Probabilistic Graphical Models for Semi-Supervised Traffic Classification

Rotsos Charalampos, Jurgen Van Gael, Andrew W. Moore, Zoubin Ghahramani

Computer Laboratory and Engineering Department, University of Cambridge
Traffic classification

- Traffic classification is the problem of defining the application class of a network flow by inspecting its packets.

- port-based $\rightarrow$ pattern match $\rightarrow$ statistical analysis.

- Useful in order to perform other network functions:
  - **Security**: Fine grain access control, valuable dimension for analysis
  - **Network Management**: network planning, QoS
  - **Performance measurement**: Performance dependence on traffic class
Problem Space

- So far research focuses on packet-level measurement with good results.
- But no systems implementations, because…
  - Required measurements are difficult
    - Focus on flow records.
    - Existing research exhibit encouraging results.
  - Inflexible and generic models
    - use modern ML techniques (Bayesian Modeling, Probabilistic graphical models)
    - Develop a problem specific ML-model with well defined parameters
    - Since records are sensitive to minor network changes, use semi-supervised learning
Outline

• Model Presentation
• Results
• Related work
• Further Development
Problem definition

- $N$ flows extracted from a router each having $M$ feature.
- Each flow is represented by a vector $x_i$ that has set of features $x_{ij}$ with $0 < j \leq M$ and $0 < i \leq N$.
- Each flow has an application class $c_i$.
- Assume that we have $L$ flows labeled and $U$ flow unlabeled with $L + U = N$.
- Define $f(.)$ such as, if $X_i \in U$, $f(X_i | C_{L}, L) = c_i$.
- