Large-scale Heterogeneous Learning in Big Data Analytics

Jun (Luke) Huan
Professor
Department of Electrical Engineering and Computer Science
University of Kansas
http://ittc.ku.edu/~jhuan/
Heterogeneous Learning

- Heterogeneous learning deals with data from complex real-world applications such as social networks, biological networks, internet of things among others.
- The heterogeneity could be found in multi-task learning, multi-view learning, multi-label learning, multi-instance learning and etc.
Multi-Task Learning: an Case of Heterogeneous Learning

- Multi-task learning is widely utilized
- In many real world applications data have “structures” and the structure could be found at different levels
  - Among samples
  - Among features
  - Among data sources
  - Among learning tasks
Big Data and Heterogeneity

- Big data: many Vs
  - Data size is too large to be stored in memory, which requires linear computational time
  - Problem at hand is under constraint, so there is a need to obtain further information from other sources
  - Q: Is a large number of i.i.d data a blessing for ML, DM, and statistics?

- A few observations that we want to highlight
  - Data are fast accumulation, but not necessarily and in many cases are not i.i.d.
  - Big data may suggest many learning tasks. For each task, relevant data is enormous, task-specific data is sporadic (“your appetite for big hypothesis is big”)

11/5/14
Overview

- Multi-task learning
- Multi-view learning
- Multi-label learning
Overview of Multi-Task Learning

- Introduction of Multitask Learning
- Large scale Multitask Learning algorithms
- Multi-Task Learning with Structured Input and Structured Output
Multitask Learning

- Multitask learning is an inductive transfer mechanism for improving generalization performance [Caruana, Machine Learning’97]
- If number of samples \( (n) \) are large in comparison with dimension \( (d) \), the tasks can be learnt individually
- When the number of samples are small, training information of several related tasks can be shared to improve generalization error
Multitask Learning

There are numerous applications where we need to predict multiple related tasks.

Prediction of Examination Scores for Each School [Argyriou, 2008]

<table>
<thead>
<tr>
<th>Student Age</th>
<th>Previous Scores</th>
<th>...</th>
<th>School Region</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 yrs</td>
<td>83</td>
<td>...</td>
<td>3</td>
<td>...</td>
</tr>
</tbody>
</table>

Task 1
Test Result for School 1

Task 2
Test Result for School 2

???

Robot Inverse Dynamics [Chai, 2008]

Trajectory of Joints

Torques Required

HIV Therapy Screening [Bickel, 2008]

Patient Treatment Records

Predict Drug Combinations
Multitask Learning

- One approach for multitask learning would be to build a separate model for each task
  \[
  \min_{f_t \in S} \frac{1}{T} \sum_{t=1}^{T} E(l(f_t(X_t), y_t))
  \]
  \(l(f_t(X_t), y_t)\) is any loss function between the estimated model predictions, \(f_t(X_t)\), and the given labels \(y_t\). \(X_t\) is the given input features of samples belonging to task \(t\). \(f_t \in \) common set \(S \forall \) tasks \(t \in 1,2,\ldots,T\). \(T\) is total number of tasks.

- If tasks are related, it is useful to share training information between tasks
  \[
  \min_{f_t \in S} \frac{1}{T} \sum_{t=1}^{T} l(f_t(X_t), y_t) + \lambda \Theta(f)
  \]
  Existing MTL algorithms majorly differ by choice of \(\Theta(f)\)
  \(\Theta(f)\) is a positive monotonically increasing regularization function,
  which binds the parameters of individual tasks together.
Multitask Learning

- Assume all tasks are similarly related to each other
  - Task parameters are constraint to lie close to each other
    \[
    \min_{\mathbf{w}} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N_t} l\left(\langle \mathbf{w}_t, \mathbf{X}_{t,i} \rangle, y_t \right) + \sum_{t=1}^{T} \| \mathbf{w}_t - \sum_{s} \mathbf{w}_s \|
    \]
    \(X_t\) and \(y_t\) are input features and output labels of given dataset for each task \(t \in \{1,2,..T\}\). Each task consists of \(N_t\) samples. Assuming linear model, \(\mathbf{w}_t\) is the parameter vector for each task.
  - L2 norm is often used to establish the similar proximity of the parameters
  - Examples: Evgeniou and Pontil, KDD’04, Bakker and Heskes, JMLR’03, Yu et al, ICML’05, Liao and Carin, NIPS’05
Multitask Learning Meets Heterogeneity

Multitask Learning

Without Task Relationship Model

With Feature Selection

Without Feature Selection

With Feature Selection

Known Task Relationships

Hierarchical Structure

Undirected Graphs

Learning Task Relationships

Clustering

Learn PSDM
Multitask Learning for Big Data

- Design of the algorithm needs to be scalable
- Memory constraints need to be addressed
- Three different directions for handling big data
  - Distributed Processing
  - Online MTL algorithms
  - Feature selection and storage techniques
Distributed Processing

- Distributed Processing can be used for any multitask learning algorithms
- Very efficient way to handle big data
- Most algorithms in machine learning are iterative and of non-parallel nature
- Challenge lies in parallelizing these algorithms
- Several efforts are being made to parallelize the optimization problem in multitask learning
  - Examples: Riezler 2013, Ahmed et al WDM’14, Ahmed et al WDM’12
Online Multitask Learning

- **Applications**
  - When data is too large to store in RAM
  - Due to time constraints, at most one pass can be made

- **General Structure**
  - Start with arbitrary parameter
  - For each step
    - Get one instance of each task
    - Evaluate the loss function using predicted and actual output
    - Update parameters for each task \( k \)
  - Most online multitask learning algorithms vary by the choice of updating function
Online MTL algorithms

- Entire data does need not be stored in RAM at the time of training update
- Incremental updates provide a good scalable alternatives
- Using stochastic gradient descent to update the model parameter
- Not as effective as batch processing methods
  - Examples: Dekel et al, Learning Theory’06, Yang et al, ICIKM’10, Yang et al, TKDD’13, Sun et al, KDE’13
Dimensionality Reduction

- Storage of data is a big problem, particularly in multitask learning where number of samples, dimensionality and tasks are big numbers

- Map the data to reduced dimensionality space
  - Spectral Value Decomposition
    - Difficult to perform for large datasets
    - Results in dense matrices
  - Feature Hashing
  - Feature Clustering
Feature Hashing

- Consider text categorization applications such as email spam filtering
  - Feature space consists of words in English language
  - Lot of words mean the same thing (some may be mis-spelt versions)
  - Lot of words are used seldom, but ignoring them is not a good idea as they become targets of spammers
  - Feature hashing solves the problem efficiently
Feature Hashing

- Efficient hashing can help in dimensionality reduction

- Opposite but similar to kernels, where data is projected in higher input space
- Weinberger et al, ICML’09, Zhao and Xing, CVPR’14, Goyal et al, NIPS’11
Feature Clustering

- Driving application is on-line targeted display advertising
  - Determines when, where, to whom to display
- Using model parameters from massive number of tasks to perform hierarchical clustering of features
- Make cuts on the cluster tree and aggregate results from subtrees
- Raeder et al, KDD’13
Multitask Learning with Structured Input and Output
Mutitask Learning with Task Relationships

- Often just using Multi Task Learning is not enough to improve generalization error
  - We find all tasks are not related in same way
  - Training unrelated tasks results in negative transfer
  - Incorporation of the task relationship structure in the learning framework becomes necessary

- Incorporating known Task Relationship Structures
  - Hierarchical Structure
  - Graph Structure

- Learning Task Relationship Structures
  - Task Clustering
  - Learn Positive Semidefinite Matrices
Multitask Learning with Task Relationship Models

Content Based Social Network User Behavior Prediction [Fei, 2011a]

<table>
<thead>
<tr>
<th>Movie Descriptors</th>
<th>User Interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

User Ratings

Tumor Diagnostics [Fei, 2011b]

Micro array gene expression data
Green: Liver specific tumors, Yellow: Kidney specific tumors

Document Classification [Cai, 2004]

Science
Math
Physics

11/5/14
Multitask Learning with Task Relationships

- Known Task Relationships
  - Hierarchical Task Structure
  - A directed Graph Structure
- Learn Task Relationships
  - Task Clusters
  - Learning Positive Semidefinite Matrix (Graphical Structure of Tasks)
MTL with Known Task Structure

- Hierarchical Structure or Directed Acyclic Graphs
  - E.g. document classification, Image classification

- Two major approaches
  - Local Classifiers
  - Global Classifiers
Hierarchical Task Structure

- Local Classifier approach
  - Build local classifier at each node, or each parent node or each level using top-down approach (Silla and Freitas, DMKD’11)
Hierarchical Task Structure

- Consistency of predictions is a problem
  - Additional measures need to be taken
- Global relationships are not taken into account
- Scalable and parallelizable
  - Examples: Silla and Freitas, DMKD’11, Eisner et al, CIBCB’05, Fagni and Sabastiani, LTC’07 etc
Hierarchical Task Structure

- Global Classifier Approach
  - Takes global hierarchical structure into account
  - Often termed as Big Bang approach
  - To make use of hierarchical task relationships, the
    regularization term ensures the proximity of leaf nodes with
    its ancestors.

\[
\min \frac{1}{T} L(W, X, Y) + \sum_{v \in V} f(w_v, w_{A(v)})
\]

Here, \(A(v)\) are the ancestors of any node \(v\).

- Examples: Cai and Hofmann, ICIKM’04, Cesa et al, JMLR’06, Kim
  and Xing, ICML’10, Dekel et al ICML’04, Wang et al, JMLR’11
Large scale hierarchical learning

- Use hierarchical layers to parallelize the algorithm
  - Gopal and Yang, KDD’13 optimize alternate layers simultaneously at each iteration
    \[
    \min \frac{1}{T} L(W, X, Y) + \sum_{v \in V} \left\| w_v - w_{\pi(v)} \right\|^2
    \]
    Here, \(\pi(v)\) are the parents of any node \(v\).
  - Ahmed et al, WDM’14 optimize subtrees together at each iteration
  - Both algorithms reduce the optimization problem to have linear computational complexity
Known Task Structure

- Graph Structure or Undirected Graphs
  - The task structures can often be represented as undirected graphs
    - E.g. Social Networks, Biological Networks etc
  - Graphical structure can be represented as a positive semi-definite matrix in the regularization term
  - The positive semi-definite matrix stores the task relationship information, and enforces the parameters of related tasks close together
  - The problem’s computational complexity can be linear depending on the loss function and optimization methods used

\[
\min_{\mathbf{W}} \frac{1}{T} L(W, X, Y) + \lambda \text{tr}(\mathbf{W} \Omega^{-1} \mathbf{W}^T)
\]

