A visual analytical approach for transfer learning in classification

Yuxin Ma [a], Jiayi Xu [a], Xiangyang Wu [c], Fei Wang [a], Wei Chen [a,b,∗]

[a] State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China
[b] Innovation Joint Research Center for Cyber-Physical-Society System, Zhejiang University, Hangzhou 310058, China
[c] Hangzhou Dianzi University, Hangzhou 310018, China

ABSTRACT

Classification can be highly challenging when the dataset is extremely large, or when the training data in the underlying domain are difficult to obtain. One feasible solution to this challenge is transfer learning, which extracts the knowledge from source tasks and applies the knowledge to target tasks. Extant transfer learning schemes typically assume that similarities between the source task and the target task to some degree. This assumption does not hold in certain actual applications; analysts unfamiliar with the learning strategy can be frustrated by the complicated transfer relations and the non-intuitive transfer process. This paper presents a suite of visual communication and interaction techniques to support the transfer learning process. Furthermore, a pioneering visual-assisted transfer learning methodology is proposed in the context of classification. Our solution includes a visual communication interface that allows for comprehensive exploration of the entire knowledge transfer process and the relevance among tasks. With these techniques and the methodology, the analysts can intuitively choose relevant tasks and data, as well as iteratively incorporate their experience and expertise into the analysis process. We demonstrate the validity and efficiency of our visual design and the analysis approach with examples of text classification.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Conventional data analysis approaches assume that the training and test data are drawn from the same feature space and the same distribution. Knowledge transfer, also known as transfer learning, seeks to fulfill a new task by leveraging the knowledge and insights from accomplished tasks. The mechanism of transfer learning is similar to the "learning by analogy" approach, which understands a new situation by leveraging familiar conditions [21]. The "*source analog*" is an existing situation that conveys information and reference in exploring a novel "target", with its main concern found in the similarity between the source and the target. The concept of "learning by analogy" that comes from cognitive psychology is relatively abstract, whereas transfer learning algorithms provide a formal description of analogy between learned tasks and new tasks in machine learning. The new task can be in the same scenario with the source tasks or even a different scenario, given that a type of analogy exists between the tasks or the underlying datasets. As such, transfer learning can be used to

---

∗ Corresponding author at: State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China.
E-mail addresses: mayuxin@zju.edu.cn (Y. Ma), jiayi.xu64@gmail.com (J. Xu), wuxy@hdu.edu.cn (X. Wu), wolflyecn@gmail.com (F. Wang), chenwei@cad.zju.edu.cn (W. Chen).

analyze a large amount of data by first training a part of it and transferring the result to the remaining data [40]. Similarly, knowledge transfer can be performed between two tasks performed in two different applications.

Transfer learning is truly beneficial because it can significantly reduce the cost and burden of exploring unknown data and unknown scenarios, and help discover common features hidden in tasks and data from different application fields. One example is the classification of web documents, in which bundles of data from different sources (e.g., forums, news press, and websites) are handled. Each data source has a specific word distribution, and leads to a unique data classification model. For a new data source, directly applying existing classification models will cause a performance drop. Transfer learning has proven to be significantly useful [14].

Despite the progress of transfer learning, many challenges exist regarding its usage in real applications. The most important challenge is the estimation of the *transferability* between the task pairs or the data pairs. The transferability actually measures the capability of the knowledge transfer for a specific task. With an appropriate transferability, the analyst can easily choose existing analysis results analogous to the target task, and perform the transfer. Conversely, determining the *reusability* of data instances for the new task is non-trivial. Both challenges cannot be fully addressed with an automatic transfer learning process when handling a complicated scenario, and may be aggravated by the complicated transfer relations and the non-intuitive transfer process.

We argue that applying visualization techniques in the learning process would be an effective means for incorporating the human intelligence into complicated analysis tasks. In particular, visualization technologies that integrate the intelligences from both the machine and human within a visual information communication interface have achieved significant success on many areas, including classification [17,23], summarization, and clustering [11,25,49]. These techniques could be certainly adapted to the transfer learning process to bridge the gap between the domain experts and the transfer learning approaches.

This paper presents a pioneering visual-assisted transfer learning scheme in the context of text classification. The core idea of our work lies in our design of a suite of visualization communication and interaction techniques to enhance the analyst's understanding and the manipulability of the knowledge transfer process. By incorporating the domain experiences and expertise into the knowledge transfer process within a visual exploration loop, the analyst can intuitively study the relevances among various tasks and data, and identify similar tasks. For analogous tasks, the analyst can interactively choose the related data instances, and use the trained data for a new task, which is supported by a pioneering visual representation of the classification result and the data instance similarity. To our best knowledge, our work is the first effort to apply visualization methods to the knowledge transfer process, and leads to the following main contributions:

- a suite of visual communication and interaction techniques that support the knowledge transfer process;
- a pioneering visual analytics based transfer learning methodology capable of analyzing relevances between different tasks and knowledge transfer on the level of data instances;
- an exploratory data classification prototype called TransXplorer that follows the proposed techniques and methodology, and demonstrates high efficiency for text classification.

The rest of the paper is organized as follows. Section 2 summarizes related works. Section 3 provides the overview of the entire transfer learning framework. Sections 4 and 5 present detailed descriptions of its two stages. Section 6 presents experimental results and analysis. Section 7 present discussion and Section 8 draws conclusions.

2. Related work

A large body of literature is devoted to machine learning and text classification. Below, we briefly review the most relevant ones.

Text classification: Text classification deals with text feature representation and classification rules. First, feature representation plays an important role in text classification. Each text classification task requires specific text feature measurements. Successful text classification methods should choose the classification criteria relatively suitable for the text classification object and text data types. Common text features include Mutual Information, Latent Semantic Indexing [41], Linear Discriminant Analysis [33], and so on. Suitable feature selection is the key of the text classification process. Then, valuable feature representation selection rules from source tasks should be regarded as knowledge transferred to target tasks.

Second, as to classification rules, different text data require specific classification methods to effectively account for text features. Some common classification methods include decision trees [22], pattern-based classifiers [22], SVM classifiers [47], Neural Network classifiers [22], Bayesian Classifiers [22], and so on. Each classifier is designed to satisfy a concrete text classification. For example, decision trees aimed at hierarchical division of the underlying data space using different text features; SVM classification is suitable for high-dimensional sparse features; Naive Bayesian is used for some simple application environments.

Different tasks require different text classification methods, which indicate that human knowledge could play an important role in text classification because humans can use their experience and expertise to decide suitable text classification features and classification rules.

Visual classification: Classification of multi-dimensional data is one of the major challenges in data mining. Recently, the idea of combining classification with interactive visualization has gained significant interest in the literature, and has been shown to be highly effective. A representative and prior visual classification system is the PBC (Perception-Based Classifier) approach [4], which integrates the domain knowledge of an expert in the tree construction phase. Furthermore, many interactive systems [17,31] are described for interactive construction, visualization, and exploration of decision trees, which have a common classification method. For probabilistic classifiers, Alsallakh [1] proposed a visual analysis approach for analyzing performance of classifiers from different aspects of classification results. Some systems, such as the EnsembleMatrix [48], combine multiple classifiers to build one superior to its components. Interactive visual classifier has stepped into multiple application areas, such as text analytics [8,26], image classification [39], social networks [2], dimension reduction [18], and so on.

Typically, a classification system is first trained with a set of labeled data. Thus, annotating labels is important, for example. Steed [46] developed a system to rank retrieval results, and found new relevant documents according to user feedback. Moehrmann [34] presented an advanced user interface suitable for non-expert users' usage. Choo [12] showed us an interactive classification system, iVisClassifier, based on linear discriminant analysis (LDA).

Cluster performance can be improved using several interactive cluster tools. ClusterSculptor [38] allowed the user to iteratively and interactively apply different clustering parameters of *K*-means to different parts of a dataset. The Morpheus system [37] supported the knowledge discovery process through visualization and interactive exploration of subspace clusterings. Schultz [43] presented an open-box spectral clustering approach to visualize the mathematical quantities, suggest parameter values, and provide feedback for decision making. Andrienko [3] proposed an approach to extract meaningful clusters from large databases using an interactive visual interface from which a human analyst can combine clustering and classification. Interactive learning is further used to refine the regression model [16,35] and the distance function [9]. Bertini [6] performed a literature review of papers in data mining and information visualization, and then categorized the observed techniques in classes. Muhlbacher [36] summarized strategies commonly used to increase user involvement in automatic data analysis approaches.

Visual data mining and visual analysis: Machine learning is typically an automatic process requiring limited user guidance. Recently, interactive machine learning is becoming a hot topic, where the integration of human and machine capabilities is advocated. Of great importance is the integration of human intelligence into knowledge discovery process by leveraging a well-designed visual interface. For example, interactive 2D visualization has been used to display projections of high-dimensional data for effective data exploration [10,27]. Tools that can modify the projection parameters are widely studied [18]. Visualization can further help users understand the pattern distribution hidden in dataset and assess their quality. Cui [13] defined two data abstraction quality measures for computing the degree to which the abstraction conveys the original dataset. Yang [50] presented techniques for visualization and interactions with the clusters, outliers, associations, and any other patterns extracted from the dataset, which supported a sense-making process. Feature subset selection is a preliminary step in data analysis. The SmartStripes system [32] can be used for feature subset selection.

Active learning has been proven to be a reliable scheme to reduce manual efforts in data labeling. Many approaches integrate the active learning techniques with interactive visualization to increase labeling efficiency. Fogarty [19] presented CueFlik, a novel approach for web image search based on interactive concept learning. Another approach allowed [24] users to manually annotate data instances and directly adjust complex classifier models. Seifert [44] proposed a visual-assisted active learning scheme that allows the user to select instances and label them with a trained classifier. Similarly, Heimerl [23] presented a system to interactively train a support vector machine model for text document retrieval by employing a visual analytics process. A new type of dual annotation interface is suitable for semi-supervised active learning [45], which can be used to rapidly develop useful natural language systems for various tasks.

3. Approach overview

For clarity, we first explain the related concepts involved in transfer learning and text classification.

Background: For simplicity, we will describe the knowledge transfer framework in the context of *binary classification* of text. First, the bag of word (BOW) model is employed to vectorize each document, yielding a vector representation.

Task, domain, and model: Compared with the conventional classification approaches, multiple datasets(document collections) and associated trained models are involved when applying transfer learning to text classification. A collection of documents is defined as a data domain, and each document is represented with a BOW. Thus, a set of word frequency vectors can be built for each document. In classifying a document collection, the labeled data are considered as the training data. Furthermore, the corresponding *classification model* is produced. The tuple of a data domain and its associated model are regarded as a *task*.

Notably, among different data domains (news in different years), the frequency distributions may be similar. Conversely, the built classification model may vary in analyzing a specific data domain. For example, when analyzing sentiments on various sets of the comments on different commodities from an e-commerce website, the trained sentiment models can be different caused by the variation of the word frequency in the data domain. The differences among the trained classification models are obstacles to share them among different data domains. This observation motivates our work.

Essentially, transfer learning in the context of binary classification is defined as follows: *transfer of data and the classification model of selected source tasks to the destination tasks to train an optimal classification model for later*. Instead of taking an
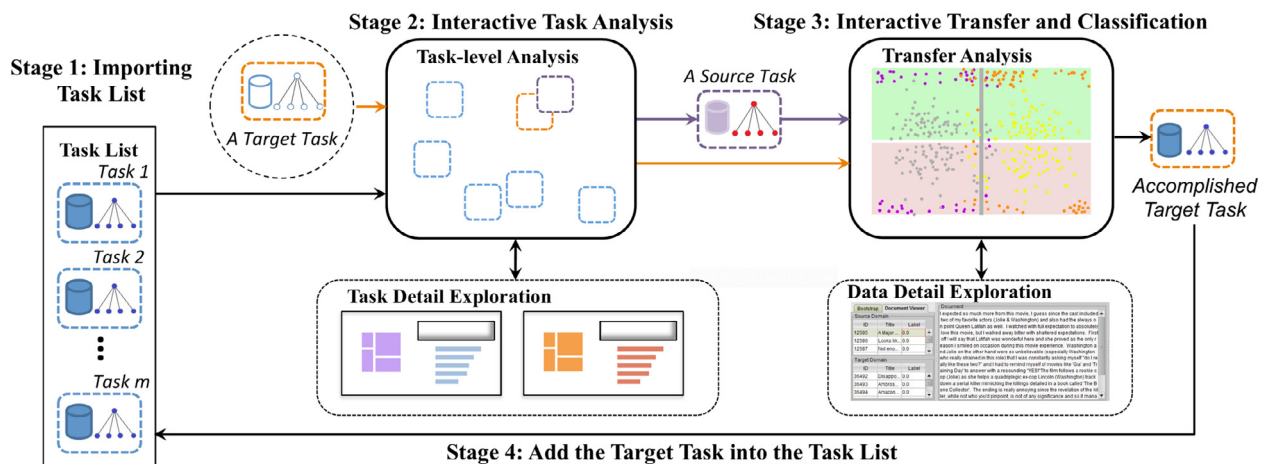
Fig. 1. The conceptual overview of our approach.

automatic knowledge transfer process, our integrated system, namely TransXplorer, leverages the human intelligence to analyze the relations between the tasks and the datasets, and make decisions in fuzzy and complicated situations. Specifically, TransXplorer assists the analyst to perform visual exploration of task relevances and visual classification transfer through the following stages (Fig. 1):

Stage 1: Given a target task, the analyst specifies a set of source tasks as well as the associated classifications from a task list. The classification results are supposed to contain an appropriate classification model.

Stage 2: The relevances between the source tasks and the target tasks are computed and depicted in the task exploration view. The analyst can visually study the tasks and their relevances, add source tasks, and specify the most relevant source tasks to the underlying target task.

Stage 3: The analyst fulfills the target task with the assistance of a task transfer view. The knowledge from the source tasks is iteratively extracted, and used to enhance the classification model of the target task.

Stage 4: The accomplished target tasks are added into the task list. Stages 1–4 are iteratively performed until the tasks are considered accomplished.

Fig. 2 shows the interface of TransXplorer, which contains the following four main views: the task exploration view, the task transfer view, associated task, and data details views. The details of Stage 2 and Stage 3 are explained in Sections 4 and 5, respectively.

4. Interactive task analysis

Conventional transfer learning approaches assume that the source task/data are highly analogous to the target task/data. In practical applications, what makes the knowledge transfer infeasible is the uncertainty of the analogy degree between the source and target parts. If the transfer is performed between two non-analogous tasks, the transfer performance can be extremely low, and can even negatively influence the analysis of the new task. Thus, a generalized and model-free metric of task relevance is necessary. Furthermore, the analyst should understand the task details. Below, we first describe the exploration and analysis of tasks and their details in the context of classification.

4.1. Computation of task relevance

As mentioned above, a classification task contains two components, namely, data domain and classification model. Correspondingly, the following two factors are considered when measuring the relevance of two tasks, namely, the data domain similarity and the transferability between them. They characterize the relevance in a static (data) view and a dynamic (the training process) view, respectively.

Domain similarity: The data domain similarity (named as *dds*) describes the proximity between the data domains of two tasks. It only relates to the data properties while being independent to the classification model and the analysis process. The data domain similarity can be employed to measure the data relations in different tasks and in terms of the data distribution. In this sense, two tasks whose data domains are similar, are potentially relevant and will benefit from the knowledge transfer. More importantly, the analyst is able to estimate the data domain similarity between a new target task and a source task before the data analysis. These observations indicate that the data domain similarity is irrelevant to the analysis tasks and process. Thus, appropriate measures can be chosen by considering the data types and features of datasets. In Table 1, we provide several similarities and distance measures. Note that for the distance measures, we use its reciprocal of the distance value as the similarity value.
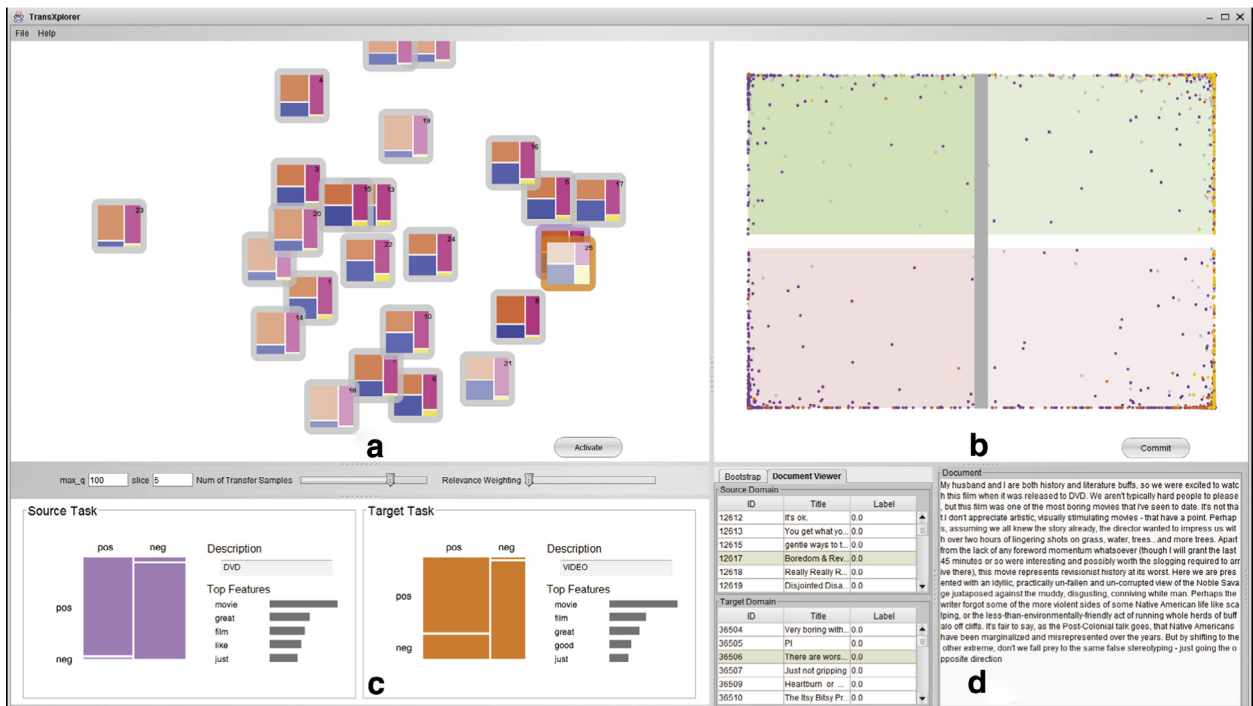
Fig. 2. The interface of TransXplorer. (a) The task exploration view. (b) The task transfer view. (c) The task details view. (d) The data details view. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)
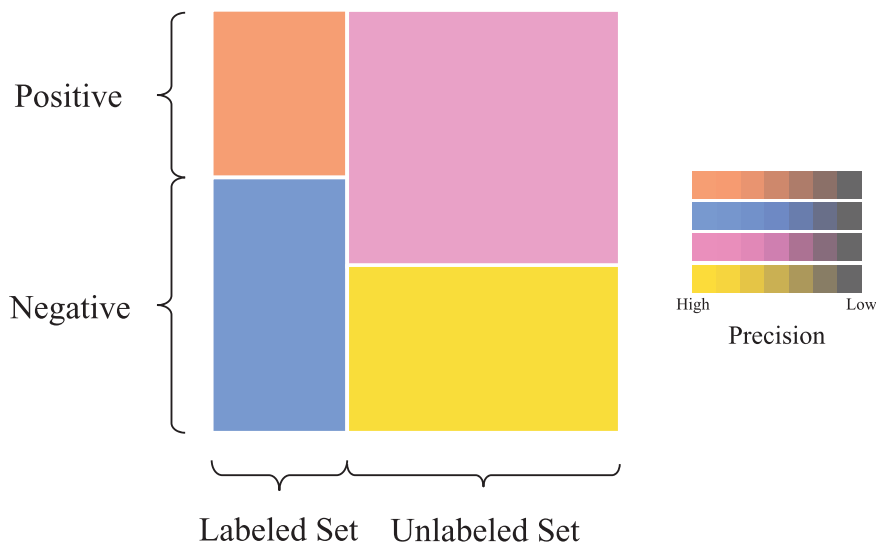
Fig. 3. A colored treemap encodes six attributes of a task: the total number of data instances (area), the overall precision of the classification model (color tone), the numbers of labeled/unlabeled and positive/negative data instances (areas of cells). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In terms of text classification, the cosine coefficients can be employed to measure the similarity of word frequency vectors, which are widely used in document analysis.

Task transferability: Apart from the similarity between two data domains, the relevance between the corresponding classification models significantly influences the task relevances. Furthermore, a measure of the task relevance should be independent to the application situations of the classification models because the analyst may use distinctive classification models when handling different tasks. Task transferability is introduced for addressing the problems caused by the differences of classification models.

Table 1

Measures for computing similarity between data domains. In the cosine similarity and Euclidean distance, $\boldsymbol{m}_i$ is the mean of all the data instances in domain D_i. For $\mathcal{A}$-distance, P_1 and P_2 are the probability distributions over D_1 and D_2, and $\mathcal{A}$ a subset of labeled data instances in the measure space of D_1 and D_2.

Metric	Definition
Cosine similarity	$similarity(D_1, D_2) = \cos(\theta) = \frac{\boldsymbol{m}_1 \cdot \boldsymbol{m}_2}{|\boldsymbol{m}_1||\boldsymbol{m}_2|}$
Euclidean distance	$d(D_1, D_2) = \sqrt{(\boldsymbol{m}_1 - \boldsymbol{m}_2) \cdot (\boldsymbol{m}_1 - \boldsymbol{m}_2)}$
$\mathcal{A}$-distance [5,30]	$d_{\mathcal{A}}(D_1, D_2) = d_{\mathcal{A}}(P_1, P_2) = 2\sup_{A \in \mathcal{A}} |P_1(A) - P_2(A)|$

For a pair of source and target tasks (s, t), the task transferability is designed by measuring the classification performance p and the performance variation pv. We denote $p(m)$ as the performance of the target task by using additional m training data. The performance variation is defined as follows: $pv(m) = p(m) - p$. Apparently, $pv(\cdot)$ may be a negative value, which indicates an unsuccessful transfer. When both tasks are analogous, the performance variation $pv(\cdot)$ can be positively high. Generally, it varies with the parameter m. To describe this dependency, the task transferability is described as

$$tf(s, t) = \left(\sum_{i=0}^{m/q} pv(iq) \right) / q \tag{1}$$

where q is a stepping variate.

The transferability denotes an average of multiple performance variation values. The variation of the transferability can be modulated by adjusting the parameters m and q. The task transferability is extremely suitable for well-trained tasks whereas it is ineffective for new tasks. Notably, the described transferability is similar to the one proposed in [15]. Instead of defining it as a specific point q, we describe it as the average over the entire interval of the point.

The task relevance t is defined as the weighted sum of two measures mentioned above as follows:

$$t = [a \times dds, t + (1 - a) \times \min(0, tf_s, tf_t)] \tag{2}$$

where a is an analyst-adjustable variant for weighting the similarity and the transferability.

Varied weights obviously lead to different implications of task relevance. The analyst can freely adjust the weights in different situations. When a large weight is adopted for the domain similarity, the computed task relevance is close to the relation between the data distribution, rendering the classification model inaccurate. Meanwhile, it further allows for computing the relevance between a new task without a classification model and others by leveraging the characterizations of data domains. The weight of the transferability can be enlarged when the classification models are trained well, enabling the task relevance to better describe the influences of the models.

4.2. Visualization and interaction of tasks

The integrated system, TransXplorer, employs a visual interface for studying the relevance among a set of source tasks and the underlying target task and choose the most suitable source tasks. Two views are specifically designed for exploring tasks, task relevance, and task details.

Task exploration view: In the task exploration view (Fig. 2(a)), a set of selected source tasks and the target task are placed with the standard force-directed layout algorithm [20]. Each task is represented with a colored 2×2 treemap. In particular, the treemap area encodes the total number of data instances in the task, and the tone of the filled color encodes the overall precision of the classification model of the task. The horizontal (primary) separation of the treemap denotes the numbers of labeled and unlabeled data instances in the task, and the vertical (secondary) separation represents numbers of positive and negative instances. The rationale of applying the force-directed layout is to make an analogy between the task relevance and the force. The shorter distance between a source task node and the target task node in the view indicates that the corresponding source task is more relevant than any other source tasks, which can be easily conveyed to the analyst.

Task details view: When a task is selected in the task exploration view, the detailed information is shown in the task details view (Fig. 4). A 2×2 colored treemap is used to show the prediction precision of the classification model: the horizontal separation encodes the ground truth of the data instances (positive vs negative), and the classification result (positive vs negative) is represented vertically. The design of the colored treemap is derived from the confusion matrix technique widely used in the field of machine learning [28]. In the context of binary classification, we improve the simple matrix form by using the treemap layout [29] to represent the proportions of data instances by area. Furthermore, the textual information of the task description and the weights of top informative words are shown. For text analysis, the top informative words can reflect the distribution of words in the dataset, and provide a summarization of the text contents.

User interactions: The analyst can visually explore the attributes of each task, and compare multiple tasks in the task exploration view. The task details can be further analyzed in the task details view. Furthermore, the analyst can modulate the weights of the domain similarity and transferability to view the variation of the task relevance as well as the layout.
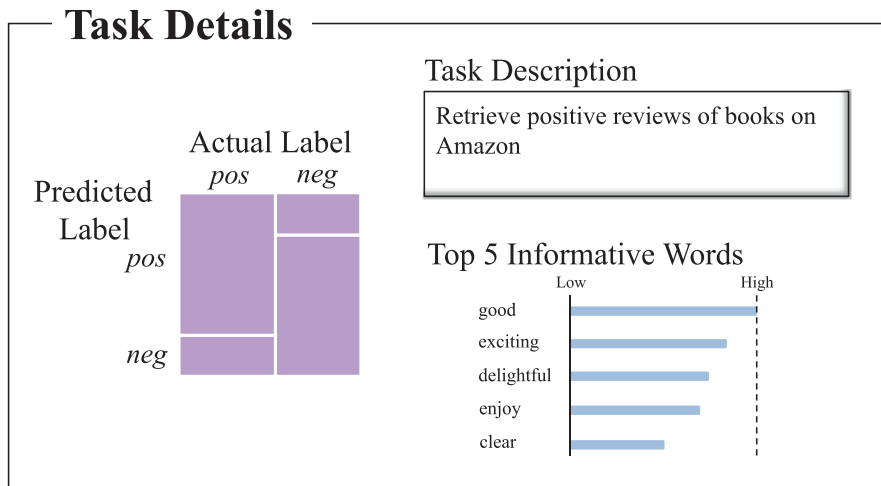
Fig. 4. The task details view. Left: a treemap encodes the prediction precision of the classification model. Top right: the task description. Bottom right: the weights of top informative words are displayed with bar charts. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Furthermore, the analyst can select a group of source tasks after investigation, and put them in the subsequent transfer learning process together with the underlying target task.
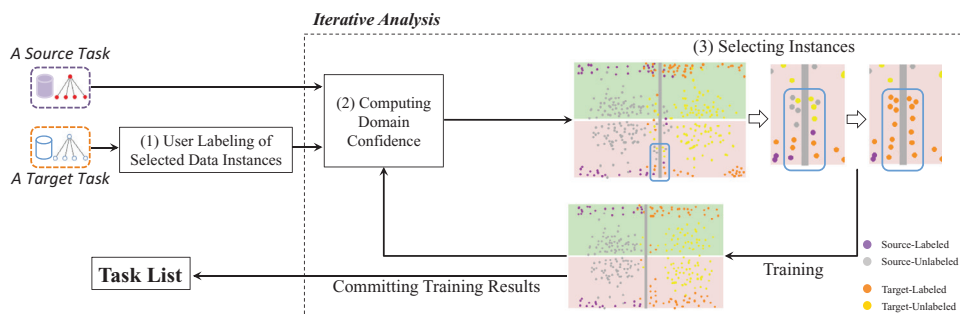
5. Interactive transfer and classification

Efficiency of the target task not only depends on the selected source tasks, but is further influenced by the data instances used for training its classification model. Thus, one challenging problem in this stage is to find appropriate data instances from the source tasks, and transfer them to the target task. With them, the target task is accomplished. This is an iterative process, which runs until the analyst collects adequate data instances, and then accomplishes the target task.

5.1. Iterative classification transfer

Conceptually, the transfer from the source tasks to the target task consists of three steps, as illustrated in Fig. 5.

In the initialization step (Step 1), the analyst labels selected data instances of the target task if inadequate labeled data instances exist in the target task. In terms of text classification, the analyst searches related documents from the documents (data instances) in the target task by using a word-based retrieval interface, and manually labels the searched documents. This process is unnecessary if the labeled data in the target domain are adequate.

Next, the analyst explores the data instances of the source tasks and target task within a classification transfer view described below. The main goal is to identify and select a group of data instances of the source tasks suitable for the target task. However, selecting the data instances with an automatic method is non-trivial because the relations among data instances of the source tasks and the target task are complicated and can hardly lead to a deterministic criteria for the selection. Our solution is to compute a confidence for each data instance in the source tasks and target task with an automatic algorithm (Step 2), as shown in Algorithm 1.

Fig. 5. The process of visual transfer and classification.

Algorithm 1 Computing domain confidence of instances.

Initialization: Classifier ϕ, and an empty set R.
for each instance s in source domain **do**
  Create training data pair (s, 0), and add it to R
end for
for each instance t in target domain **do**
  Create training data pair (t, 1), and add it to R
end for
Train classifier ϕ with training set R
for each instance i in source domain and in target domain **do**
  Get classification probability of i predicted by ϕ, and set it as the domain confidence of i
end for


In this algorithm, we compute the domain confidence of instances that indicates the probability of a data instance from the source domain or target domain. Instances with confidence less than 0.5 are considered more similar in instances from the source domain than in the target domain, and instances with confidence greater than 0.5 are likely considered the instances from the target domain. Notably, not all data instances from source domain have confidence levels near 0. In particular, the source domain instances with confidence levels near or higher than 0.5 require further investigation because of their higher similarity to the target domain data. These data instances may be used to train target classifier with less precision loss caused by the difference between the source domain and the target domain, namely, "transferred", from the source domain into the target domain. In our exploration process, domain confidence is a significant indicator to decide the data instances from the source domain to be transferred. Thereafter, the analyst interactively explores the confidence and label of each data instance in the source tasks and target task within a visual interface. The data instances of the source tasks with domain confidence levels near 0.5 or above are identified and selected (Step 3). They are added into the training dataset of the target task. Subsequently, the classification model of the target task is updated and used to classify the data instances in the target task. The accomplished target task is added into the task list for the next iteration.

5.2. Visualization and interaction of data instances

The classifier transfer view: In the classifier transfer view (Fig. 6), all data instances from the source tasks and the target task are represented as points and placed on a 2D view. Inspired by [23,24], the view is vertically separated into two panels, where the top one in green shows instances labeled as positive, and the bottom one represents their counterparts. Unlike the two works mentioned above, the data instances come from two different domains in which visual clutter may occur if
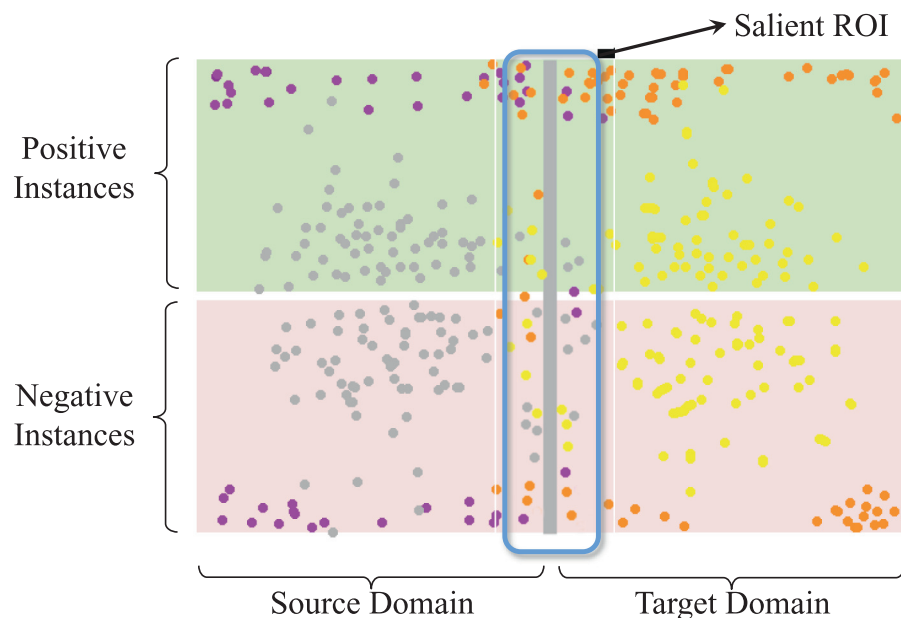
Fig. 6. Illustration of the task transfer view. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 2
Visual encoding for projected data instances in the task transfer view.

Visual encoding	Data instance type
	Labeled data in the source domain
	Unlabeled data in the source domain
	Labeled data in the target domain
	Unlabeled data in the target domain

all the instances are mixed together. Thus, each view is further divided into two half-planes by a vertical boundary to show the source data and target data, respectively. In this way, the entire view is separated into four quadrants. The position of a point indicates its data domain and classification result.

Specifically, the positions of data instances in the panels encode two types of probabilities. The coordinate along the y axis represents the classification probability *pclass* of the associated domain classifier. The closer the probability to 0.5, it is closer to the panel boundary (e.g., another class). Meanwhile, the coordinate on the x axis encodes the domain confidence *cdomain* (marked in Fig. 6), which represents domain confidence value from 0 on the left-most side of the entire view to 1 on the rightmost side. Hence, those instances with domain confidence near 0.5 are projected near the vertical domain boundary in this 2D view. Given the existing similarity between the source and target data domains, a piece of data may locate near the domain boundary, and even cross the boundary, meaning that they may belong to another data domain.

The classifier transfer view represents the confidences of the data instances with respect to the associated domain classifier and data domain. As mentioned in [23], the data instances near the class boundary have high class uncertainty and are necessary for the analyst to perform further investigation and labeling (Table 2). Similarly, in the classifier transfer view, the data instances near the domain boundary are of great importance for the classification process because of the high uncertainty of the domain to which they belong. This region is considered the Salient ROI (blue box in Fig. 6) of transferable instance candidates.

Data exploration view: To allow the analyst to further explore data instances in both source and target domains, Data exploration view (Fig. 2(d)) is provided in our system for the analyst to view details of instances. On the left part of this view, documents from source and target domains are listed in a table. When selecting data instances in the classifier transfer view or in the table, the content of the corresponding document will be displayed on the right side.

User interactions: On the classifier transfer view, the analyst can view the points to obtain classification probability and domain confidence by its coordinates. The text field on the right side displays document content when selecting a point on the classifier transfer view or in the document table. Furthermore, the analyst can select a group of points and mark them to add to the target dataset of the target task.

6. Case studies

Two case studies were conducted with TransXplorer, namely, the Amazon product reviews dataset and the 20 newsgroups dataset, which are widely used in studying transfer learning.

6.1. Amazon product reviews

The Amazon product review dataset [7] contains product reviews collected from Amazon.com from many different product types. Reviews from different product types may vary from each other, whereas they simultaneously share some common features on word distribution. In each product type, several thousand reviews are collected and used in the experiment.

In this case, each product type is considered a specific domain of reviews and has different word distribution. Our goal is to perform binary sentiment classification (i.e., to predict if a review is positive or negative) on each domain to make the classification model fit best its corresponding domain. These domain-classifier pairs are combined to form tasks listed in Table 3, where the tasks in the table are considered as existed well-trained tasks with plenty of labeled reviews without task, "video", which is set to a target candidate with 50 training data instances in each sentiment class and without classification model. To test the performance of the model in the target task, 100 positive documents and 100 negative ones are separated as a test dataset from the task "video".

Data pre-processing: In this dataset, each review consists of several attributes, including reviewer ID, a product name, reviewer location, a 0-to-5-stars rating, review title, and text. For the purpose of binary sentiment classification, we only consider the review title, text, and rating. To transform the multiple-value rating system into bipolar ratings, reviews with ratings greater/less than 3 are regarded as positive and negative. The ones with 3 are discarded because of the ambiguities.

Table 3
Product types (i.e. domains of reviews) in the dataset. The product type "dvd" is set to be the target task.

Tasks
apparel, automotive, baby, beauty, books camera_and_photo, cell_phones_and_service computer_and_video_games, dvd, electronics gourmet_food, grocery, health_and_personal_care jewelry_and_watches, kitchen_and_housewares, magazines music, musical_instruments, office_products, outdoor_living software, sports_and_outdoors, tools_and_hardware toys_and_games, **video**

Fig. 7. Task exploration view of the Amazon product review dataset ($m = 0$, $a = 1$).

Task relevance exploration: First, tasks listed in Source List are imported into the TransXplorer. The corresponding task glyphs are generated based on the properties of each task. In the task relation view, glyphs are initially arranged with $a = 1$, which means only word frequency similarity between domains are used. Then the analyst slightly changes the weight of task relevance and raise the number of transfer instances (namely, m in Eq. (1)) to see the relevance change when transferability is involved (Figs. 7–9). For those task glyphs close to each other in the layout, the analyst selects them to display their details in the Task Property View, and modifies the stepping variate (the parameter q in Eq. (1)) to view the performance variation.

Specifying source tasks: After the exploration of source tasks, a new task ("video") with few training data and a classification model is added as a target task. To correctly characterize task relevance, the analyst sets the relevance weight a to nearly 1 to avoid inaccurate task transferability between source tasks and the new task. From the relevance biased to the domain similarity, the source task ("dvd") closest to the new task in the task exploration view is selected as a candidate source task. Hence, the classifier transfer view will be activated to investigate instance-level transfer between them.

Data instance transfer: Few training instances exist in "video". Thus, a group of data instances is labeled. Then, the source task "dvd" and the target task "video" are studied. After computing the confidence of each data instance, the involved data instances are shown in the classifier transfer view. The analyst investigates them, and decides whether they can be reused by the target classifier. The analyst can find that reviews with large domain uncertainty always include some comments

Fig. 8. Task exploration view of the Amazon product review dataset ($m = 15$, $a = 0.78$).

without domain-specific words (Fig. 10). For example, the two labeled documents from the source domain near the domain boundary with circle marks in Fig. 10 are listed below. Note that the underlined words are common words for expressing sentiment but not the ones, which are only applicable in specific domains.

- Positive one: *I caught a glimpse of this movie years ago and I could not remember the name but I did know who stared in it...so when I went online to get the name of this I was* **pleased** *to see that the sender had it at a very* **good** *price...***Thanks** *so much for having the movie and second getting it to me so quickly!!!* **Thanks** *again!!*
- Negative one: **Waste** *of money.* **Waste** *of time. Well-known cast. Amateurish direction. Plotless. Politically correct to the point of being stupid. The film's incessant attempts at humor* **fail***. Eric Roberts' character laughs-off his grave circumstance throughout most of the film. He is self-important, Godless and emotionally* **false***. It's difficult to empathize with such an unrealistic character. The best performances in this film were given by Gregory Harrison and Lee Grant. The best scene is Gregory Harrison and Eric Roberts' passionate farewell kiss. Most authentic moment in the film: Lee Grant's reaction to her dying son being carried off. Very difficult to watch through in one sitting because it's so repetitive and vapid. You will wait and wait for something to happen and* **nothing** *ever does until the final moments. If you choose to* **waste** *your time with this film, you'll be* **sorry** *you did.*

These reviews can be tagged as labeled training data in "video." When many adequate training reviews from "dvd" are selected and used, the training result of "video" can be committed, and it will be added to the task list as a well-trained task.

Quantitative analysis: To compare with the non-transfer method, a naive Bayes classifier is trained on the original labeled data in "video" and tested on the test dataset. Table 4 shows the test accuracy of the method without transfer and our method of five iterative analysis loops.

6.2. 20 newsgroups dataset

The 20 newsgroups dataset contains approximately 20,000 newsgroups documents, which are partitioned almost evenly into 20 different newsgroups [42] with hierarchical topic levels.
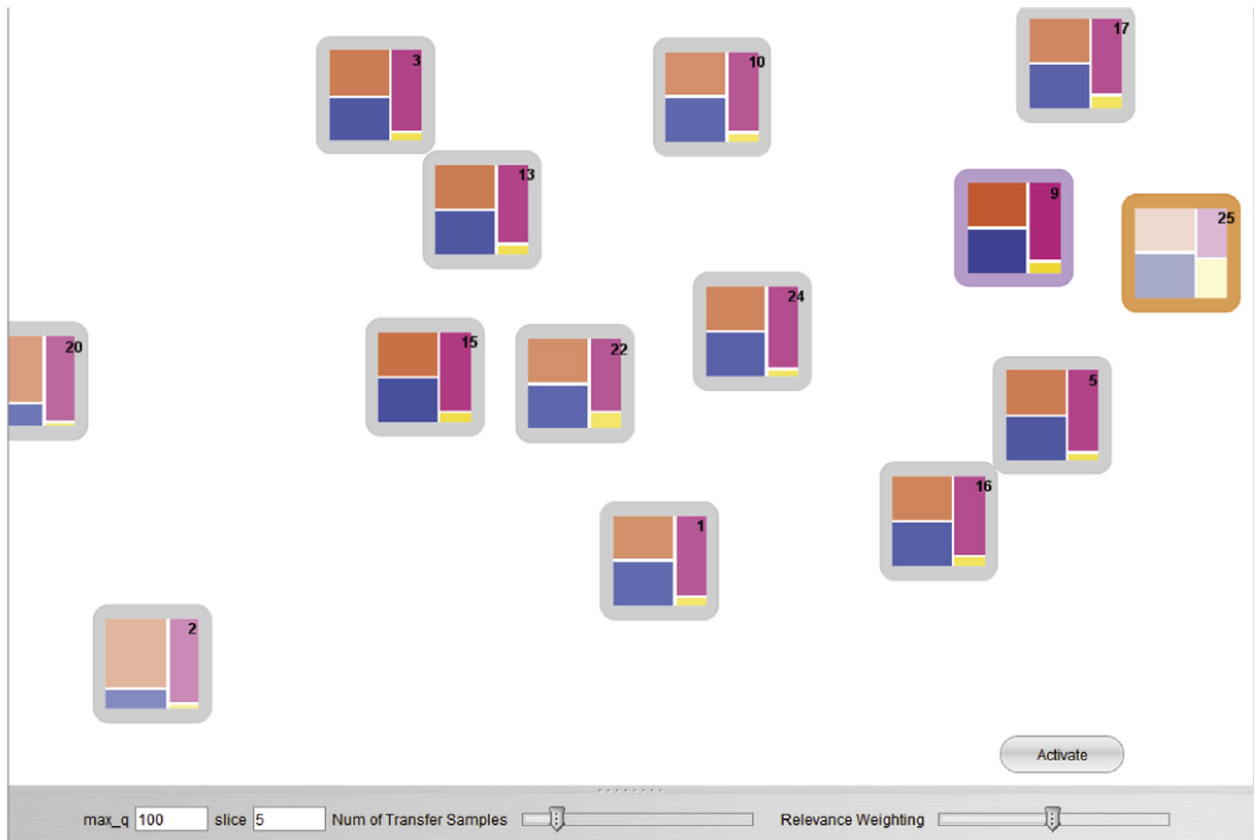
Fig. 9. Task exploration view of the Amazon product review dataset ($m = 15$, $a = 0.5$).
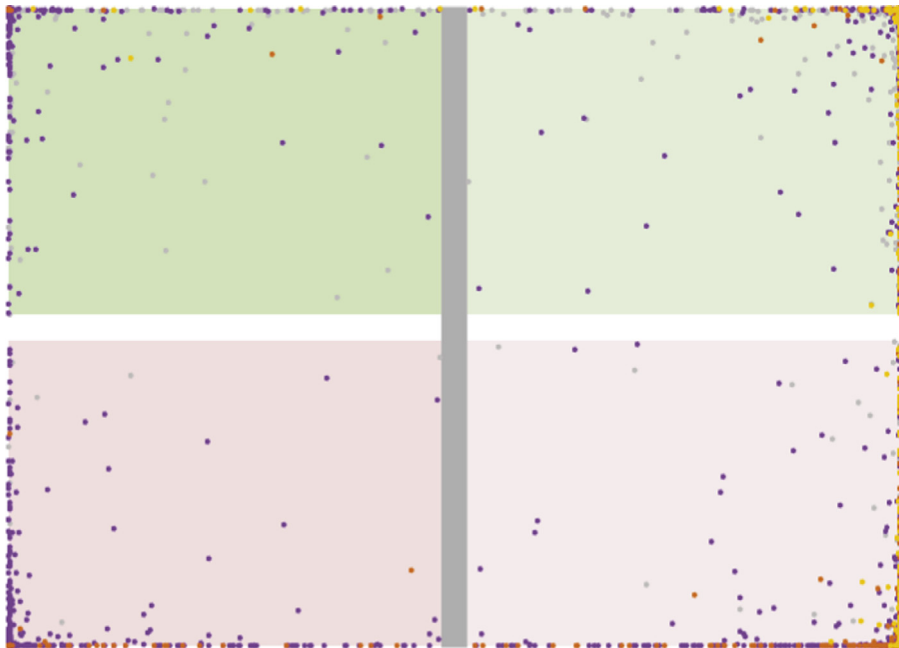

Fig. 10. The classifier transfer view of the Amazon product review dataset. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 4
Prediction accuracy on the test dataset with two methods.

Method	Test accuracy
Naive Bayes	62.09%
Our method	63.47%, 67.82%, 69.01%, 71.13%, 72.51%

Table 5
Sub-categories selected from the 20 newsgroups dataset.

comp.graphics	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.os.ms-windows.misc	rec.sport.baseball
comp.sys.mac.hardware	rec.sport.hockey
comp.windows.x	

Table 6
Prediction accuracy on the test dataset with two methods.

Method	Test accuracy
Naive Bayes	89.04%
Our method (5 iterative selection loops)	90.18%, 90.68%, 93.32%, 95.71%, 96.35%

Data pre-processing: To generate the classification scenario, we select the news from top level topics "comp" and "rec" that contain five and four sub-categories, respectively (listed in Table 5), and combine sub-categories from two top topics to create 20 different domains. The binary classification task is set to decide to which top topic an unlabeled document belongs. The documents from two top topics are from different sub-categories in each domain. Thus, the word distributions of the documents in the same topic from different domains may vary from each other, which are compatible with our problem settings. Furthermore, we choose the domain of "comp.windows.x" + "rec.sport.hockey" as the target domain, and a small test set is reserved from this domain as performed in Case 1.

Task relevance exploration: The task glyph arrangement is shown in Fig. 11 after importing the task list and the target task. In this view, the analyst finds that the classifiers trained in Tasks 5 (comp.windows.x + rec.autos), 6 (comp.graphics + rec.motorcycles), 7 (comp.sys.ibm.pc.hardware + rec.motorcycles), and 8 (comp.os.ms-windows.misc + rec.motorcycles) are more saturated than other tasks, indicating the low performance of the trained classifiers in these tasks. The analyst then discovers that the glyph of Task 3 (comp.os.ms-windows.misc + rec.autos) is the nearest one to the target task glyph (No. 20, comp.windows.x + rec.sport.hockey), and selected as the source task.

Data instance transfer: In the classifier transfer view (shown in Fig. 12), the analyst finds that some labeled data instances in the source domain are located in the panel of the target domain. These data instances can be selected and further investigated by the analyst.

Quantitative analysis: We compare our approach with a naive Bayes classifier which is trained on the original labeled data in Task 20 and tested on the test dataset. The prediction accuracy comparison is listed in Table 6.

6.3. Expert review

To evaluate the actual experience of our system, we interviewed a machine learning researcher from university *Z* and obtained the feedback. We first explained to the reviewer our conceptual overview, interface, and analytical process of our system, and then, presented the two case studies. The feedback is summarized as follows.

