Transfer Learning from User's App Profile to Content Profile

LIN Jianxia
Hong Kong University of Science and Technology
Hong Kong, China
jlinax@connect.ust.hk

WEI Huan
Hong Kong University of Science and Technology
Hong Kong, China
hweiad@connect.ust.hk

WEI Yujun
Hong Kong University of Science and Technology
Hong Kong, China
yweiar@connect.ust.hk

CHEPENKO Daniil
Hong Kong University of Science and Technology
Hong Kong, China
dchepenko@connect.ust.hk

ACM Reference Format:
LIN Jianxia, WEI Huan, WEI Yujun, and CHEPENKO Daniil. 2017. Transfer Learning from User's App Profile to Content Profile. In *Proceedings of MSBD 5001 Project Report.* ACM, New York, NY, USA, 4 pages. https://doi.org/0.000/0000

1 INTRODUCTION

1.1 Background

When a new app starts, it lacks data on user behavior (User behavior is important for advertising and preference recommendation). This will cause app recommendation fail because nowadays recommendation system are mostly based on machine learning which requires a big mount of data. So, there is always a cold start problem, because of lacking data in project domain especially in recommendation system. This can be solved by transfer learning when we have data in a similar domain. We aim to solve the problem by transfer learning target domain to a similar domain with high confidence and get the predictive result.

1.2 Problem Definition

1.2.1 Domain Definition

. A domain D consists of two components: feature space $\mathbf{X}$ and a marginal probability distribution *P(X)*. So a specific domain $D = \{\mathbf{X}, P(X)\}$ and $X = \{x_1, x_2, ..., x_n\} \in \mathbf{X}$. x_i is a vector of attributes $x_i = (attr_1, attr_2, attr_3, ..., attr_n)$.

1.2.2 Task Definition

. A task consists two components: a label space $\mathbf{Y}$ and an objective predictive function $f(\cdot)$, $Task = \{\mathbf{Y}, f(\cdot)\}$

1.2.3 Transfer Learning

. Give a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S. In our task, $D_S \neq D_T$ and $T_S \neq T_T$

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
MSBD 5001 Project Report, December 2017, HKUST, Hong Kong
© 2017 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/17/10.
https://doi.org/0.000/0000

1.3 Related Works

In the paper of *Domain-Adversarial Training of Neural Networks* [2]:The research scope of the paper describes the new representation learning approach for domain adaptation, in which data at training and test time come from similar but different distribution. This approach does not require labeled target-domain data, but at the same time, it can easily incorporate such data, when they are available. The approach is motivated by the domain adaptation theory of Ben-David. During training neural network and the domain regressor are competing each other. Neural network is referred to be trained as Domain-Adversarial Neural Networks (DANN). The main idea behind DANN is to join the network hidden layer to learn a representation which is predictive of the source example labels, but uninformative about the domain of the input (source or target). Author provided experiments both shallow and deep neural networks. The research shows, that these approach is flexible and achieves state-of-art results on a variety of benchmarks in domain adaptation, namely for sentiment analysis and image classification tasks. The following approach describes one of setting of transfer learning, in case when labeled data is available only in the source domain. For our specific case, the results, that proposed DANN algorithm showed in sentiment analysis, could help to find a representation suitable for the target domain.

In the paper of *Ridesourcing Car Detection by Transfer Learning* [5]:This paper proposes a method to detect ridesourcing cars form a pool of vehicles based on their trajectories. For this purpose, a two-stage transfer-learning framework was designed. The first stage, with taxi and bus open data as input, identifies ridesourcing cars using features that are shared by both taxis and ridesourcing cars and can withstand the open data limitation like time misalignment. Using these features a random forest (RF) classifier and adopt it to predict whether it is ridesourcing car or not. On the second stage, taking the cars, classified with high confidence leverages a co-training process to further infer ridesourcing-speci c features. Two classifiers, RF and CNN, are iteratively refined, and the final ridesourcing label for each candidate car is determined by an ensemble of the two classifiers. AUC and top-k% precisions were used as an evaluation metrics. The experiment results show that, with no need of any pre-labeled ridesourcing car dataset, described method can achieve a comparable detection accuracy as the supervised learning methods.

For our specific case the design of the described method could be applied. As each category in our source domain could be seen as a special case of a corresponding category in the target domain, so the proper design of feature space is a crucial task.

In the paper of *A survey on transfer learning* [4]:It introduces various transfer learning, such as inductive transfer learning, unsupervised transfer learning and transductive transfer learning, illustrates the relationship between the above transfer learning settings and traditional machine learning. In the inductive transfer learning setting, the target task is different from the source task, no matter when the source and target domains are the same or not. In the transductive transfer learning setting, the source and target tasks are the same, while the source and target domains are different. In the unsupervised transfer learning setting, the target task is different from but related to the source task.

In the paper of *Mobile recommendations based on interest prediction from consumers installed appsCinsights from a large-scale field study* [1]:It illustrates that nowadays most existing profiling approaches on mobile suffer problems such as non-real-time, intrusive, cold-start, and non-scalable, which prevents them from being adopted in reality. To tackle the problems, this work developed real-time machine-learning models with app-based method to predict user profiles of smartphone users from openly accessible data.

2 DESIGN AND IMPLEMENTATION

2.1 Overview of Solution

Since the data from source domain and data from target domain have the different distributions, we cannot use the model trained by training data to predict the article categories that users like. The model we proposed is called DANN, which is a good method to deal with the domain adaption problem. We use transfer learning technology to modify the model trained by using source domain data in order to make this model also fit target domain.

2.1.1 *Analysis and Select features*

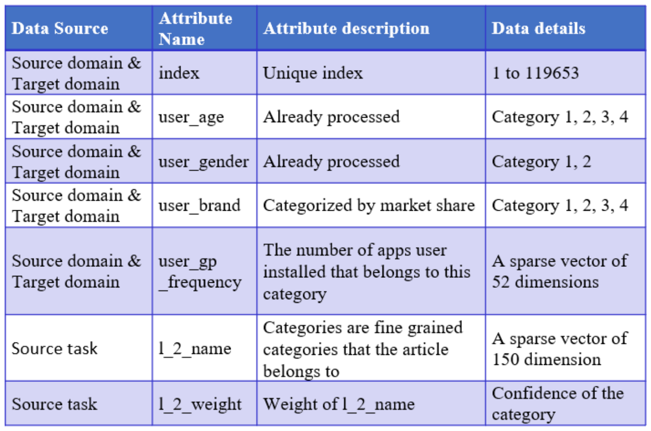
. In our analysis, we think we can infer article type a person like from his user_age, user_gender, user_brand and user_gp_frequency. And L_2_category contains detail information to predict the result. So we select L_2_category as a predict target.

2.1.2 *Data Clearing*

. Delete records which miss necessary information. Original data has 170,126 records and cleaned data has 119,653 records.

2.1.3 *Categorize and Feature expansion*

. We select 4 features as input–user_age, user_gender, user_brand and user_gp_frequency. We categorized user_brand by its market share and expanded user_gp_frequency for its a vector of 52 dimensions. Then we get a vector of 55 dimensions which is our input. We split L_2_category and expand it to 150 dimension vectors.



Data Source	Attribute Name	Attribute description	Data details
Source domain & Target domain	index	Unique index	1 to 119653
Source domain & Target domain	user_age	Already processed	Category 1, 2, 3, 4
Source domain & Target domain	user_gender	Already processed	Category 1, 2
Source domain & Target domain	user_brand	Categorized by market share	Category 1, 2, 3, 4
Source domain & Target domain	user_gp _frequency	The number of apps user installed that belongs to this category	A sparse vector of 52 dimensions
Source task	l_2_name	Categories are fine grained categories that the article belongs to	A sparse vector of 150 dimension
Source task	l_2_weight	Weight of l_2_name	Confidence of the category

Figure 1: Processed Data Description

2.2 Insight of Solution

2.2.1 *Different Part of DANN*

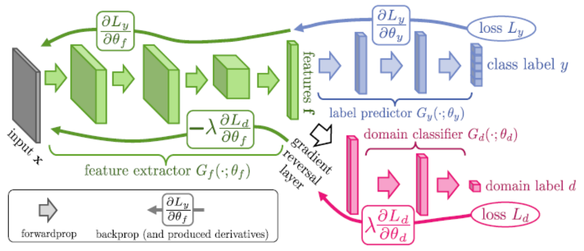
. There are three parts of this DANN model, the domain projector, category classifier and domain classifier. The main idea of the model is that, we want to project the input data from source domain and target domain into a new space with another dimension, then, we suppose if category classifier can predict the category correctly for users from source domain, it can also predict the category correctly for users from target domain. Domain classifier can detect whether the data from source domain and target domain have the same distribution.

2.2.2 *Loss Function and Activation Function*

. The output of category classifier is the probability of users preference of each category, so we use MSE to calculate the loss. The output of domain classifier is the probability the data belong to source domain, we also use MSE to calculate the loss.

2.2.3 *How to update the parameter?*

We use MES method to calculate the loss function of both category classifier and domain classifier, we use the gradient descent method to update to parameter of the whole model. But, here is a key point that the loss function of category classifier update $Loss_y$ updates a positive gradient descent to the all parameter relating to this loss function. But for the loss function of domain classifier $Loss_d$, it updates a positive gradient descent to domain classifier, but updates a negative one to projector, since we want the projector projects the data from source domain and target domain into a new space, also we want domain classifier can be more and more accurate.



