Statistical Machine Learning
for Data Mining and Collaborative Multimedia Retrieval

Presented by Steven C.H. Hoi
Supervisor: Prof. Michael R Lyu
The Chinese University of Hong Kong
Date: 28 Aug, 2006
Time: 4:00 – 6:00 p.m.

Outline
- Background
- Contributions
- Learning Unified Kernel Machines
- Batch Mode Active Learning
- Collaborative Multimedia Retrieval
- Conclusions
- Future Work

Outline
- Background
- Contributions
- Learning Unified Kernel Machines
- Batch Mode Active Learning
- Collaborative Multimedia Retrieval
- Conclusions
- Future Work

Background
- Statistical Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Semi-Supervised Learning
  - Active Learning
  - Distance Metric Learning
  - Others (reinforcement learning, etc.)

Contributions
- Learning Unified Kernel Machines
  - Spectral Kernel Learning
  - Unified Kernel Logistic Regression
  - Kernel Design via Marginalized Kernel
  - Publications: KDD 06, WWW 06
- Batch Mode Active Learning
  - BMAL for Text and Image Categorization
  - Publications: ICML 06, WWW 06
- Distance Metric Learning
  - Discriminative Component Analysis (DCA) and KDCA
  - Publication: CVPR 06
- Collaborative Multimedia Retrieval
  - Learning Log-Based Relevance Feedback
  - Learning Reliable Distance Metrics
  - Publications: MM04, EMMA 05, TKDE 06, MMSJ 06

Overview of Contributions
Part I: Learning Unified Kernel Machines

- Motivation of Our Framework
  - Kernel machines play an important role in the state-of-the-art machine-learning techniques for data mining.
  - Supervised Learning
    - Support Vector Machines (SVM)
    - Kernel Logistic Regressions (KLR)
    - Regularized Least-Square Classifiers (RLS)
  - Unsupervised Learning
    - Spectral Clustering, Kernel PCA, ...
  - Active Learning
    - Margin-Based Active Learning with Kernel Machines, etc.
  - How to combine these kernel machine-learning techniques in a unified solution?

Semi-Supervised Kernel Learning

- Goal
  - To learn an effective kernel (matrix) from both labeled and unlabeled data
- Theoretical Principles
  - Unsupervised Kernel Design
    - Learning Kernel from unlabeled data
  - Kernel Target Alignment
    - Learning Kernel from labeled data

Overview of Kernel Machine Learning

- Semi-Supervised Learning
  - Given \( l \) training examples \((x_1, y_1), \ldots, (x_l, y_l)\), and \((n-l)\) unlabeled data examples \((x_{l+1}, x_{l+2}, \ldots, x_n)\), let \( f \) be \( n \)-dimensional real vector, which is learned by the following semi-supervised learning method:

\[
 f = \arg\min_{f} \left( \frac{1}{l} \sum_{i=1}^{l} C(y_i, f(x_i)) + \lambda \|f\|_2^2 \right)
\]  

- Theorem (Zhang et al., NIPS'05): The solution of (3) is equivalent to the solution of (1):

\[
 \hat{f} = \hat{f}(y) \quad f = 1, \ldots, n
\]

Unsupervised Kernel Design

- The equivalence theorem shows that, in order to exploit the unlabeled data, we can consider the following supervised learning approach with unsupervised kernel design:
  - (1) Design a new kernel \( K' \) using unlabeled data
  - (2) Apply the new \( K' \) in the supervised learning formula
- Spectral Kernel Design
  - Principle: A kernel with faster spectra decay should be more preferred. (Zhang et al., NIPS'05)
Formulation of Algorithm (cont')

- Spectral Kernel Learning

**Kernel Target Alignment**

- **Kernel Alignment** (Cristianini et al. 2002): The empirical alignment of two given kernels $K_1$ and $K_2$ with respect to a sample set is the following quantity:

$$X(K_1, K_2) = \frac{\langle K_1, K_2 \rangle}{\sqrt{\langle K_1, K_1 \rangle \langle K_2, K_2 \rangle}}$$

where $\langle K_1, K_2 \rangle = \sum_{x \in \mathcal{X}} \langle K_1(x), K_2(x) \rangle$.

- **Target Kernel**
  - Let $y = [y_1, ..., y_l]$ be a label vector of training data, for binary classification, the target kernel can be defined as:
    $$T = yy^T = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

- **Kernel Alignment (Cristianini et al. 2002): Part I: UKM**

$$\text{Maximizing}$$

Principle: A better kernel can be optimized by maximizing the following kernel target alignment:

$$\hat{X}(K_0, T) = \frac{\langle K_0, T \rangle}{\sqrt{\langle K_0, K_0 \rangle \langle T, T \rangle}}$$

**Spectral Kernel Learning**

- **Principles**
  - Maximizing kernel target alignment while keeping fast spectra decay!
  - Formulation of Algorithm

$$\begin{align*}
\min_{r, \mu} & \quad r^T D^T D r \\
\text{subject to} & \quad \mu \geq 0 \\
& \quad \mu \geq C_{\mu} \quad i = 1, ..., d - 1 \\
& \quad \mu_i = 0 \\
& \quad ||D||_1 = 1 \\
& \quad ||D||_0 = 1 \\
& \quad \mu \geq 0 \\
& \quad \mu_i = 0 \\
\end{align*}$$

This is a standard Quadratic Programming (QP) problem.

Connections to Other Kernel Techniques

- Spectral Kernel Learning (SKL)
  - Cluster Kernel ($[1, ..., 1, 0, ..., 0]$. Spectral Clustering)
  - Truncated Kernel (top eigen components, Kernel PCA)

When setting $C=1$, $d=n$, and assuming the initial kernel $K$ is constructed from graph Laplacian $L$, our SKL method is equivalent to the order-constrained graph kernel (Jung Zhu, NIPS 2005)
Remarks
- It is an open issue to determine the convergence condition!
- We simply repeat the learning procedure in a fixed step.
- Active learning may be done more elegantly, e.g., to search a batch of informative examples.

Unified KLR Paradigm for Classification
1) Calculate an initial kernel matrix $K_0$
2) Learn a new kernel by the SKL algorithm
   $$K = \text{Spectral Kernel}(K_0, \lambda, \nu)$$
3) Train a standard KLR classifier with new $K$
   $$f(x) = \sum_{i=1}^{n} \alpha_i (C_i | x) \log (p(C_i | x))$$
4) Active learning to seek informative data

Experimental Testbed and Setups
- Four UCI datasets

Table 1: List of UCI multiclass learning datasets
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attributes</th>
<th>Classes</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ionosphere</td>
<td>34</td>
<td>2</td>
<td>351</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>4</td>
<td>270</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>3</td>
<td>178</td>
</tr>
</tbody>
</table>

Two objectives of experimental evaluation
- How effective is our SKL algorithm in learning semi-supervised kernels?
- How effective is our UKLR scheme compared with traditional classification solutions?
Experimental Results

- Semi-Supervised Kernel Learning
  - Compared Kernels
    - 3 standard kernels
    - Linear, Quadratic, RBF
    - 5 semi-supervised kernels
    - 3 SKL methods with different initial kernels
    - 2 Order-constraint graph kernels
  - Standard KLR classifier for classification
  - Settings
    - Fix decay factor C (C>1)
    - Set dimension cut-off d = 20
    - 20 trials for each experimental comparison

Experimental Results

Table 2. Classification performance of different kernels using KLR classifiers on UCI datasets. The mean accuracies and standard errors are shown in the table. Each cell in the table has two rows. The upper row shows the test set accuracy with standard error; the lower row gives the average time used in kernel learning.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Linear</th>
<th>Quadratic</th>
<th>RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Acc</td>
<td>Time (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UKM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Experimental Results

Table 3: Classification performance of different classification schemes on four UCI datasets. The mean accuracies and standard errors are shown in the table. “KLR” represents the initial classifier with the initial train size; other three methods are trained with additional 10 random/active examples.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Linear</th>
<th>Quadratic</th>
<th>RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Acc</td>
<td>Time (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UKM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summary of Part I

- We presented a framework of learning unified kernel machines (UKM) for classification.
- A new semi-supervised kernel learning algorithm was proposed, which is related to an equivalent quadratic programming (QP) problem.
- A classification paradigm was developed by applying our UKM framework on the KLR model.
- Empirical evaluations are conducted on several UCI datasets.
Part II: Batch Mode Active Learning for Text Categorization

Motivation

Text Categorization
Logistic Regression and Active Learning
Batch Mode Active Learning
Theoretical Foundation
Convex Optimization Formulation
Eigen Space Simplification
Bound Optimization Algorithm
Experimental Results
Summary

Motivation

Text Categorization

Problem: assign documents to predefined topics
Significances
Core Web data mining technique
Applications: category browsing, vertical search, etc.

Challenges
To build efficient classifiers
To minimize human labeling effort

Motivation

Logistic Regression

Efficiency for Training and Prediction
Natural Probability Output
State-of-the-art performance, etc.
Linear model

\[ p(y|x) = \frac{1}{1 + \exp(-y(w^T x + b))} \]

where \( y \in \{+1, -1\} \) is the class label.
Simplified notation:

\[ p(y|x) = \frac{1}{1 + \exp(-y\alpha^T x)} \]

Theoretical Foundation

Main Idea:
Based on the theoretical framework of maximization of Fisher information

Problem Setting
In a probabilistic classification framework, assume the classification model is a semi-parametric form

\[ p(x, y|\alpha) = p(x)p(y|x, \alpha) \]

For example, the logistic regression model:

\[ p(x, y|\alpha) = \frac{1}{1 + \exp(-y\alpha^T x)} p(x) \]
Theoretical Foundation

The problem of batch mode active learning can be regarded as a problem to seek a resample distribution \( q(x) \) of the unlabeled data.

The examples with large resampling probabilities will be selected as the most informative ones for labeling.

According to statistical estimation theory, active learning should consider a resample distribution \( q(x) \) that maximizes the following Fisher information:

\[
I_q(\alpha) = -\int q(x)dx \int \frac{\partial^2}{\partial \alpha^2} \log p(y|x, \alpha) dy
\]

Convex Optimization Formulation

Rewrite the objective function \( \text{tr}(I_q^{-1} I_p) \) as

\[
\text{tr}(I_q^{-1/2} I_p I_q^{-1/2})
\]

Introduce a slack matrix \( M \in \mathbb{R}^{n \times n} \), then turn the original problem into the following optimization:

\[
\min_{q,M} \text{tr}(M)
\]

\[
\text{s. t. } \sum_{i=1}^n q_i \pi_i (1 - \pi_i) \left( x_i x_i^T \frac{I_p^{-1/2}}{I_q} \right) \geq 0
\]

\[
\sum_{i=1}^n q_i = 1, q_i \geq 0, i = 1, \ldots, n
\]

In the above, we use \( \text{tr}(A) \geq \text{tr}(B) \) if \( A \succeq B \)

Eigen Space Simplification

Directly solving the above optimization problem may be computationally expensive for the large-size slack matrix variable of \( M \).

In order to reduce the computational complexity, we propose an Eigen space simplification method to make the solution simpler and more effective.

We assume that \( M \) is expanded in the Eigen space of the Fisher information matrix \( I_p \).
Eigen Space Simplification

Let \( \{ (\lambda_1, v_1), ..., (\lambda_s, v_s) \} \) be the top \( s \) eigen vectors of the Fisher information matrix \( I_p \), where \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s \), then we assume the matrix \( M \) has the following form:

\[
M = \sum_{k=1}^{s} \gamma_k v_k v_k^T, \quad \gamma_k \geq 0, \ k = 1, \ldots, s.
\]

The inequality \( M \geq I_p^{1/2} M^{-1} I_p^{1/2} \) can be rewritten as:

\[
I_q \geq I_p^{1/2} M^{-1} I_p^{1/2}, \quad \sum_{k=1}^{s} \frac{\gamma_k}{\lambda_k} \geq 1, \quad k = 1, \ldots, s.
\]

The inequality can be rewritten as:

\[
\frac{I_q}{\gamma_k} \geq I_p^{1/2} M^{-1} I_p^{1/2} v_k v_k^T, \quad \sum_{k=1}^{s} \frac{\gamma_k}{\lambda_k} \geq 1, \quad k = 1, \ldots, s.
\]

Using the eigen expression, we have

\[
I_q^{1/2} I_p^{-1/2} I_q^{1/2} \geq \frac{1}{\gamma_k} I_p^{1/2} M^{-1} I_p^{1/2} v_k v_k^T, \quad \forall v \in \mathbb{R}^d,
\]

we then have the following result

\[
\frac{v_k^T I_q v}{\gamma_k} \geq \lambda_k \frac{v_k^T v}{v_k^T v}, \quad k = 1, \ldots, s.
\]

The previous necessary condition leads to following constraints:

\[
\gamma_k \geq \frac{\lambda_k}{v_k^T v_k} = \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1 - \pi_i) (v_i^T v_k)^2}, \quad k = 1, \ldots, s.
\]

Meanwhile, the objective function of \( \text{tr}(M) \) can be expressed as

\[
\text{tr}(M) = \sum_{k=1}^{s} \gamma_k
\]

By putting the above two expressions together, we transform the SDP problem into the following approximate optimization problem:

\[
\min_{q_i} \sum_{i=1}^{n} q_i \pi_i (1 - \pi_i) (v_i^T v_k)^2
\]

subject to

\[
\sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1, \ldots, n.
\]

Note that the above optimization problem belongs to convex optimization since \( f(x) = 1/x \) is convex when \( x \geq 0 \).