\(\Omega\) is a task covariance matrix

- Example: Chen and Kim, Applied Statistics’10
Learn Task Relationships

- Learn Task Clusters
  - Non-overlapping clusters
    - Aim at keeping the parameters close to the mean of the clusters
      - Example: Jacob et al, NIPS’08
  \[
  \min \frac{1}{T} L(W, X, Y) + \sum_{c=1}^{r} \sum_{j \in J_c} \| w_j - \bar{w}_c \|^2
  \]
  - Overlapping clusters
    - Learn the basis functions for each task where only some components are shared
      - Example: Kumar and Daume III, ICML’2012
Learn Task Relationships

- Learn a Positive Semidefinite Matrix
  - The correlation between tasks can be represented as a positive semidefinite matrix which sometimes needs to be learnt
  - Enforcing sparsification of the correlation matrix promotes the learning of a sparse graph

\[
\min \frac{1}{T} L(W, X, Y) + \lambda tr(W \Omega^{-1}W^T)
\]

\(\Omega\) is a task covariance matrix

- Examples: Fei and Huan, ICDM’11b, Argyriou et al, NIPS’07b, Zhang and Schneider, NIPS’10
Discuss complexities and challenges of implementation

- Learning task relationships is more challenging problem but necessary for improved generalization error
- Need to develop algorithms with at most linear complexities for scalability
Additional Information
Sources

- Structured Input
- Information from multiple views
Structured Input and Output

- Unstructured input: content only, e.g. body of an email, video and audio file.
- Structured data: content + organization

Protein Structure (source: wikipedia)    Web pages (Source: Lampert, NIPS’11)
Learning with Structured I/O

- Normal machine learning: $f : X \rightarrow Y$
  - Input $X$ is a data matrix in vector space.
  - Output $Y$ is a real number or categorical number
    - Classification, regression, ranking, density estimation,…

- Structured data learning: $f : X_s \rightarrow Y_s$
  - Input $X_s$ could be any complex structured objects.
    - Trees, sequences, graphs, images,…
  - Output $Y_s$ could be any complex structured objects
    - Sequences, graphs,…
Related Work

Image segmentation [Plath et al, ICML’09]; Background subtraction [Huang and Zhang, ICML’09]

Gene selection [Rapaport et al, BMC08]
Genome-Phenome association [Kim and Xing, ICML’10]

Text categorization [Sun, KDD’10]

Information flow [Kleinberg, SIGMOD’10]
Community outlier discovery [Gao, KDD’10]
Multi-task Learning

- Learn multiple related tasks simultaneously
- Improve the generalization performance
  - Leverage commonality among tasks
  - Especially helpful when training samples are very limited for each task

**T1**: prostate (PR) cancer prediction

**T2**: kidney (KI) cancer prediction

**T3**: bladder (BL) cancer prediction

**Joint Learning**

- PR cancer or not
- KI cancer or not
- BL cancer or not

*Figure: An MTL example from multiple cancer prediction.*
Examples of MTL with SISO

- Multiple cancer prediction from Microarray data
  - SI: Biological pathways formed by genes
  - SO: relationship among different cancers.

- Text categorization
  - SI: “wordnet” of key word features or hyper link between web pages.
  - SO: text label hierarchy.

- Collaborative filtering
  - SI: Item-item relationship
  - SO: user-user relationship
fMRI: an MTL example with SISO

- **fMRI**: Functional magnetic resonance imaging
- **Input**
  - A word/picture describing an object.
  - Features are words describing the object.
- **Output**
  - Each voxel activity for the stimulus word/picture.

Figure: An MTL application from [Mitchell08, Liu10]. Each “task” is to predict the activity of a voxel.
Objective and Hypothesis

- Objective: design an MTL model that handles structured input and structured output.
- Hypothesis: the interplay between SI and SO enables us to build more accurate MTL models.
  - Informative features guide more accurate task relationship inference.
  - Accurate task relationship benefits informative feature selection.

MTL with SISO

Smoothness across features:
\[ \sum_{i,j} g_{ij} \| \tilde{w}_i - \tilde{w}_j \|^2 = \text{tr}(W^T L W) \]

Smoothness across tasks:
\[ \text{tr}(W \Omega^{-1} W^T) \]

[Zhang and Yeung, UAI’09]
MTL Feature Selection with SISO

- Traditional way: block sparse regularization, e.g. group lasso
- A more relaxed framework with $L_1$ penalty
  - Each task can select its own feature sets.
  - Imposed “smooth effect” based on the structured input graph.

Feature graph $G$
Objective function and Optimization

- **Objective**
  \[
  \min_{W, \Omega} \sum_{i=1}^{k} \sum_{j=1}^{n} l(y^i_j, x^T_i w_i) + \lambda_1 \| W \|_1 + \frac{\lambda_2}{2} \text{tr}(W^T LW) + \frac{\lambda_3}{2} \text{tr}(W\Omega^{-1}W^T)
  \]
  s.t. \( \Omega f = 0 \), \( \text{tr}(\Omega) = k \)

- **Optimization**
  - The objective is jointly convex.
  - Alternatively optimize two variables.
    - Given \( W \): Cauchy-Schwarz Inequality
    - Given \( \Omega \): accelerated gradient descent + gradient projection
Experiment

- Real world Datasets

- Microarray data [Nutt et al, Cancer Research’03, Singh et al, Cancer Cell’02, Su et al, Cancer Research’01]
  - 8 cancer types corresponding to 8 tasks.
  - Table 1: Microarray data sets for 8 tasks. #S: Total number of samples; #P: Number of positive samples; #N: number of negative samples.