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features f
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input x
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
$\lambda \frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
backprop (and produced derivatives)

Figure 2: DANN Architecture

2.3 Implementation Details

2.3.1 *Deal with source domain labels:*

Because we have about 5800 users with their information details in source domain, and also have 110 thousand reading log of these users. The reading log contains several categories, article title, article update time, category 1, category 2, category 1 weight, category 2 weight etc.

The goal of this DANN model is to predict the article category that user like. So we choose the category 2 and category weight as label of each input, which is a 150 vector, each element is the probability that user like this article category.

2.3.2 *Projector:*

The DANN model has a projector with 2 layers, the first layer project input data, which is 55 dimensions, into 100 dimensions, and the second layer is 100 to 100 dimensions projection. The activation function of both two layers are RELU.

2.3.3 *Category Classifier:*

The Category classifier has just one layer without activation function, and use softmax function to generate the probability of each category.

2.3.4 *Domain Classifier:*

The Domain classifier has just one layer without activation function, and use sigmoid function to generate the probability of the data belongs to source domain.

3 EVALUATION

3.1 Experimental Setup

If the number of categories which user read is larger than 5,

$$accuracy = \frac{TP}{5}$$

If the number of categories which user read is no more than 5,

$$accuracy = \frac{TP}{number\ of\ categories\ user\ read}$$

3.2 Performance Metrics

There are large diversity of available metrics to evaluate recommender systems. Researches trying to introduce new metric, when they evaluate their systems. The common practice involving such estimation as RMSE, Precision vs Recall, top-N recommendation. We used $F_1 score$ to evaluate the performance of the model.

$$F_1 = 2 \times \frac{precison \times recall}{precision + recall}$$

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

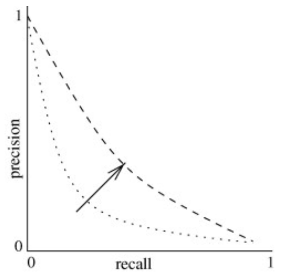


Figure 3: Optimization of precision and recall

3.3 Experimental Results

Our max training accuracy up to 48.13% and max test accuracy up to 48.85%. We also did a training on the dataset with decision tree, it has higher performance of 36.52% training accuracy. So, if we can make a proper transfer, we can reach a higher accuracy. We still have a far way in learning transfer learning.

For article accuracy, we select five categories and select top 1 article from each category as recommendation. In source domain, training data article accuracy is 5.01%. In target domain our article accuracy is 1.32%. A random select of article recommendation's accuracy is only 0.013%.

4 CONCLUSION AND FUTURE WORK

4.1 Main Results of Project

The result is not so good, because of the data limitation and information lose. For example, we cannot predict the users' attitude towards particular article as we are working only with monthly data. More over the data is skewed to frequent users. There is a large variation of users preferences exist among top 1000 users. The 10% of them read more that 30 different categories, and 75% more than 10. Also, the frequent users are generation the major part of read records. Top 1000 users generate 76% records. And we can not make use of every feature well in transfer and prediction when we select some features then we lost other information.

Project link: https://github.com/huanhuanBOY/homeworks/tree/master/5001/project

4.2 Thoughts and Suggestions for Follow-up Work

During data preprocessing, we deleted the records which have missing value in specific attributes. It may lose some information. To avoid missing too much useful information, we can use SVD to predict the missing value.

Latent factor models, such as Singular Value Decomposition (SVD), comprise an alternative approach by transforming both items and users to the same latent factor space, thus making them directly comparable. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. [3] So, better data preprocessing may improve the model selection and accuracy.

5 TEAM MEMBERS

Lin Jianxia 20459661
WEI Huan 20463442
WEI Yujun 20475524
CHEPENKO Daniil 20478954

REFERENCES

[1] Remo Manuel Frey, Runhua Xu, Christian Ammendola, Omar Moling, Giuseppe Giglio, and Alexander Ilic. 2017. Mobile recommendations based on interest prediction from consumer's installed appsCinsights from a large-scale field study. *Information Systems* 71, Supplement C (2017), 152 – 163. https://doi.org/10.1016/j.is.2017.08.006

[2] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial Training of Neural Networks. *J. Mach. Learn. Res.* 17, 1 (Jan. 2016), 2096–2030. http://dl.acm.org/citation.cfm?id=2946645.2946704

[3] Helge Langseth and Thomas Dyhre Nielsen. 2012. A Latent Model for Collaborative Filtering. *Int. J. Approx. Reasoning* 53, 4 (June 2012), 447–466. https://doi.org/10.1016/j.ijar.2011.11.002

[4] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[5] Leye Wang, Xu Geng, Jintao Ke, Chen Peng, Xiaojuan Ma, Daqing Zhang, and Qiang Yang. 2017. Ridesourcing Car Detection by Transfer Learning. *CoRR* abs/1705.08409 (2017). http://arxiv.org/abs/1705.08409