Machine Learning Potpourri

Semi-supervised Learning, Learning with Multiple Kernels, and Network Intrusion Detection

Ulf Brefeld

22.10.2009, DTIC-Seminar, Universitat Pompeu Fabra
Building prediction models from data

- Task: Learn a mapping that assigns every \( x \in X \) the correct \( y \in Y \)
- Find \( f : X \rightarrow Y \) such that \( f(x) \) outputs the true \( y \in Y \)
- Measure the quality of \( f \) by a loss function \( \ell(y, f(x)) \)

Example 1: Text classification

- Input \( X = \{ x : x \text{ is a text document} \} \)
- Output \( Y = \{ \text{politics, sports, economy, ...} \} \)
- 0/1 loss function \( \ell(y, f(x)) = [[y \neq f(x)]] \)

Example 2: Regression (unknown function approximation)

- Input \( X = \{ x : x \in [a, b] \} \)
- Output \( Y = \{ y : y = g(x), x \in X, g \text{ unknown} \} \)
- Squared loss \( \ell(y, f(x)) = (y - f(x))^2 \)
Risk Minimization (=Generalization Error)

Building prediction models from data

• Task: Learn a mapping that assigns every $x \in X$ the correct $y \in Y$
• Find $f : X \rightarrow Y$ such that $f(x)$ outputs the true $y \in Y$
• Measure the quality of $f$ by a loss function $\ell(y, f(x))$

Find minimizer of

$$f^* = \min_f \int_{X \times Y} \ell(y, f(x)) dP(x, y)$$

|generalization error|

BUT: $P(x, y)$ is not known in practical applications!
Empirical Risk

Remedy: Evaluate function $f$ on a sample drawn independently from $P(x, y)$

- Given: a sample $(x_1, y_1), \ldots, (x_n, y_n) \in X \times Y$

Find $f$ that minimizes the empirical risk

$$\hat{f}^* = \min_f \sum_{i=1}^n \ell(y_i, f(x_i))$$

BUT: Not a well-defined optimization problem

- Might not be convex (depends on $\ell$)!
- There are generally $\infty$-many indistinguishable solutions
Regularized Empirical Risk Minimization

• Given: a sample \((x_1, y_1), \ldots, (x_n, y_n) \in X \times Y\), loss \(\ell\)

Find \(f\) that minimizes the regularized empirical risk

\[
\hat{f}^* = \min_f \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda \underbrace{\Omega(f)}_{\text{regularization}}
\]

Additional term \(\Omega(f)\)...

• Introduces a bias (e.g., towards simple functions \(f\))
• Turns minimization into well-defined optimization problem
• (Unique solution!)
Representer Theorem

\[ \hat{f}^* = \min_f \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda \Omega(f) \]

Representer Theorem: (mild assumptions on \( \ell \) and \( \Omega \))

- Optimal solution \( \hat{f}^* \) can always be written as

\[ \hat{f}^* (x) = \sum_{i=1}^{n} \alpha_i \langle x_i, x \rangle \]

- Data appears only within inner product!
- Encapulate data by a kernel function \( K = \langle \phi(x), \phi(x') \rangle \)
- \( \phi \) is a possibly non-linear mapping

\( \Rightarrow \) Model (and thus also algorithm) is independent of actual data!

\[ f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) \]
Semi-supervised Learning
Semi-supervised Learning

Given labeled and unlabeled examples
  • e.g., \((x_1, y_1), \ldots, (x_n, y_n)\) and \(x_{n+1}, \ldots, x_{n+m}\)

Frequently: assume cluster structure in the data
  • e.g., transductive and graph-based approaches

BUT: What if the data does not meet this assumption?
  • e.g., regression
Co-learning (=Multi-view Learning)

Assumption: independent sets of features (=views)
- Example for 2-view learning: Web page classification
  - e.g., View $V_1$ content and view $V_2$ context of a web page

Learn a classifier $h_j$ in each view
Classifiers exchange predictions on unlabeled examples
Goal: Hypotheses shall agree on labeling of unlabeled examples
Why does it work?

Two independent hypotheses $h_1$ and $h_2$.
Rather weak assumption: $p(h_j(x) \neq y) \leq \frac{1}{2}$

Disagreement upper bounds error on unlabeled examples!