<table>
<thead>
<tr>
<th>Data</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_5$</th>
<th>$T_6$</th>
<th>$T_7$</th>
<th>$T_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#S</td>
<td>54</td>
<td>22</td>
<td>54</td>
<td>14</td>
<td>16</td>
<td>21</td>
<td>29</td>
<td>102</td>
</tr>
<tr>
<td>#P</td>
<td>27</td>
<td>11</td>
<td>27</td>
<td>7</td>
<td>8</td>
<td>14</td>
<td>14</td>
<td>50</td>
</tr>
<tr>
<td>#N</td>
<td>27</td>
<td>11</td>
<td>27</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>52</td>
</tr>
</tbody>
</table>

- fMRI data [Mitchell et al, Science’08]
  - Neuroimages of voxel activity for 60 words from 9 subjects.
  - 500 tasks (voxels) and 5000 co-occurrence word features.
Modeling accuracy

Figure: prediction accuracy comparison. Left: Microarray data; Right: fMRI data. MTLapTR: our method; MTLasso: [Liu et al, ICML’09]; MTLPTR: [Zhang and Yeung, NIPS’10]; MTLTR: [Zhang and Yeung, UAI’09].
Task relationship inference study

MTLapTR (Our Method)  MTLPTR [Zhang10]  MTLTR [Zhang09]

Figure: Task Relationship embedding in 3D space from Ndaona.
T1: Breast cancer; T2: kidney; T3: ovary cancer; T4: Liver cancer; T5: bladder; T6: Brain Classific GBM; T7: Brain Classic AO; T8: Prostate tumor.
Applying MTL to Social Network Data Analysis

- User behaviors
  - Forward, like or comment on a message
  - Click an advertisement
  - Attitude conversion

- Applications
  - Smart advertising
  - Personalized news delivery
  - Enhanced search

Fig.1: A tiny snapshot from facebook.com.
Objective and Challenges

- **Goal**: to predict users’ behavior towards the information content, *e.g.*, news, ad., webpage.

- **Challenges**
  - User behavior data in SN is heterogeneous.
    - Heterogeneous topics, *e.g.*, science, travelling, entertainment…
    - Heterogeneous contents, *e.g.*, video, text, image.
  - The data is sparse and imbalanced.
    - In 2009, on average only 143.5 actions/user on 151,000 news stories in digg.com.
  - The data is in large scale.
    - Twitter: ~200 million users.
    - Facebook: 750 million active users.
Why is MTL promising?

- Multi-task learning
  - Leverage performance via learning multiple tasks simultaneously.
  - Handle heterogeneous multi-tasks.
  - Empirically perform well for the case of limited training samples per task.
  - Potentially suitable for large scale learning with the use of cloud computing.
An Case Study: Information Flow in Social Network

- Given a “seed” user and its followers together with their historical activity profile, design a MTL framework that jointly models the follower’s behavior towards the seed’s posting.
- We use linear function to model each follower’s behavior (task) and derive the task relationship from users’ historical data.
Methodology Overview

- Data Collection and feature extraction.
- MTL construction
  - Treat each follower as a task \( f_i(x) = w_i^T x \)
  - Task relationship derivation
  - Task relationship incorporation
- Optimization
user Internet

\[
X = \begin{bmatrix}
0.67 & 0.83 \\
0.62 & 0.35 \\
0.68 & 0.53 \\
0.49 & 0.45 \\
0.68 & 0.51 \\
\end{bmatrix}
\]

Feature (idf-tf)

Extraction

\[
y_1 = \begin{bmatrix}
-1 \\
1 \\
-1 \\
1 \\
-1 \\
? \\
\end{bmatrix}
\]

\[
y_2 = \begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
5 \\
6 \\
? \\
\end{bmatrix}
\]

\[
y_3 = \begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
5 \\
6 \\
? \\
\end{bmatrix}
\]

Fig2. Data representation and feature extraction

<table>
<thead>
<tr>
<th>#</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tech. : Google maps, Android</td>
</tr>
<tr>
<td>2</td>
<td>Tech. : Apple TV, iOS…</td>
</tr>
<tr>
<td>3</td>
<td>Tech. : Samsung galaxy …</td>
</tr>
<tr>
<td>4</td>
<td>Tech. : iPhone, SIM card…</td>
</tr>
<tr>
<td>5</td>
<td>Tech. : Verizon, iPhone 4 …</td>
</tr>
<tr>
<td>6</td>
<td>Tech. : apple iOS 5.0 …</td>
</tr>
</tbody>
</table>
Content based user similarity

- With the heterogeneity of social networks, we cannot rely on network topology to group tasks.
- User interest w.r.t. the information content plays a major role in social information diffusion.
- Calculate the similarity between user $i$ and $j$ for the $l$th category from user profiles as

$$a_{ij}^{(l)} = \frac{\langle P_{:,l}^{(i)}, P_{:,l}^{(j)} \rangle}{\|P_{:,l}^{(i)}\| \|P_{:,l}^{(j)}\|}$$

#activity vector for user $j$ on the $l$th category, e.g. voting, posting et al. from the user profile matrix
Let $A^{(l)} = \{a^{(l)}_{i,j}\}_{i,j=1}^{n}$ we can view $A^{(l)}$ as an weight matrix for a weighted graph $G^{(l)}$ capturing the structure of users for category $l$.

With $t$ categories, a multi-graph $G = \{G^{(l)}\}_{l=1}^{t}$ is built.

Users that share the same interests are expected to have similar prediction models when seeing the contents from their favorite information categories (e.g. Technology, Entertainment, Science).
Heterogeneous Task Relationship Incorporation cont.

- Adding a $L_2$ smooth regularization term

$$\sum_{l=1}^{n} \sum_{i,j=1}^{k} I_{i,j}^T \cdot \alpha_{ij} \| w_i - w_j \|_2^2 = \sum_{i=1}^{t} r_i tr(WL^{(i)}W^T)$$

where \(n\): #samples, \(k\): #users, \(t\): #categories, \(I\): \(t\) by \(n\) indicator matrix with \(I_{jk} = 1\) if the \(k\)th posts belongs to the \(j\)th category and 0 otherwise. \(r_i\): \(r_i = \sum_{j} I_{ij}, (1 \leq i \leq t)\) summarize the number of posts belonging to the \(i\)th category.
Objective

Where $B$ is the weight matrix based on positive and negative sample ratio for each task to guarantee that the misclassification cost is more on rare samples.
Optimization

- The objective is convex.

- First order gradient method
  - Accelerated gradient descent [Nesterov, 2003]
  - Projected gradient [Boyd, Cambridge Univ Press’04]

- Convergence rate: $O(1/\sqrt{\varepsilon})$, where $\varepsilon$ is the desired accuracy.

- Refer to our paper for details.
Experiment

- Data Sets: 4 data sets collected from digg.com.

Table 1: Data set: the symbol of the data set. \( \#T \): total number of tasks (followers), \( \#S \): total number of samples (stories), \( \#F \): total number of features, \( \#C \): total number of categories.

<table>
<thead>
<tr>
<th>Seed</th>
<th>username</th>
<th>#T</th>
<th>#S</th>
<th>#F</th>
<th>#C</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 )</td>
<td>nichewp</td>
<td>11</td>
<td>71</td>
<td>942</td>
<td>3</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>buhlerchelsey</td>
<td>15</td>
<td>61</td>
<td>3217</td>
<td>5</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>GIVINGAWAY</td>
<td>12</td>
<td>57</td>
<td>1167</td>
<td>2</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>arjunchauhan24</td>
<td>10</td>
<td>41</td>
<td>1416</td>
<td>5</td>
</tr>
</tbody>
</table>
Model Construction and Evaluation

- **Model Construction**
  - 5-fold cross validation, 5 fold CV on training data to select parameters
  - Grid search on a wide range of parameters.

- **Model Evaluation**
  - Compare with SVM-RFE, MTLF [Liu et al, UAI’09], MTLALap (network topology task relationship).
  - Collect precision, recall and F1 score:

\[
\begin{align*}
\text{precision} &= \frac{TP}{TP + FP} \\
\text{recall} &= \frac{TP}{TP + FN} \\
F_1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\end{align*}
\]
Results

Figure 3: Average F1 score for 4 seed users.
Relationship of MTL with other algorithms

- Multi-class
  - Problem is often framed as multiple binary classification problem
  - Problem reduces to multitask

- Multi-Label
  - Each label can be framed as one task then learn multiple related tasks

- Transfer Learning
  - Multitask Learning across several domains
Multi-View Learning Overview

- Multi-view learning set up
- Active Multi-view learning
- Multi-Task Multi-view learning
Multi-View Setting

- Course webpage classification [Blum & Mitchell, COLT’98]
Multi-View Setting

- Course webpage classification [Blum & Mitchell, COLT’98]
Multi-View Setting

- Each data is described from multiple views
  - web: page content + hyperlink [Blum and Mitchell, COLT’98]
  - patient: multiple tests
  - social network: user profile + friend links [Yang et al, CIKM’14]
  - image: color + texture [Zhou et al, ML’04]
- One classifier for each view, learn all classifiers jointly
- Major assumptions [Du et al, TKDE’10]
  - each view is sufficient for learning (view redundancy)
  - conditional view independency or expansion
Main Idea of MVL

- Maximize view agreement of *unlabeled* data
  - Semi-supervised MVL: classifiers agree on different views of unlabeled data, e.g. co-regularization [Sindhwani et al, ICML ’05]
    \[
    \min \sum_{j=1}^{2} \sum_{i=1}^{n} \left\| h_j(x_i^{(j)}) - y_i \right\|^2 + \lambda \cdot \sum_{i=1}^{m} \left\| h_1(u_i^{(1)}) - h_2(u_i^{(2)}) \right\|^2
    \]
  - Unsupervised MVL: clusters agree on different views of unlabeled data, e.g. co-regularized spectral clustering [Kumar and Daume, ICML ’11]
    \[
    \min \sum_{j=1}^{2} tr\left( (U^{(j)})^T L U^{(j)} \right) + \lambda \cdot D\left( U^{(1)}, U^{(2)} \right)
    \]
Advantages of MVL

- Effectively use massive unlabeled data to improve learning
  - Use view-agreement to restrict hypothesis space [Sindhwani et al, ICML’05]
- Alleviate curse of dimensionality
  - Each view has lower dimension than stacked view [Chen et al, ICML’11]
- Better performance than single-view learning
  - Classification [Blum and Mitchell, COLT’98, Nigam and Ghani, CIKM’00, Chen et al, ICML’11]
  - Clustering [Bickel and Scheffer, ICDM’04, Kumar and Daume, ICML’11, Kumar and Daume, NIPS’11]
MVL Algorithm

Multi-view Learning

Semi-supervised

Passive

Co-training [Blum98]
Co-EM [Nigam00]
Co-regularization [Sindhwani05]
Bayesian co-training [Yu08]

Active

Label Request

Co-testing [Muslea10]

View Request

Active view sensing [Yu09]

Unsupervised

Passive

Multi-view clustering [Bickel04]
CCA [Chaudhuri09]
Co-trained spectral clustering [Kumar11a]
Co-regularized SC [Kumar11b]
Feature learning [Wang13]
Active View Sensing (AVS)

- Actively collect views to improve learning efficient
- Existing AVS algorithms are heuristic [Yu et al, AISTAT’09]
  - Maximal predictive variance
  - Maximal mutual information
- Open questions
  - What guarantees can be obtained for AVS?
  - How to design more efficient AVS algorithms?
Active Sensing

- Question 1: What if some views are not complete?
- Question 2: How to select an unobserved (sample, view) pair for feature acquisition? \(\rightarrow\) Active Sensing
- Important in many medical diagnosis problems
Active Sensing: The Problem

- Features are incomplete in a multi-view setting
- Goals
  - Actively select a (sample, view) pair for feature acquisition
  - Improve overall system performance (not just that sample)!
- Challenges
  - How to measure the quality of each unobserved pair?
  - How to let one selected pair \((i,j)\) help all other samples?
- Active sensing in Bayesian co-training:
  - The whole co-training kernel is changed if one additional pair is observed – one pair helps all the samples!
Sample Complexity of MVL

- **Labeled sample complexity**
  - number of labeled data to guarantee the accuracy of a supervised multi-view learner [Balcan et al, NIPS’04, Wang and Zhou, ICML’08, Wang et al, ICML’13]

- **Unlabeled sample complexity**
  - number of unlabeled data to guarantee the accuracy of a semi-supervised multi-view learner [Balcan et al, NIPS’04]

- **Pseudo unlabeled sample complexity**
  - unlabeled data assumed assigned with true labels [Balcan et al, NIPS’04, Wang and Zhou, ICML’08]
  - jointly used with the assumption of hypotheses being never confident but wrong about their predictions
Sample Complexity of MVL

- **Labeled sample complexity**
  - number of labeled data to guarantee the accuracy of a supervised multi-view learner [Balcan et al, NIPS’04, Wang and Zhou, ICML’08, Wang et al, ICML’13]

- **Unlabeled sample complexity**
  - number of unlabeled data to guarantee the accuracy of a semi-supervised multi-view learner [Balcan et al, NIPS’04]

- **Pseudo unlabeled sample complexity**
  - unlabeled data assumed assigned with true labels [Balcan et al, NIPS’04, Wang and Zhou, ICML’08]
  - jointly used with the assumption of hypotheses being never confident but wrong about their predictions
View Complexity

● Definition

  ● The number of requested samples in an *arbitrary* view that guarantees a desired learning accuracy of an output multi-view classifier

● Difference to labeled sample complexity

  ● *Label* has ground truth for supervised analysis [Hanneke, ICML’07, Wang and Zhou, IDML’08]

  ● *View* does not necessarily provide ground truth (unlabeled sample)

● Relation to unlabeled sample complexity

  ● In two-view case, view complexity = unlabeled sample complexity
Disagreement-based AVS

Algorithm 1 Disagreement-based Active View Sensing

Input: Training set $S$ containing incomplete samples

for $i = 1$ to $n$ do

1: Select $k$ incomplete data from $S \cap VDR(\mathcal{H}(S))$
2: Request missing views for these data
3: Update $S$

end for

Output: Hypothesis $h$ from $\mathcal{H}(S)$

Definition 2. The view-disagreement region of any $S \subseteq \mathcal{X}$ with respect to any $\mathcal{T} \subseteq \mathcal{H}$ is

$$VDR(S; \mathcal{T}) = \{x \in S \mid \exists h \in \mathcal{T}, h_1(x) \neq h_2(x)\}.$$
Main Result

- View complexity of DAS is $O(\lg 1/\epsilon)$

**Theorem 4.1.** Let $\mathcal{D}$ be $\alpha$-expanding with respect to $\mathcal{H}$, and suppose $er(h_1) \leq \epsilon_0$ and $er(h_2) \leq \epsilon_0$ for all $h \in \mathcal{H}$ and some $\epsilon_0 > 0$. Let there be $m$ incomplete unlabeled samples requested by Algorithm 1. For any $\epsilon \leq (1 - \epsilon_0^2)/2$, $\delta > 0$ and all $h \in \mathcal{H}(S)$, with probability at least $1 - \delta$, $er(h) \leq \epsilon$ if

$$m \geq \log_2 \frac{1}{\lambda \epsilon} \cdot 2\theta \left( \log |\mathcal{H}| + \log \frac{3 \log_2 \frac{1}{\lambda \epsilon}}{\delta} \right),$$

where $\lambda = 2\alpha/(2 + \alpha)$. 
Experimental Study

- Data sets
  - Citeseer: paper content + citation link
  - Pima India Diabetes (PID): patient profile + medical tests

<table>
<thead>
<tr>
<th>Task</th>
<th>Training Set</th>
<th>Testing Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>labeled</td>
<td>unlabeled (complete)</td>
</tr>
<tr>
<td>AI vs AG</td>
<td>75 (^3) 6</td>
<td>548</td>
</tr>
<tr>
<td>DB vs ML</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>IR vs HCI</td>
<td>100</td>
<td>8</td>
</tr>
<tr>
<td>Diabetes</td>
<td>100</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Statistic of Four Classification Tasks
Experimental Study