Interactive visualization and analysis process: The researcher considered that the interface of our system can be a useful tool for the user in the field of machine learning. He commented that "it is an interesting idea to use multiple interactive visualization methods to present the knowledge transfer process. Differing from only displaying the results in a static form, the interactive interface can assist the user in understanding and exploring the data instances on the domain boundary, and the user is able to perform intensive analysis on the transfer candidate instances".

Furthermore, the reviewer appreciated the idea of the four-stage visual analysis process. He specifically pointed out that the integration of task relevance analysis and data instance transfer analysis enhances the applicability of the system for real-world transfer learning tasks.

Improvements: Several valuable comments were proposed to improve our system. A promising feature in our system lies in providing a view that visualizes the data instances by using several high-dimensional visualization methods to show the data distributions and patterns. Applying different visualization methods based on the data type may be useful; such methods include parallel coordinates for real-valued data or text cloud for text data. Another productive comment includes the possible additional flexibility in making the domain similarity measures changeable to fit different data scenarios.
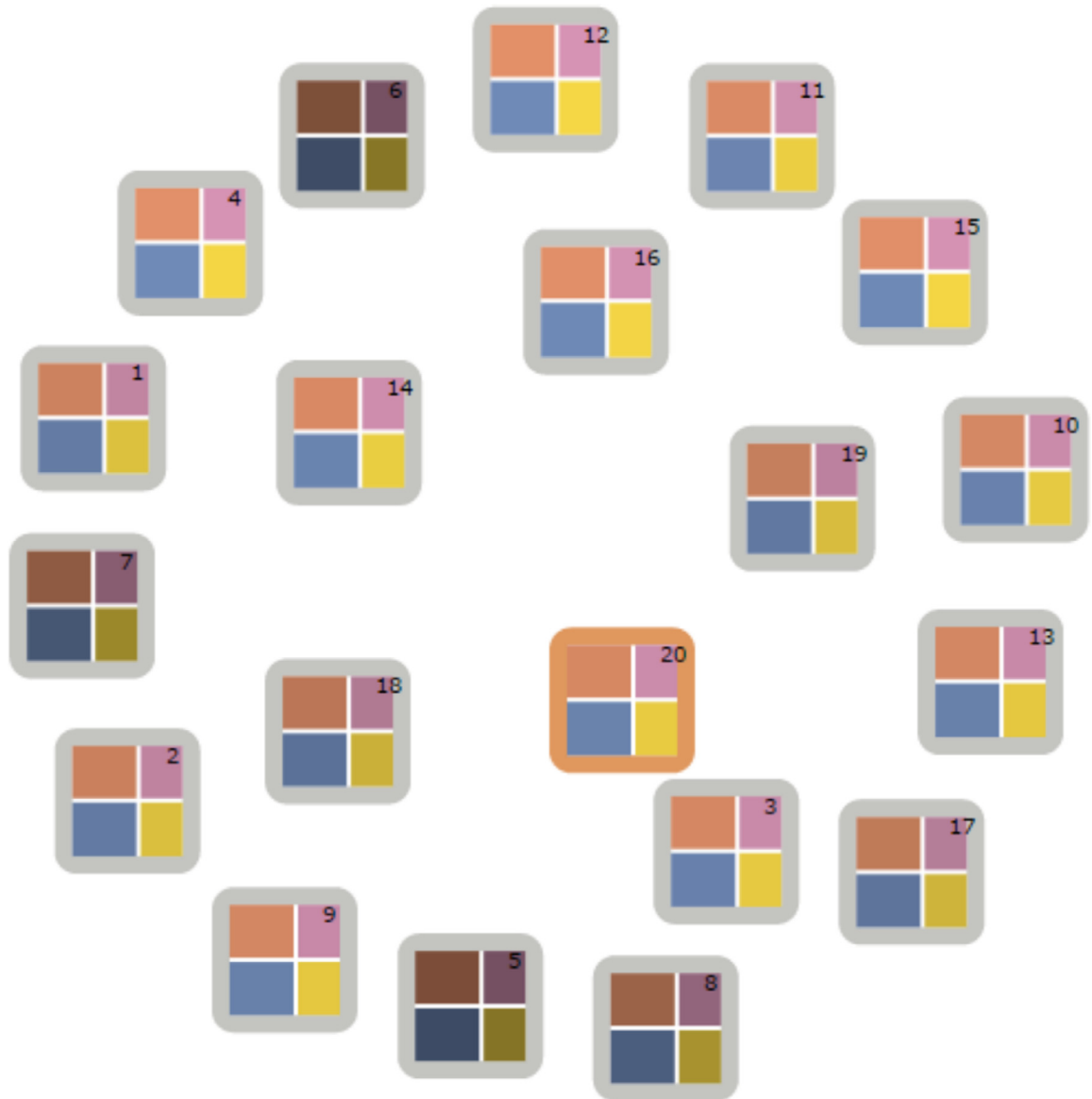
Fig. 11. Task exploration view of the 20 newsgroups dataset ($m = 45$, $a = 1$).

7. Discussion

Case study: The case study demonstrates the applicability of our system in visualizing the task relevance and data instances. Our current visualization technique depicts the relations of tasks with the force-directed layout. The layout of task nodes may be slightly changed because of the instability of the iterative force-directed graph layout algorithm. The results demonstrate that it does not influence recognition of similar tasks because we use the distance to identify the task relevance.

Limitations: Our approach assumes that the dataset is properly pre-processed. In the scenario of document classification, the document should be cleaned and vectorized, and then, imported into our system. Moreover, for the raw dataset, a proper preliminary data exploration process may be useful for the analyst to understand the characteristics of the dataset before the visual transfer process.

Future work: Our approach is designed for binary classification. In the future, we will extend the visual transfer process into multi-class classification, and support comparison of data instances from multiple classes. Another extension of our work is to test additional data types or tasks, such as a sequence of historical documents or classification of images.
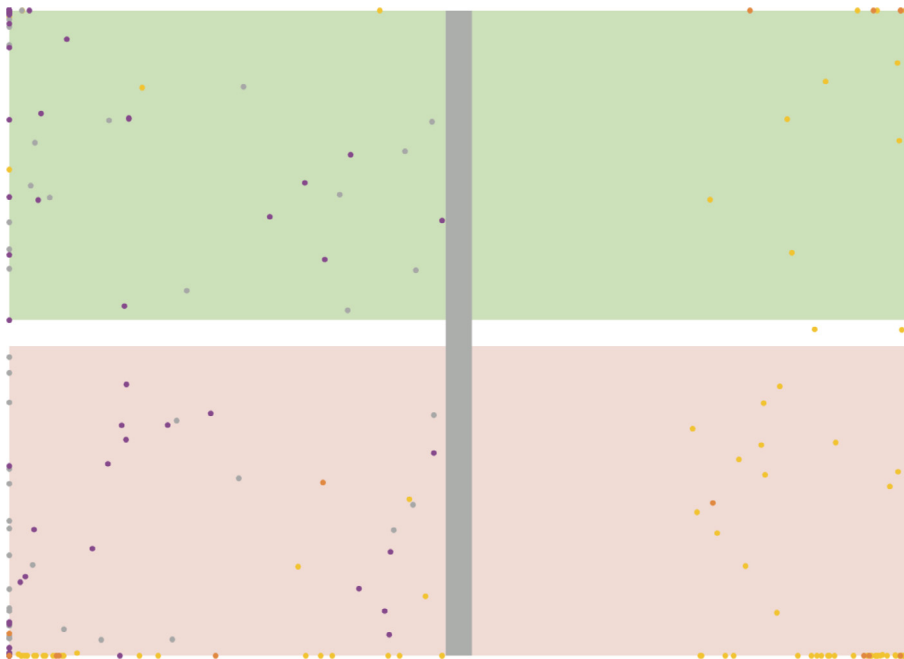
Fig. 12. The classifier transfer view of the 20 newsgroups dataset.

8. Conclusion

This paper presents a suite of visualization techniques and a novel visual-assisted transfer learning methodology for performing binary classification. The key idea is to first assign a source task in the task exploration view, and then, select useful data instances from the source task by using the classifier transfer view. Experimental results on two real datasets demonstrate the validity and efficiency of our approach.

Acknowledgments

References

[1] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, A. Rauber, Visual methods for analyzing probabilistic classification data, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 1703–1712.