Lemma 1: Let \( L(q) \) be the objective function,

\[
L(q) = \sum_{i=1}^{n} q_i \pi_i (1 - \pi_i) (v_i^T v_i)^2
\]

we have the following conclusion:

\[
L(q) \leq \sum_{i=1}^{n} \left( \frac{q_i}{q_i} \right)^2 \pi_i (1 - \pi_i) \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1 - \pi_i) (v_i^T v_k)^2}
\]

Proof in Appendix.

Bound Optimization Algorithm

Given the lemma 1, now instead of optimizing the original objective function \( L(q) \), we can optimize its upper bound using simple updating equations:

\[
q_i \rightarrow q_i \pi_i (1 - \pi_i) \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1 - \pi_i) (v_i^T v_k)^2},
\]

\[
q_i \rightarrow \frac{q_i}{\sum_{i=1}^{n} q_i}.
\]

This algorithm will guarantee to converge to a local optimal. Since the original problem is a convex optimization problem, the above updating procedure will guarantee to converge to a global optimal.
Bound Optimization Algorithm

- The updating step:

\[ \psi_i = \pi_i (1 - \pi_i) \sum_{k=1}^{n} \frac{(x^T v_k)^2}{\alpha_k} \]

- Some Observations
  1. The example with a large classification uncertainty will be assigned with a large probability.
  2. The example that is similar to many unlabeled examples is more likely to be selected.

Experimental Testbeds

- 3 standard text datasets
  - Reuters-21578 dataset (10788)
  - Two web-related datasets: WebKB (4518) and Newsgroup (10966)

Experimental Settings

- A standard feature selection by Information Gain is conducted to remove uninformative features, in which 500 of the most informative features are selected.
- The F1 metric is adopted as our evaluation metric, which has been shown to be more reliable metric than other metrics such as the classification accuracy. More specifically, the F1 is defined as

\[ F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]

where p and r are precision and recall.
- Parameters of LogReg and SVM are determined by a standard cross validation method.

Comparison Schemes

- Two popular active learning methods:
  1. SVM-AL: the classification uncertainty of an example \( x \) is determined by its distance to the decision boundary

\[ d(x; w, b) = \|w^T x + b\| \]

The smaller the distance \( d(x; w, b) \) is, the more the classification uncertainty will be.
  2. LogReg-AL: the logistic regression active learning algorithm that measures the classification uncertainty based on the entropy of the distribution \( p(y|x) \).

\[ H(p) = -p(-|x|) \log p(-|x|) - p(+|x|) \log p(+|x|) \]

The larger the entropy of \( x \) is, the more uncertain we are about the class labels of \( x \).
- Our Batch Mode Active Learning algorithm with logistic regression, i.e., LogReg-BMAL in short.

Empirical Evaluation

- Experimental Results with Reuters-21578
  - average results over 40 executions
  - 100 training examples and 100 active examples

<table>
<thead>
<tr>
<th>Category</th>
<th>SVM</th>
<th>LogReg</th>
<th>SVM-AL</th>
<th>LogReg-AL</th>
<th>LogReg-BMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sports</td>
<td>0.12</td>
<td>0.22</td>
<td>0.19</td>
<td>0.14</td>
<td>0.09</td>
</tr>
<tr>
<td>Business</td>
<td>0.09</td>
<td>0.13</td>
<td>0.12</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>Health</td>
<td>0.08</td>
<td>0.12</td>
<td>0.12</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Sci/Tech</td>
<td>0.10</td>
<td>0.13</td>
<td>0.11</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>Travel</td>
<td>0.11</td>
<td>0.14</td>
<td>0.13</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Opinion</td>
<td>0.09</td>
<td>0.12</td>
<td>0.11</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.10</td>
<td>0.13</td>
<td>0.11</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>100%</td>
<td>0.10</td>
<td>0.13</td>
<td>0.12</td>
<td>0.11</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 4: Experimental results of F1 performance on the Reuters-21578 dataset using 10 training samples (\( \alpha \)).

- Experimental Results with Reuters-21578

![Graph showing F1 performance](image)
Empirical Evaluation

- **Experimental Results with Web-KB Dataset**

<table>
<thead>
<tr>
<th>Category</th>
<th>SVM</th>
<th>LogReg</th>
<th>SVMLog</th>
<th>LogRegLog</th>
<th>LogRegBMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>sports</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>scientific</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>health</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>project</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>stuff</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>statistic</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
</tbody>
</table>

Table 1: Empirical results of F1 performance on the WebKB dataset using 10 training samples (%).