Brefeld & Scheffer (ICML 2004)
Co-regularized Least Squares Regression

Given:

- Sets of labeled examples \( L = \{x_1, \ldots, x_n\} \) and target function \( y(x) \in \mathbb{R} \)
- Set of unlabeled examples \( U = \{x_{n+1}, \ldots, x_{n+m}\} \)
- Model in view \( v \): \( f^v(x) = \langle w^v, \phi^v(x) \rangle \)

Optimization problem:

\[
Q(f) = \sum_{v=1}^{V} \left[ \sum_{x \in L_v} (y(x) - f_v(x))^2 + \eta \| f_v(\cdot) \|^2 \right] + \lambda \sum_{u,v=1}^{V} \sum_{x \in U} (f_u(x) - f_v(x))^2
\]

Solution of the form:

\[
M = L\alpha \quad \Rightarrow \quad \alpha = LM^{-1}
\]

BUT: matrix inversion needs cubic time in \((n + m)!\)

Brefeld, Gaertner, Scheffer & Wrobel (ICML 2006)
Approximate Co-regularized Least Squares Regression

Approximation:

\[
f_{opt}^v = \sum_{x \in L^v \cup U} \alpha^v(x) k^v(x, \cdot) \approx \sum_{x \in L^v} \alpha^v(x) k^v(x, \cdot)
\]

Approximate solution can be found in time \(O(V^3 n^2 m)\) (assuming \(m \geq n\)).

Table: rmse \times 10 values for KDD Cup 98 data with 100 labeled instances

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>1k</th>
<th>10k</th>
<th>50k</th>
<th>90k</th>
</tr>
</thead>
<tbody>
<tr>
<td>rlslr</td>
<td>4.21 ± 0.46</td>
<td>4.21 ± 0.46</td>
<td>4.21 ± .46</td>
<td>4.21 ± 0.46</td>
<td>4.21 ± 0.46</td>
</tr>
<tr>
<td>exact</td>
<td>2.20 ± 0.27</td>
<td>1.58 ± 0.12</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>approx</td>
<td>3.36 ± 0.89</td>
<td>2.29 ± 0.37</td>
<td><strong>1.31 ± 0.06</strong></td>
<td><strong>1.07 ± 0.04</strong></td>
<td><strong>1.25 ± 0.06</strong></td>
</tr>
</tbody>
</table>
Co-SVMs for Predicting Complex Outputs

- Input $x$ and output $y$ are now structured and interdependent
- Prediction of peer view $y_i^{\bar{v}}$
- Confidence of peer view $\gamma_i^{\bar{v}}$

$$\min_{w^v, \xi^v} \frac{1}{2} \|w^v\|^2 + \frac{C}{r} \left( \sum_{i=1}^{n} (\xi_i^v)^r + C_u \sum_{i=n+1}^{n+m} (\min\{\gamma_i^{\bar{v}}, 1\})(\xi_i^v)^r \right)$$

$$\forall_{i=1}^{n}, \forall_{\tilde{y} \neq y_i} \langle w^v, \Phi^v(x_i, y_i) - \Phi^v(x_i, \tilde{y}) \rangle \geq 1 - \frac{\xi_i^v}{\sqrt{\Delta(y_i, \tilde{y})}}$$

$$\forall_{i=n+1}^{n+m}, \forall_{\tilde{y} \neq y_i^{\bar{v}}} \langle w^v, \Phi^v(x_i, y_i^{\bar{v}}) - \Phi^v(x_i, \tilde{y}) \rangle \geq 1 - \frac{\xi_i^{\bar{v}}}{\sqrt{\Delta(y_i^{\bar{v}}, \tilde{y})}}$$

$$\forall_{i=1}^{n+m} \xi_i^{v} \geq 0.$$

Brefeld & Scheffer (ECML 2005), Brefeld & Scheffer (ICML 2006)
Natural Language Parsing

WSJ (Penn Treebank)
- Subsets 2-21, 8,666 sentences up to 15 tokens
- CFG contains 4,800 distinct production rules

Negra
- 14,137 sentences of between 5–25 tokens
- CFG contains more than 26,700 production rules

<table>
<thead>
<tr>
<th></th>
<th>L:4</th>
<th></th>
<th>L:40</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U:0</td>
<td>U:80</td>
<td>U:0</td>
<td>U:80</td>
<td>U:200</td>
<td></td>
</tr>
<tr>
<td>WSJ</td>
<td>SVM</td>
<td>45.40 ± 0.61</td>
<td>-</td>
<td>71.73 ± 0.29</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coSVM</td>
<td>47.92 ± 0.59</td>
<td>48.23 ± 0.55</td>
<td>73.85 ± 0.24</td>
<td>74.07 ± 0.25</td>
<td>75.01 ± 0.31</td>
</tr>
<tr>
<td>NEG</td>
<td>SVM</td>
<td>47.58 ± 0.37</td>
<td>-</td>
<td>63.70 ± 0.29</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coSVM</td>
<td>48.81 ± 0.37</td>
<td>49.46 ± 0.33</td>
<td>64.94 ± 0.27</td>
<td>65.13 ± 0.25</td>
<td>65.70 ± 0.25</td>
</tr>
</tbody>
</table>
Multiple Kernel Learning
Data Representation

\[ k_{RBF}(x, x') = \exp \left( -\frac{\|x - x'\|^2}{2\gamma} \right) \]

What is a good value for \(\gamma\)? Which kernel represents the data well? Sophisticated guesses, trial and error, cross validation, ... Can we do better?
Multiple Kernel Learning (MKL)

Given $m$ different feature mappings $\psi_1, \ldots, \psi_m$ (or kernels $K_1, \ldots, K_m$)
MKL: find the optimal linear mixture of the mappings:

$$K = \sum_{j=1}^{m} \beta_j K_j$$

The model is given by:

$$f(x) = \sum_{j=1}^{m} \beta_j w_j^\prime \psi_j(x) + b = w_\beta^\prime \psi_\beta(x) + b,$$

where the components have a block structure, i.e.,

- $w_\beta = (\sqrt{\beta_j} w_j)_{k=1,\ldots,m}$
- $\psi_\beta(x_i) = (\sqrt{\beta_j} \psi_j(x_i))_{j=1,\ldots,m}$
- mixing coefficients $\beta_j \geq 0$
1-norm MKL

Standard approaches to MKL:

- 1-norm constraint on the mixing coefficients
- results in sparse mixtures
- interpretable solutions

\[
\min_{\beta, w, b, \xi} \frac{1}{2} w' \beta w + \eta \|\xi\|_1 \\
\text{s.t. } \forall i : y_i (w'_\beta \psi_\beta(x_i) + b) \geq 1 - \xi_i; \quad \xi \geq 0; \quad \beta \geq 0; \quad \|\beta\|_1 = 1
\]

- BUT: sparse mixtures hardly perform well in practice
- Often outperformed by unweighted-sum kernel \( K = \sum_{j=1}^m K_j \)
Our Contribution: Non-sparse MKL

Standard approaches to MKL:

- Sacrifice interpretability for accuracy
- Allow for arbitrary $p$-norm mixtures ($1 \leq p \leq \infty$)

$$\min_{\beta,w,b,\xi} \frac{1}{2} w' \beta w + \eta \|\xi\|_1$$

s.t. \quad \forall i : y_i (w'_\beta \psi_\beta(x_i) + b) \geq 1 - \xi_i; \quad \xi \geq 0; \quad \beta \geq 0; \quad \|\beta\|_p = 1

Math looks pretty straight forward but it isn’t that simple...

- e.g., no longer a convex problem (but tight relaxation exists)
- translate into equivalent convex min-max problem (cannot be optimized efficiently)

Kloft, Brefeld, Sonnenburg, Zien (NIPS, 2009)
Optimization

We finally arrive at the following semi-infinite program (SIP)

$$\min_{\theta, \beta} \theta \quad \text{s.t.} \quad \theta \geq \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \sum_{m=1}^{M} \beta_m y K_m y \alpha$$

for $\alpha \in \mathbb{R}^n$ with $0 \leq \alpha \leq \eta \mathbf{1}$, and $y^T \alpha = 0$ as well as $\|\beta\|_p \leq 1$ and $\beta \geq 0$.

Alternating optimization scheme:

1: ($\alpha$-step) Solve vanilla SVM with actual mixture $K = \sum \beta_j K_j$
   - Solution is a new constraint for SIP
2: ($\beta$-step) Solve SIP wrt active constraints
   - Solution is a new mixing $\beta$
   - Optimization wrt $p$-norm constraint, for $p = 1, 2$ easy
   - For $p \neq 1, 2$ approximate:

$$\|\beta\|_p^p \approx 1 + \frac{p(p-3)}{2} \sum_{m=1}^{M} p(p-2)(\bar{\theta}_m)^{p-1} \theta_m + \frac{p(p-1)}{2} \sum_{m=1}^{M} \bar{\theta}_m^{p-2} \theta_m^2$$
Gene Start Recognition

- Detect transcription start sites (= the interesting parts of genes)
- 5 different kernels representing
  - the TSS signal (weighted degree with shift)
  - the promoter (spectrum)
  - the 1st exon (spectrum)
  - angles (linear)
  - and energies (linear)
Network Intrusion Detection
Network Intrusion Detection

Machine Learning approach:
- Learn a concise description of normal data
- Assumption: Malicious payload deviates from normal byte streams
- Intrusion detection = anomaly detection
Support Vector Domain Description (SVDD)
- Compute minimal enclosing sphere with center $c$ and radius $R$

Anomaly score of payload $x$:
- Distance to center $c$, that is $f(x) = \|\phi(x) - c\|$%

Intrusion detection:
- Accept payload $x$ if $f(x) \leq R$ and ...
- ... reject $x$ if $f(x) > R$
Support Vector Domain Description (SVDD)

- Compute minimal enclosing sphere with center $c$ and radius $R$

Drawbacks:
- Center $c$ and radius $R$ depend only on the location of the data points
- Experts can hardly influence with the adaptation process
- Resulting classifications may not be interpretable
- Prior knowledge cannot be included in the training
Contributions

Active learning for anomaly detection

- Exploit prior and expert knowledge
- Uncertain guesses can be verified by expert
- Learner guides the expert in the classification process

Techn./Emp. Contributions

- Generalized and semi-supervised SVDD
- Devised an effective active learning strategy
- Empirical evaluation on real network data

Goernitz, Kloft & Brefeld (ECML 2009)
Goernitz, Kloft, Rieck & Brefeld (AISEC 2009)
Semi-supervised generalization of SVDDs

- Allows for the inclusion of unlabeled and labeled data
- Parameters: center $c$, radius $R$ and confidence $\gamma$
- Continuous and differentiable objective
- Optimization with conjugate gradients

Constrained optimization problem:

$$\min_{R, \gamma, c, \xi} R^2 - \kappa \gamma + \eta_u \sum_{i=1}^{n} \xi_i + \eta_l \sum_{j=n+1}^{n+m} \xi_j$$

subject to:

$$\forall_{i=1}^{n} : \|\phi(x_i) - c\|^2 \leq R^2 + \xi_i$$

$$\forall_{j=n+1}^{n+m} : y_j \left(\|\phi(x_j) - c\|^2 - R^2\right) \leq -\gamma + \xi_j$$

$$\forall_{i=1}^{n} : \xi_i \geq 0,$$

$$\forall_{j=n+1}^{n+m} : \xi_j \geq 0.$$
ActiveSVDD - New Problem Formulation & Summary

- Non-convex optimization problem, dual optimization is prohibitive
- Unconstrained, continuous objective:

\[
\min_{R, \gamma, c} P = \min_{R, \gamma, c} \left[ R^2 - \kappa \gamma + \eta_u \sum_{i=1}^{n} \ell_{0,\epsilon} \left( R^2 - \| \phi(x_i) - c \| ^2 \right) \right]
\]

\[
+ \eta_l \sum_{j=n+1}^{n+m} \ell_{0,\epsilon} \left( y_j ( R^2 - \| \phi(x_j) - c \| ^2 ) - \gamma \right)
\]

- Efficiently solvable using gradient-based techniques
- Use Representer Theorem to obtain a non-linear version:

\[
c = \sum_{i} \alpha_i \phi(x_i) + \sum_{j} \alpha_j y_j \phi(x_j)
\]
Active Learning

- Which points to query?
Active Learning - The Good Old Margin Strategy

- Query points lying close to the decision boundary to reduce uncertainty

\[ x' = \arg\min_{x_i \in \{x_1, \ldots, x_n\}} \frac{|R^2 - \|\phi(x_i) - c\|^2|}{\max_i |R^2 - \|\phi(x_i) - c\|^2|} =: \lambda_1(x_i) \]
Active Learning - Find novel Attack Classes

- Use the available information to find clusters of unknown attacks
- (via labeling of $k$ nearest neighbors)

$$x' = \arg\min_{x_t \in \{x_1, \ldots, x_n\}} \frac{\sum_{i=1}^{n} a_{it} + \sum_{j=n+1}^{n+m} y_j a_{jt}}{2k} =: \lambda_2(x_i)$$
• Combine both strategies to find anomalies with high confidence
• Trade-off parameter $\tau \in [0, 1]$

$$x' = \arg\min_{x_t \in \{x_1, \ldots, x_n\}} \tau \lambda_1(x_t) + (1 - \tau) \lambda_2(x_t)$$
Empirical Results

Data

Recorded within 10 Days at Fraunhofer FIRST Institute
145,069 normal HTTP Connections, mean length 489 bytes
27 real Attack classes with 2 – 6 Instances each (Metasploit)
Obfuscate attacks:

\[
\text{attack} = \text{randomly chosen normal HTTP-Header} + \text{malicious payload}
\]

Effect: feature representation is close to that of normal traffic

Setup

3-gram representation of bytestream
Training (966+34), Holdout (795+27), and Test (795+27)
Attacks of same class occur either in train or test set
10 repetitions, AUC in the false positive interval [0, 0.01]
Normal vs. Mimicry Attacks

- ActiveSVDD outperforms baselines significantly
- ActiveSVDD with only 3% labeled data (comb. strategy) achieves almost perfect separation!
ActiveSVDD effectively detects unknown attack classes
Summary

Semi-supervised Learning
- Co-learning for regression and structured output prediction
- Agreement between classifiers

Multiple Kernel Learning
- Sparse $\Rightarrow$ non-sparse MKL
- Choose optimal parameters and kernel mixture simultaneously

Network Intrusion Detection
- Active learning for anomaly detection
- Results support a real-world deployment