- Comparing AVS algorithms
  - **RAN**: random sensing
  - **PV**: select data with maximal predictive variance AVS [Yu et al, AISTAT’09]
  - **DAS**: select data in view-disagreement region (VDR) *randomly*
  - **DAS-PV**: select data in VDR with maximal predictive variance

- At each round of sensing, each algorithm
  - pick up one data for view sensing
  - update classifier (co-regularization with least square loss)
  - evaluate performance on testing set
Experimental Result

(a) AI vs AG
Multi-view and Multi-task Learning

- Labeling samples is costly, and there are always much more unlabeled samples than labeled samples
  - Multi-view learning: incorporating data from multiple sources with unlabeled samples
- Different views are complementary and conditionally independent to each other.
- Learn one classifier on each view separately
- Single-view learning methods usually do not work well for ignoring other views or merging all views into one
Examples of Learning multiple related tasks with multi-view data

- Sentiment classification of movie reviews and news comments is two related tasks, and each can have multi-view data.
- In protein functional classification, each protein can be classified into multiple functional classes, and each protein has features from its sequence, chemical binding profiles among others.
Multi-view Multi-task (MVMT) Learning

Zhang & Huan, KDD’12
Multi-view Multi-task (MVMT) Learning

Task relatedness?
Multi-view Multi-task (MVMT) Learning

View consistency?
Inductive MVMTL

- Given multiple related tasks with multiple views for each task, the inductive MVMT learning problem is formulated as:
  - Within each task, the co-regularized MVL learning is applied to the unlabeled samples of multiple views, the disagreement among different view decision functions is minimized.
  - For the multiple tasks, the learned task functions are assumed “similar”, and hence their difference should be “small”.

11/5/14
The Objective Function

\[
\min_{\{w_t^v\}} \sum_{t=1}^{T} \frac{1}{2} \| y_t - \frac{X_t w_t}{V} \|^2 + \lambda \sum_{v=1}^{V} \| w_t^v \|^2 \\
+ \frac{\mu}{2} \sum_{v' \neq v} \| U_t^v w_t^v - U_t^{v'} w_t^{v'} \|^2 \\
+ \frac{\gamma}{2} \sum_{t' \neq t} \| w_t^v - w_{t'}^v \|^2
\]
Extension: Task Relationship Learning

• When task relationships are unknown, and can be positively or negatively correlated, or uncorrelated,
  • Learning a positive semi-definite covariance matrix $\Omega$ to model the task relationships

$$\min_{W,\Omega} \sum_{t}^{T} \frac{1}{2} \| y_t - X_t w_t \|^2 + \frac{\lambda}{2} \| W \|_F^2 + \frac{\gamma}{2} \text{tr}(W\Omega^{-1}W^T),$$

s.t. $\Omega \succeq 0$, $\text{tr}(\Omega) = 1$
Extension: Task Relationship Learning with Multiple Views

\[
\min_{\mathbf{w}, \Omega} \sum_{t}^{T} \frac{1}{2} \left\| \mathbf{y}_t - \mathbf{X}_t \mathbf{w}_t \right\|^2 + \frac{\mu}{2} \sum_{v' > v}^{V} \left\| \mathbf{U}_t \mathbf{w}_t - \mathbf{U}_t \mathbf{w}_t' \right\|^2 \\
+ \frac{\lambda}{2} \sum_{v=1}^{V} \left\| \mathbf{w}_t^v \right\|^2 + \frac{\gamma}{2} \sum_{v=1}^{V} \text{tr}(\mathbf{W}_t^v \Omega^{-1}_v \mathbf{W}_t^v T),
\]

s.t. \( \Omega \succeq 0, \ \text{tr}(\Omega) = 1 \)
Table 1: Statistics of Data Sets Used.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>WebKB</th>
<th>ECML2006</th>
<th>NUS-WIDE Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>3</td>
<td>4</td>
<td>2 $\sim$ 5</td>
</tr>
<tr>
<td>$T$</td>
<td>4</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>View Missing?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$N_p$</td>
<td>230</td>
<td>2,543</td>
<td>389 $\sim$ 1325</td>
</tr>
<tr>
<td>$N_n$</td>
<td>821</td>
<td>2,929</td>
<td>2220 $\sim$ 3156</td>
</tr>
<tr>
<td>$D$</td>
<td>2,096</td>
<td>5,597</td>
<td>634</td>
</tr>
</tbody>
</table>
Baseline Methods

- regMT: regularized MTL
- regMT+: multi-task relationship learning
- coMV: co-regularized MVL
- IteM2: Iterative MTMV

Our methods:
- regMVMT: task-regularization MVMT
- regMVMT+: task-relationship-learning MVMT
Results: complete-view data

Figure 3: Experimental results on the WebKB data set (left) and on the NUS-WIDE Object data set (right).
Results: missing-view data

Figure 4: Experimental results on the WebKB data set with one missing view for each task (left) and on the ECML2006 Email data set that has two missing views for each task (right).
Multi-label Learning

- Classification problem where multiple labels can be assigned to the same instance
Large number of labels

- Easy to find applications where number of labels are large
  - E.g. webpage classification, image annotation
  - If there are L unique labels, then each instance can be assigned any one of the $2^L$ subset.