- **Experimental Results with Newsgroup Dataset**

<table>
<thead>
<tr>
<th>Category</th>
<th>SVM</th>
<th>LogReg</th>
<th>SVMLog</th>
<th>LogRegLog</th>
<th>LogRegBMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>1</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>3</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>4</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>5</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>6</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>7</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>8</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>9</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
<tr>
<td>10</td>
<td>0.93±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
</tr>
</tbody>
</table>

Table 2: Empirical results of F1 performance on the Newsgroup dataset using 10 training samples (%).

Summary of Part II

- A new active learning scheme is suggested for text categorization to overcome the limitation of traditional active learning;
- A batch mode active learning solution is formulated by convex optimization techniques;
- An effective bound optimization algorithm is proposed to solve the batch mode active learning problem;
- Extensive experiments are conducted for empirical evaluations in comparisons with state-of-the-art active learning approaches for text categorization.

Collaborative Multimedia Retrieval via Regularized Distance Metric Learning

- **Problem Definition**
  - Collaborative Multimedia Retrieval (CMR) is a Multimedia Information Retrieval (MIR) problem which involves human interactions, either with online relevance feedback explicitly or with historical log data of users’ relevance feedback implicitly.

Motivation

- **Relevance Feedback**
  - A powerful tool for multimedia information retrieval
  - Popular methods: SVM Based solutions
- **Log-based Relevance Feedback (LRF)**
  - Combining log data for online relevance feedback
  - Our contribution: Soft Label SVM for LRF (MM 04, TKDE 06)
- **Learning Distance Metrics with Log Data**
  - Our contribution: Regularized Distance Metric Learning for learning robust and scalable metrics (ACM MM Journal 06)

Regularized Distance Metric Learning

- **Overview**
  - The basic idea of this work is to learn a desired distance metric in the space of low-level image features that effectively bridges the semantic gap.
  - It is learned from the log data of user relevance feedback based on the Min/Max principle, i.e., minimize/maximize the distance between similar/dissimilar images.
Regularized Distance Metric Learning

### Formulation

- The log data are given in terms of log sessions.
- Each log session: each image was marked either relevant (+1), irrelevant (-1), or unknown (0).

**Image examples in the database**

<table>
<thead>
<tr>
<th>Session</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>Image 4</th>
<th>Image 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

This tells us:
- **When two images are judged as relevant in the same log session,** they **could** be similar to each other.
- **When one image is judged as relevant and another is judged as irrelevant in the same log session,** they **must** be dissimilar to each other.

Where $Q$ stands for number of log sessions in the log data.

### Formulation

The formulation in (4) may not be robust for noise, we form a new objective function for distance metric learning that takes into account both the **discriminative** issue and the **robustness** issue as:

$$
\min_{\mathbf{A}} \|\mathbf{A}\|_F + \alpha \sum_{i,j} |x_i - x_j|^2
$$

s.t. $\mathbf{A} \succeq 0$

(5)

$$
\text{where } \|\mathbf{A}\|_F \text{ stands for the Frobenius norm}. \text{ If } \mathbf{A} = [a_{i,j}]_{m \times n}, \text{ the Frobenius norm is defined as:}
$$

$$
\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{i,j}^2}
$$

### Formulation

Putting Eqn. (6), (7), (8) together, we have the final formulation for the regularized metric learning:

$$
\min_{\mathbf{A}} \left( \sum_{i,j} a_{i,j}^2 \right)^{1/2} + \alpha \sum_{i,j} a_{i,j} - \beta \sum_{i,j} a_{i,j}^2
$$

s.t. $\mathbf{A} \succeq 0$

(9)

### Formulation

To convert the above problem into the standard form, we introduce a slack variable $t$ that upper bounds the Frobenius norm of matrix $\mathbf{A}$, which leads to an equivalent form of (9), i.e.,

$$
\min_{\mathbf{A}} t + \alpha \sum_{i,j} a_{i,j} - \beta \sum_{i,j} a_{i,j}^2
$$

s.t. $\sum_{i,j} a_{i,j}^2 \leq t$

$$
\mathbf{A} \succeq 0
$$

(10)

The first constraint is called a second order cone constraint.
The second constraint is a positive semi-definite constraint.
A special form of Convex optimization problems!
There exists efficient solutions to solve it in a polynomial time.
Experimental Results

- **Datasets**
  - 20-Category
  - 50-Category

- **Image Representation**
  - 9-dimensional Color Histogram
  - 18-dimensional Edge Histogram
  - 9-dimension texture

---

Collection of Users’ Log Data

| Table 1: The characteristics of users' log data on the 20-category and 50-category testbeds. |
|-----------------|-----------------|-----------------|-----------------|
| # Log Sessions | Noise Degree | # Log Sessions | Noise Degree |
| 20-Category | 100 | 7.3% | 100 | 16.2% |
| 50-Category | 150 | 7.3% | 150 | 17.3% |

---

Compared Schemes:

1. “Euclidean”: Euclidean metric without log data.
2. “IML”: based on the semantic representation learned from the manifold learning algorithm.
3. “DML”: based on the metric learned by a typical distance metric learning algorithm.
4. “RDML”: based on the metric by proposed regularized metric learning algorithm.