[2] S. Amershi, J. Fogarty, D. Weld, Regroup: interactive machine learning for on-demand group creation in social networks, in: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems, 2012, pp. 21–30.

[3] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nani, D. Pedreschi, F. Giannotti, Interactive visual clustering of large collections of trajectories, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2009, pp. 3–10.

[4] M. Ankerst, C. Elsen, M. Ester, H.-P. Kriegel, Visual classification: an interactive approach to decision tree construction, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'99, 1999, pp. 392–396.

[5] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, Adv. Neural Inf. Process. Syst. 19 (2006) 137.

[6] E. Bertini, D. Lalanne, Surveying the complementary role of automatic data analysis and visualization in knowledge discovery, in: Proceedings of ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration, VAKD'09, 2009, pp. 12–20.

[7] J. Blitzer, M. Dredze, F. Pereira, Biographies, Bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of Annual Meeting of the Association For Computational Linguistics, vol. 45, 2007, p. 440.

[8] H. Bosch, D. Thom, F. Heimerl, E. Puttmann, S. Koch, R. Kruger, M. Wörner, T. Ertl, ScatterBlogs2: real-time monitoring of microblog messages through user-guided filtering, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 2022–2031.

[9] E.T. Brown, J. Liu, C.E. Brodley, R. Chang, Dis-function:learning distance functions interactively, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2012, pp. 83–92.

[10] A. Buja, D. Swayne, M. Littman, N. Dean, H. Hofmann, Interactive data visualization with multidimensional scaling, J. Comput. Graph. Stat. 17 (2) (2008) 444–472.

[11] N. Cao, D. Gotz, J. Sun, H. Qu, DICON: interactive visual analysis of multidimensional clusters, IEEE Trans. Vis. Comput. Graph. 17 (12) (2011) 2581–2590.

[12] J. Choo, H. Lee, J. Kihm, H. Park, iVisClassifier: an interactive visual analytics system for classification based on supervised dimension reduction, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2010, pp. 27–34.

[13] Q. Cui, M.O. Ward, E.A. Rundensteiner, J. Yang, Measuring data abstraction quality in multiresolution visualization, IEEE Trans. Vis. Comput. Graph. 12 (5) (2006) 709–716.

[14] W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Co-clustering based classification for out-of-domain documents, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 210–219.

[15] E. Eaton, M. desJardins, T. Lane, Modeling transfer relationships between learning tasks for improved inductive transfer, in: Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases, Springer-Verlag, 2008.

[16] E. Eaton, G. Holness, D. McFarlane, Interactive learning using manifold geometry, in: Proceedings of AAAI Fall Symposium on Manifold Learning and its Applications, 2009, pp. 10–17.

[17] S. van den Elzen, J.J. Wijk, BaobabView: Interactive construction and analysis of decision trees, in: Proceedings of IEEE VAST'11, 2011, pp. 151–160.

[18] A. Endert, C. Han, D. Maiti, L. House, S. Leman, C. North, Observation-level interaction with statistical models for visual analytics, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2011, pp. 121–130.

[19] J. Fogarty, D. Tan, A. Kapoor, S. Winder, CueFlik: interactive concept learning in image search, in: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems, 2008, pp. 29–38.

[20] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw.: Pract. Exp. 21 (11) (1991) 1129–1164.

[21] D. Gentner, K.J. Holyoak, Reasoning and learning by analogy: introduction., Am. Psychol. 52 (1) (1997) 32–34.

[22] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2000.

[23] F. Heimerl, S. Koch, H. Bosch, T. Ertl, Visual classifier training for text document retrieval, IEEE Trans. Vis. Comput. Graph. 18 (12) (2012) 2839–2848.

[24] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, G. Heidemann, Inter-active learning of ad-hoc classifiers for video visual analytics, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2012, pp. 23–32.

[25] M.S. Hossain, P.K.R. Ojili, C. Grimm, R. Muller, L.T. Watson, N. Ramakrishnan, Scatter/gather clustering: flexibly incorporating user feedback to steer clustering results, IEEE Trans. Vis. Comput. Graph. 18 (12) (2012) 2829–2838.

[26] M. Jankowska, V. Keselj, E. Milios, Relative n-gram signatures: document visualization at the level of character n-grams, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2012, pp. 103–112.

[27] D.H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, R. Chang, iPCA: an interactive system for PCA-based visual analytics, Comput. Graph. Forum 28 (3) (2009) 767–774.

[28] H. Jiawei, M. Kamber, Data Mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann, San Francisco, CA, 2011.

[29] B. Johnson, B. Shneiderman, Tree-maps: a space-filling approach to the visualization of hierarchical information structures, in: Proceedings of IEEE Conference on Visualization, Visualization'91, IEEE, 1991, pp. 284–291.

[30] D. Kifer, S. Ben-David, J. Gehrke, Detecting change in data streams, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases, Vol. 30, VLDB Endowment, Inc., 2004, pp. 180–191.

[31] Y. Liu, G. Salvendy, Design and evaluation of visualization support to facilitate decision trees classification, Int. J. Humam-Comput. Study 65 (2) (2007) 95–110.

[32] T. May, A. Bannach, J. Davey, T. Ruppert, Guiding feature subset selection with an interactive visualization, in: Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2011, pp. 111–120.

[33] S. Mika, G. Rätech, J. Weston, B. Schölkopf, K.-R. Müller, Fisher discriminant analysis with kernels, in: Proceedings of IEEE Signal Processing Society Workshop, 1999, pp. 41–48.

[34] J. Moehrmann, G. Heidemann, Efficient annotation of image data sets for computer vision applications, in: Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications, VIGTA'12, ACM, New York, NY, USA, 2012, pp. 2:1–2:6.

[35] T. Muhlbacher, H. Piringer, A partition-based framework for building and validating regression models, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 1962–1971.

[36] T. Muhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, M. Streit, Opening the black box: strategies for increased user involvement in existing algorithm implementations (2014).

[37] E. Müller, I. Assent, R. Krieger, T. Jansen, T. Seidl, Morpheus: interactive exploration of subspace clustering, in: Proceedings of ACM SIGKDD, ACM, New York, NY, USA, 2008, pp. 1089–1092.

[38] E.J. Nam, Y. Han, K. Mueller, A. Zelenyuk, D. Imre, ClusterSculptor: a visual analytics tool for high-dimensional data, in: Proceedings of IEEE Symposium on Visual Analytics Science and Technology, 2007, pp. 75–82.

[39] J.G.S. Paiva, L. Florian-Cruz, H. Pedrini, G.P. Telles, R. Minghim, Improved similarity trees and their application to visual data classification, IEEE Trans. Vis. Comput. Graph. 17 (12) (2011) 2459–2468.

[40] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[41] C.H. Papadimitriou, H. Tamaki, P. Raghavan, S. Vempala, Latent semantic indexing: a probabilistic analysis, in: Proceedings of ACM Symposium on Principles of Database Systems, 1998, pp. 159–168.

[42] J. Rennie, 20 newsgroups dataset. http://qwone.com/~jason/20Newsgroups/, 2008, access date Feb. 26, 2016.

[43] T. Schultz, G.L. Kindlmann, Open-box spectral clustering: applications to medical image analysis, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 2100–2108.

[44] C. Seifert, M. Granitzer, User-based active learning, in: Proceedings of IEEE International Conference on Data Mining Workshops, 2010, pp. 418–425.

[45] B. Settles, Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 1467–1478.

[46] C.A. Steed, C.T. Symons, F.A. Denap, T.E. Potok, Guided text analysis using adaptive visual analytics, in: Proceedings of International Conference on Visualization and Data Analysis, 2012.

[47] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.

[48] J. Talbot, B. Lee, A. Kapoor, D.S. Tan, EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers, in: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems, 2009, pp. 1283–1292.

[49] J. Wei, Z. Shen, N. Sundaresan, K.-L. Ma, Visual cluster exploration of web clickstream data, in: Proceedings of IEEE Visual Analytics Science and Technology.

[50] D. Yang, Z.-X. Xie, E.A. Rundensteiner, M.O. Ward, Managing discoveries in the visual analytics process, SIGKDD Explor. Newsl. 9 (2) (2007) 22–29.