.7513±0.0005	0.6905±0.0007	*0.6776*±0.0006	**0.6612**±0.0028
Netflix	0.2% (tr. 9, val. 1)	0.7765±0.0006	0.8879±0.0008	0.7994±0.0017	*0.7589*±0.0175	**0.7405**±0.0007
	0.4% (tr. 19, val. 1)	0.7429±0.0006	0.8467±0.0006	0.7508±0.0008	*0.7195*±0.0055	**0.7080**±0.0002
	0.6% (tr. 29, val. 1)	0.7308±0.0005	0.8087±0.0188	0.7365±0.0004	*0.7031*±0.0005	**0.6948**±0.0007
	0.8% (tr. 39, val. 1)	0.7246±0.0003	0.7642±0.0003	0.7295±0.0003	*0.6962*±0.0009	**0.6877**±0.0007

Figure 2.43: Results of experiments on *TCF* with other baselines [12].

Discussion

- Pros:
  - Taking advantage of auxiliary binary data is proved to be helpful for predicting the missing data in target domain.
  - More than one method are provided to solve the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, which brings in more flexibility.
- Cons:
  - The prediction performance decreases dramatically if the sparsity of auxiliary data increases.

2.20 Transfer Learning by Borrowing Examples for Multi-class Object Detection

Motivation

In this paper a transfer learning algorithm is proposed to borrow examples from other categories, so that the overall detection and classification performance is improved. In order to further expand the availability of the examples, they are transformed to get closer to the target category.

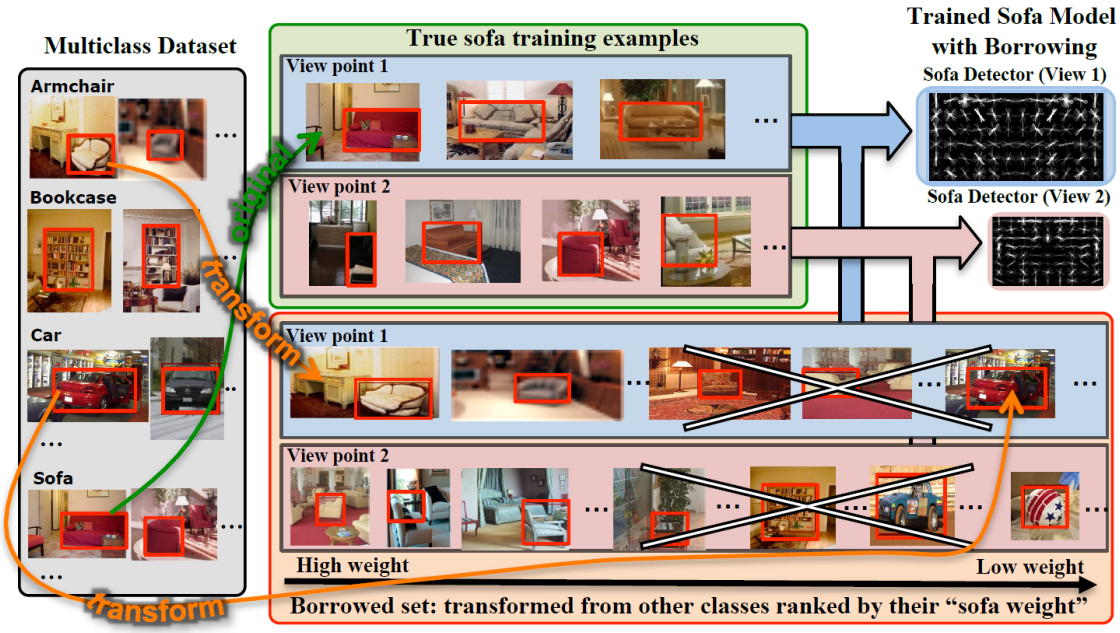


Figure 2.44: Example of transferring the samples in category "armchair" to learn the category "sofa" [9].