---

Table 2: Average precision (%) of top-ranked images on the 20-Category testbed over 2,000 queries. The relative improvement of algorithms IML, DML, and RDML over the baseline Euclidean is included in the parenthesis following the average accuracy.

<table>
<thead>
<tr>
<th>Top Images</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>IML (Large Noise)</td>
<td>37.94(4.9%)</td>
<td>30.16(7.9%)</td>
<td>26.95(8.17)</td>
<td>25.58 (+1.1%)</td>
<td>24.47(2.2%)</td>
</tr>
<tr>
<td>DML (Large Noise)</td>
<td>33.93(2.7%)</td>
<td>30.50(3.1%)</td>
<td>20.48(0.4%)</td>
<td>20.41 (0.9%)</td>
<td>20.42(0.8%)</td>
</tr>
<tr>
<td>RDML (Large Noise)</td>
<td>33.93(2.7%)</td>
<td>30.49(2.5%)</td>
<td>20.48(0.4%)</td>
<td>20.41 (0.9%)</td>
<td>20.42(0.8%)</td>
</tr>
</tbody>
</table>

---

Table 3: Average precision (%) of top-ranked images on the 50-Category testbed over 5,000 queries.

<table>
<thead>
<tr>
<th>Top Images</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>39.98</td>
<td>36.56</td>
<td>33.94</td>
<td>31.42</td>
<td>29.19</td>
</tr>
<tr>
<td>IML (Large Noise)</td>
<td>33.96(6.3%)</td>
<td>29.39(8.7%)</td>
<td>24.75(4.6%)</td>
<td>21.66 (2.4%)</td>
<td>18.84(4.4%)</td>
</tr>
<tr>
<td>DML (Large Noise)</td>
<td>32.07(7.9%)</td>
<td>27.93(6.8%)</td>
<td>23.73(4.9%)</td>
<td>21.44 (3.5%)</td>
<td>17.46(2.7%)</td>
</tr>
<tr>
<td>RDML (Large Noise)</td>
<td>30.41(10.6%)</td>
<td>28.92(12.0%)</td>
<td>20.44(13.7%)</td>
<td>20.17 (13.4%)</td>
<td>18.80(13.3%)</td>
</tr>
</tbody>
</table>

---

Table 4: Average precision (%) of top-ranked images on the 20-Category testbed for IML, DML, and RDML using noisy log data. The relative improvement over the baseline Euclidean is included in the parenthesis following the average accuracy.

<table>
<thead>
<tr>
<th>Top Images</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>36.30</td>
<td>27.96</td>
<td>25.56</td>
<td>22.21</td>
<td>20.48</td>
</tr>
<tr>
<td>IML (Large Noise)</td>
<td>38.09(1.5%)</td>
<td>27.96(4.8%)</td>
<td>25.56(5.1%)</td>
<td>22.21 (4.8%)</td>
<td>18.86(18.3%)</td>
</tr>
<tr>
<td>DML (Large Noise)</td>
<td>32.45(3.1%)</td>
<td>28.86 (+1.3%)</td>
<td>25.68 (5.2%)</td>
<td>22.21 (4.8%)</td>
<td>19.46(4.4%)</td>
</tr>
<tr>
<td>RDML (Large Noise)</td>
<td>28.40(2.9%)</td>
<td>25.89(3.1%)</td>
<td>24.88(3.8%)</td>
<td>22.21 (4.8%)</td>
<td>20.47(4.4%)</td>
</tr>
</tbody>
</table>

---

Table 5: Average precision (%) of top-ranked images on the 50-Category testbed for IML, DML, and RDML using noisy log data.