- Two major ways to deal with large number of labels
  - Data Space reduction
  - Label space reduction
Data Space Reduction

- Yu et al, ICML’14, proposed a scalable solution for large scale multilabel learning with data space reduction
- Often users do not label all instances (Missing Labels)
  - While tagging articles, human only tag categories they know about
  - In image classification, only prominent objects are tagged and less prominent objects are neglected
- Algorithm handles missing values as well
Problem Formulation

$$X \in \mathbb{R}^{n \times d}$$

$$\hat{Z} = \arg \min Z^* J(Z) = \sum_{(i,j) \in \Omega} l(Y_{i,j}, f_j(x_i; Z)) + \lambda r(Z)$$

$$F(X, Z) = Z^T X$$

$$Z \in \mathbb{R}^{d \times L}$$

$$Y \in \{0,1\}^{n \times L}$$

Here, $\Omega$ is set of indices with known labels.
Problem Formulation

- $Z$ is assumed to be low rank to incorporate label correlations

$$Z = WH^T, \text{ where } W \in \mathbb{R}^{d \times k} \text{ and } H \in \mathbb{R}^{L \times k}$$

- Here, $k$ is less than both $d$ and $L$

$$r(Z) = \|Z\|_{tr} = \frac{1}{2}(\|W\|_F^2 + \|H\|_F^2)$$

$$J(W,H) = \sum_{(i,j) \in \Omega} l(Y_{i,j}, x^T_{i}Wh_{j}) + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)$$
Label space Reduction

- If number of labels are large, project labels to a reduced dimensional space (L<<K)

\[ X \in \mathbb{R}^{N \times D} \quad H : \text{Prediction} \quad \tilde{Y} \in \mathbb{R}^{N \times L} \quad P : \text{Encoding} \]

\[ \tilde{Y} \in \mathbb{R}^{N \times L} \quad \quad Y \in \mathbb{R}^{N \times K} \quad Q : \text{Decoding} \]

Input Space \quad Latent Label Space \quad Original Label Space
Label Space Reduction

• Two important components for successful learning
  • Model H for prediction of labels in latent label space
  • Decoded Q for conversion from latent label space to original space

• Methods of Label Space Reduction
  • Use sparsity of label space and utilize compressed sensing
    • E.g. Hsu et al, NIPS’09, Kapoor et al, NIPS’12
  • Use Transformations on Label Space such as PCA
    • E.g. Tai and Lin, Neural Computation’10, Chen and Lin, NIPS’12, Lin et al, ICML’14
Feature-aware Implicit Label Space Encoding (FaIE)

- Proposed by Lin et al, ICML’14
- Assumes $Y$ can be decomposed into code matrix $C_{NxL}$, and decoding matrix $D_{LxK}$, such that $Y=CD$
- Code vectors $C$ should also have strong correlation with input features $X$
- Original Labels also need to be fully recoverable by coding matrix $D$
Feature-aware Implicit Label Space Encoding (FaIE)

- Need to maximize recoverability and predictability

\[ \Omega = \max_{C,D} \Phi(Y, C, D) + \alpha \Psi(X, C) \]

- Recoverability is maximized by minimizing distance between Y and CD
- Predictability is maximized by maintaining correlation coefficient between X and C to be maximum
Conclusion

- Multi-label learning with a large number of labels is a challenging problem
- Current scenarios include datasets with labels of order $10^6$
Conclusion

- Heterogeneous learning is a critical problem at the interplay of machine learning and big data.
- Heterogeneous learning has natural connections to MTL, MVL, MLL, transfer learning and many others.
- Scalability is one of a major factor preventing the wide application of the aforementioned learning paradigm in big data.
Group Members

Ph.D. Students:  
Aaron Smalter, Bria Quanz, Hongliang Fei, Leo Zhang,  
Jia Yi, Meenakshi Mishra, Chao Lan,  
Department of EECS, University of Kansas
Acknowledgments

- The work is partially supported by
  - National Science Foundation, “CAREER: Mining Genome-wide Chemical-Structure Activity Relationships in Emergent Chemical Genomics Databases”, (IIS 0845951)
  - National Human Genome Research Institute “KU Specialized Chemistry Center” (U54 HG005031)
  - National Center for Research Resources, “KU Bioinformatics Computing Facility Core Renovation and Improvement” (RR031125)
  - GlaxoSmithKline
  - Some of the MVL slides are from Shipeng Yu
Thank you!

- For more information:
- Principal Investigator: Jun Huan
- Web: http://ittc.ku.edu/~jhuan/
References

- Balcan, Blum and Yang, Co-training and expansion: towards bridging theory and practice, in *Advances in Neural Information Processing Systems (NIPS)*, 2004
- Bickel and Scheffer, Multi-view clustering, in *International Conference on Data Mining (ICDM)*, 2004.
References

- Chaudhuri, Kakade, Livescu and Sridharan, Multi-View Clustering via Canonical Correlation Analysis, in International Conference on Machine Learning (ICML), 2009.
- Chen, Weinberger and Chen, Automatic feature decomposition for single view co-training, in International Conference on Machine Learning (ICML), 2011
References

- Gao, Liang, Fan and et al.. On community outliers and their efficient detection in information networks. KDD 2010.
References


- Hongliang Fei and Jun Huan. Structured Feature Selection and Task Relationship Inference for Multi-Task Learning, in Proceedings of the IEEE International Conference on Data Mining (ICDM'11)


- Jintao Zhang and Jun Huan , Inductive Multi-Task Learning with Multiple View Data , *The 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'12)* , Beijing, China, August 2012
References

References

- Nigam and Ghani, Analyzing the effectiveness and applicability of co-training, in *ACM Conference on Information and Knowledge Management (CIKM)*, 2000.
References

References

- Yu, Krishnapuram, Rosales and Rao, Active Sensing, in International Conference on Artificial Intelligence and Statistics (AISTAT), 2009.
References