Methods

- What to transfer and from where to transfer:
  The samples are transferred to target from other learned categories.
- How to transfer:
  - Standard optimization problem for binary classification:

$$\min_{\boldsymbol{\beta}^c} \left(\sum_{i=1}^{n_c+b} Loss(\boldsymbol{\beta}^c \boldsymbol{x}_i, sign(y_i)) + \lambda R(\boldsymbol{\beta}^c) \right)$$

    where n_c denotes the number of labeled samples in class c, $\boldsymbol{\beta}^c$ consists of the regression coefficients for class c, $R(\cdot)$ is the regularization part and $Loss(\cdot)$ represents the loss function.
  - Proposed optimization model:

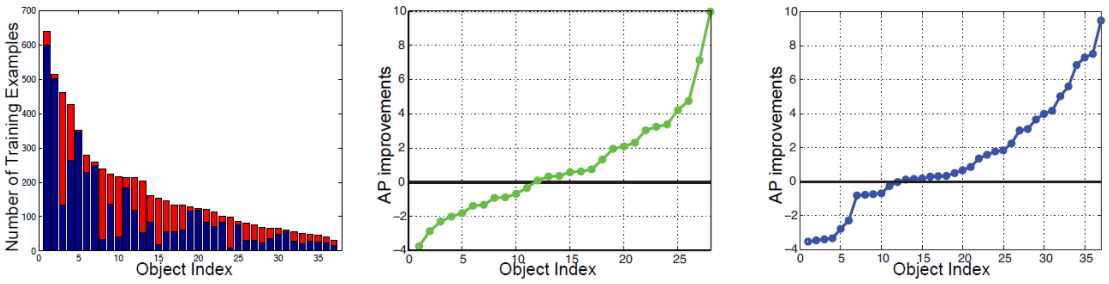
$$\sum_{c \in \mathcal{C}} \min_{\boldsymbol{\beta}^c} \min_{\boldsymbol{w}^{*,c}} \left(\sum_{i=1}^{n+b} (1 - \boldsymbol{w}^{*,c}) Loss(\boldsymbol{\beta}^c \boldsymbol{x}_i, sign(y_i) + \lambda R(\boldsymbol{\beta}^c) + \Omega(\boldsymbol{w}^{*,c}) \right)$$

where $\boldsymbol{w}_i^c$ indicates the amount of information class c borrows from the training sample $\boldsymbol{x}_i$.

- The parameters $\boldsymbol{w}$ and $\boldsymbol{\beta}$ can be solved iteratively. That is, solve for $\boldsymbol{\beta}$ given $\boldsymbol{w}$ with help of the standard optimization problem, and solve for $\boldsymbol{w}$ with fixed $\boldsymbol{\beta}$ by dealing with the new optimization model.
- Transformations are conducted on the transferred data such that they are able to be closer to the target data. Proposed transformations include translation, scaling and affine transformation.

Experiments and Results

- Datasets: the SUN09 dataset and the PASCAL VOC 2007 challenge
- Results:
  As shown in Figure 2.45, categories with fewer original samples tend to borrow more examples from other categories.



(a) Number of examples before/after borrowing

(b) Borrowed-set AP improvements

(c) Borrowed-transformed AP improvements

Figure 2.45: Results of experiments on borrowing examples from other categories [9].

Discussion

Borrowing examples from other known categories can improve the learning quality of a class. One of the most creative techniques in this paper is the transformations of transferred samples, which enables the training examples to be used in more situations.

List of Figures

Bibliography

[1] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2252–2259. IEEE, 2011.

[2] Xianbin Cao, Zhong Wang, Pingkun Yan, and Xuelong Li. Transfer learning for pedestrian detection. *Neurocomputing*, 100:51–57, 2013.

[3] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014.

[4] Efstratios Gavves, Thomas Mensink, Tatiana Tommasi, Cees GM Snoek, and Tinne Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2731–2739, 2015.

[5] Luo Jie, Tatiana Tommasi, and Barbara Caputo. Multiclass transfer learning from unconstrained priors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1863–1870. IEEE, 2011.

[6] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From n to n+ 1: Multiclass transfer incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3358–3365, 2013.

[7] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. Scalable greedy algorithms for transfer learning. *arXiv preprint arXiv:1408.1292*, 2014.

[8] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. Transfer learning through greedy subset selection. In *International Conference on Image Analysis and Processing*, pages 3–14. Springer, 2015.