<table>
<thead>
<tr>
<th>Top Images</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>58.30</td>
<td>36.56</td>
<td>25.56</td>
<td>22.21</td>
<td>20.48</td>
</tr>
<tr>
<td>IML (Large Noise)</td>
<td>38.09(1.5%)</td>
<td>27.96(4.8%)</td>
<td>25.56(5.1%)</td>
<td>22.21 (4.8%)</td>
<td>18.86(18.3%)</td>
</tr>
<tr>
<td>DML (Large Noise)</td>
<td>32.45(3.1%)</td>
<td>28.86 (+1.3%)</td>
<td>25.68 (5.2%)</td>
<td>22.21 (4.8%)</td>
<td>19.46(4.4%)</td>
</tr>
<tr>
<td>RDML (Large Noise)</td>
<td>28.40(2.9%)</td>
<td>25.89(3.1%)</td>
<td>24.88(3.8%)</td>
<td>22.21 (4.8%)</td>
<td>20.47(4.4%)</td>
</tr>
</tbody>
</table>

---

Efficiency and Scalability

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>20-Category</th>
<th>50-Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>IML</td>
<td>68.25</td>
<td>68.25</td>
</tr>
<tr>
<td>DML</td>
<td>68.25</td>
<td>68.25</td>
</tr>
<tr>
<td>RDML</td>
<td>68.25</td>
<td>68.25</td>
</tr>
</tbody>
</table>
Summary of Part III

- We proposed a novel algorithm for distance metric learning, which boosts the retrieval accuracy of CBIR by taking advantage of the log data of users’ relevance judgments.
- A regularization mechanism is used in the proposed algorithm to improve the robustness of solutions, when the log data is small and noisy.
- It is formulated as a positive semi-definite programming problem, which can be solved efficiently.
- Experiment results have shown that the proposed algorithm for regularized distance metric learning substantially improves the retrieval accuracy of the baseline CBIR system.

Summary of Other Contributions

- Distance Metric Learning for Clustering
  - Discriminative Component Analysis (DCA)
  - Kernel DCA for learning nonlinear metrics
  - Details in Appendix A
- Marginalized Kernels for Web Mining
  - Time-dependent similarity measure scheme
  - Marginalized kernels to exploit both explicit similarity and implicit cluster semantic for similarity measure
  - Details in Appendix B

Conclusions

- We proposed a framework of statistical machine learning for data mining and collaborative multimedia retrieval.
- We suggested a unified framework to learn the unified kernel machines, in which a new semi-supervised kernel learning algorithm was proposed.
- We explored the batch mode active learning problem and proposed a novel algorithm to search a batch of informative examples.
- We studied a real-world application, collaborative multimedia retrieval, and proposed a regularized distance metric learning algorithm for learning robust and scalable metrics for multimedia retrieval.

Future Work

- Theoretical Analysis on UKM …
- More effective algorithms and extensions to UKM …
- Employing UKM to solve real-world problems, classification, regressions, information retrieval, …

Selected Publications (Regular Papers)

Appendix

- A: Distance Metric Learning for Clustering
- B: Marginalized Kernels for Web Mining
- C: Proof of Lemma 1 in BMAL
- D: Definition of Semi-Definite Programming

Appendix A:
Distance Metric Learning for Clustering

Motivation
- We address important limitations of existing metric learning methods. Relevant Component Analysis (RCA)
- It lacks of considering negative constraints
- It cannot capture nonlinear relationship of data instances via linear transformation

Solution:
- Discriminative Component Analysis (DCA)
- Kernel DCA to learn nonlinear metrics

Discriminative Component Analysis

Formulation
- Given a set of data points $X = \{x_i\}_{i=1}^N$ and a set of contextual constraints
- Form $n$ chunklets using the positive $C_p = \{x_p^i\}_{i=1}^n$ constraints:
- Form a discriminative set $D_n$ to indicate which chunklets can be discriminated each other by the negative constraints.

Discriminative Component Analysis

Algorithm for solving DCA
- Idea: Based on the Fisher’s criterion, the DCA problem can be solved by diagonalizing $C_b$ and $C_w$ simultaneously

$$J(A) = \arg \max_A \frac{|A^T C_b A|}{|A^T C_w A|}$$

Steps:
1) Compute the covariance matrices $C_b$ and $C_w$ by Eq. (1),(2)
2) Diagonalize $C_b$ by eigenanalysis
3) Project and diagonalize $C_w$ by eigenanalysis
4) Output transformation matrix $A$

Kernel DCA

- The kernel techniques first map the input data into a feature space $F$.
- The data can be then analyzed in the projected feature space.
- The linear transformation in the feature space corresponds the nonlinear analysis in the input space.
- For example: Kernel PCA, Kernel ICA, Kernel LDA, etc.
Kernel DCA

Formulation

- We implicitly map the original data \( X = \{x_i\}_{i=1}^n \) in the input space \( I \) to a high-dimensional feature space \( F \) via some defined basis function.

\[ \phi : x \rightarrow \phi(x) \in F \]

- The similarity of two instances is measured:

\[
K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.
\]

- In general, we want to find the optimal \( M \):

\[
d_i(x_i, x_j) = \sqrt{(\phi(x_i) - \phi(x_j))^T M (\phi(x_i) - \phi(x_j))}
\]

\[ M = W^T W \]

The transformation matrix \( W \) can be represented as

\[ W = [w_1, \ldots, w_n]^T \]

where \( \alpha_i \) are the coefficients for the samples in the feature space.