[9] Joseph J Lim, Ruslan Salakhutdinov, and Antonio Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 118–126. Curran Associates Inc., 2011.

[10] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[11] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[12] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2318, 2011.

[13] Novi Patricia and Barbara Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1442–1449, 2014.

[14] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54, 2013.

[15] Bernardino Romera-Paredes, Min SH Aung, Massimiliano Pontil, Nadia Bianchi-Berthouze, Amanda C de C Williams, and Paul Watson. Transfer learning to account for idiosyncrasy in face and body expressions. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.

[16] Nitish Srivastava and Ruslan R Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems*, pages 2094–2102, 2013.

[17] Tatiana Tommasi and Barbara Caputo. The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In *BMVC*, number LIDIAP-CONF-2009-049, 2009.

[18] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3081–3088. IEEE, 2010.

[19] Hua Wang, Feiping Nie, Heng Huang, and Chris Ding. Dyadic transfer learning for cross-domain image classification. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 551–556. IEEE, 2011.

[20] Wei Wang, Hao Wang, Chen Zhang, and Fanjiang Xu. Transfer feature representation via multiple kernel learning. In *AAAI*, pages 3073–3079, 2015.

[21] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.

[22] Tatiana Tommasi, Francesco Orabona, Mohsen Kaboli, and Barbara Caputo. Leveraging over prior knowledge for online learning of visual categories. In *British Machine Vision Conference*, 2012.

[23] Mohsen Kaboli. Leveraging over Prior Knowledge for Online Learning of Visual Categories across Robots. In *KTH, The Royal Institute of Technology-Stockholm-Sweden*, 2012.

[24] Mohsen Kaboli, Philipp Mittendorfer, Vincent Hugel,and Gordon Cheng. Humanoids learn object properties from robust tactile feature descriptors via multi-modal artificial skin. In *IEEE International Conference on Humanoid Robots*, pages 187–192. IEEE, 2015.

[25] Mohsen Kaboli, T De La Rosa, Rich Walker, and Gordon Cheng. In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors. In *IEEE International Conference on Humanoid Robots*, pages 1155–1160. IEEE, 2015.

[26] Yogeswaran, Nivasan and Dang, Wenting and Navaraj, William Taube and Shakthivel, Dhayalan and Khan, Saleem and Polat, Emre Ozan and Gupta, Shoubhik and Heidari, Hadi and Kaboli, Mohsen and Lorenzelli, Leandro and others. New materials and advances in making electronic skin for interactive robots. In *Advanced Robotics*, pages 1359–1373. Taylor & Francis, 2015.

[27] Mohsen Kaboli, Alex Long, and Gordon Cheng. Humanoids learn touch modalities identification via multi-modal robotic skin and robust tactile descriptors. In *Advanced Robotics*, pages 1411–1425. Taylor & Francis, 2015.

[28] Mohsen Kaboli and Gordon Cheng. Novel Tactile Descriptors and a Tactile Transfer Learning Technique for Active In-Hand Object Recognition via Texture Properties. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*,IEEE, 2016.

[29] Mohsen Kaboli, Kunpeng Yao, and Gordon Cheng. Tactile-based manipulation of deformable objects with dynamic center of mass. In *IEEE International Conference on Humanoid Robots*, pages 752-757. IEEE, 2016.

[30] Mohsen Kaboli, Rich Walker, and Gordon Cheng. Re-using prior tactile experience by robotic hands to discriminate in-hand objects via texture properties. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2242-2247. IEEE, 2017.

[31] Mohsen Kaboli, Di Feng, Kunpeng Yao, Pablo Lanillos, and Gordon Cheng. A Tactile-based Framework for Active Object Learning and Discrimination using Multi-modal Robotic Skin. In *IEEE Robotics and Automation Letters*, pages 2143-2150. IEEE, 2016.

[32] Mohsen Kaboli, Rich Walker, and Gordon Cheng. Active Tactile Transfer Learning for Object Discrimination in an Unstructured Environment using Multi-modal Robotic Skin. In *International Journal of Humanoid Robotics (IJHR)*,World Scientific, 2017.