The optimization problem for Kernel DCA can therefore be given as follows

\[ J(A) = \max_A \frac{A^T K A}{A^T K A} \]

The algorithm to solve the Kernel DCA is similar to the linear DCA.

Experimental Results

- Datasets

- Compared Schemes

  - (1) K-means-EU: the baseline method, i.e., typical k-means clustering based on the original Euclidean distance;
  - (2) CK-means-EU: the constrained k-means clustering method based on the original Euclidean distance [146];
  - (3) CK-means-RCA: the constrained k-means clustering method based on the distance metrics learned by RCA [8];
  - (4) CK-means-Xing: the constrained k-means clustering method based on the distance metrics learned by Xing et al. [153];
  - (5) CK-means-DCA: the constrained k-means clustering method based on the distance metrics learned by our DCA algorithm;
  - (6) CK-means-RBF: the constrained k-means clustering method based on the RBF kernel metrics;
  - (7) CK-means-KDCA: the constrained
Our Approach

Motivations

Time-Dependent Concepts

Motivation
Our Approach
Time-Dependent Concepts
Marginalized Kernels for Similarity Measure
Empirical Results

Calendar schema and pattern

Example
- Calendar schema <day, month, year>
- Calendar pattern <15, *, *>
- <15, 1, 2002> is contained in the pattern <15, *, *>

We studied the problem of learning distance metrics and data transformation using the contextual information for data clustering.

We proposed the Discriminative Component Analysis (DCA), which can exploit both positive and negative constraints in an efficient learning scheme.

We proposed KDCA to learn nonlinear metrics for data clustering.

Exploit the click-through data for semantic similarity of queries by incorporating temporal information.

To combine explicit content similarity and implicit semantic similarity via marginalized kernel techniques.
Time-Dependent Concepts

- Click-Through Subgroup

Example
- Based on the schema <day, week>, and the pattern <1,*>, <2,*>…<7,*>, we can partition the data into 7 groups, which correspond to Sun, Mon, Tue…Sat.

Similarity Measure

- Query similarity measures
  - Cosine function
  - Marginalized kernel
    - By introducing query clusters, one can model the query similarity in a more semantic way.

Empirical Evaluation

- Dataset
  - Click-through log of a commercial search engine:
    - June 16, 2005 to July 17, 2005
    - Total size of 22GB
    - Only queries from US
  - Calendar schema and pattern
    - <hour, day, month>, <1,*>, <2,*>…
    - Divide the data into 24 subgroups
    - Average subgroup size: 59,400,000 query-page pairs
Empirical Examples

- weather + forecast, fox + news

Summary

- Presented a preliminary study of the dynamic nature of query similarity using click-through data
- Using marginalized kernels for building an time-dependent model
- Conducted empirical evaluations from real-world web search data

Appendix C: Proof of Lemma 1

**Lemma 1:** Let $L(q)$ be the objective function in (15), we have the following conclusion

$$L(q) \leq \sum_{i=1}^{n} \frac{(q_i^{(1)} - q_i^{(2)})^2}{\sum_{j=1}^{m} \pi_j (1 - \pi_j) (x_j^T y_j)^2}$$

**Proof.**

$$L(q) = \sum_{i=1}^{n} \frac{(q_i^{(1)} - q_i^{(2)})^2}{\sum_{j=1}^{m} \pi_j (1 - \pi_j) (x_j^T y_j)^2}$$

This finishes the proof of the inequality lemma. □

Appendix D – Semi-Definite Programming (SDP)

$$\begin{array}{ll}
\text{minimize} & x^T P x \\
\text{subject to} & x_1 f_1 + x_2 f_2 + \cdots + x_n f_n + G \preceq 0 \\
& A x = b
\end{array}$$

with $F, G \in S^V$

- inequality constraint is called linear matrix inequality (LMI)
- includes problems with multiple LMI constraints: for example,
  $$x_1 f_1 + \cdots + x_n f_n + G \preceq 0, \quad x_1 f_1 + \cdots + x_n f_n + G \preceq 0$$

is equivalent to single LMI

$$x_1 [f_1, 0] + x_2 [f_2, 0] + \cdots + x_n [f_n, 0] + [G, 0] \preceq 0$$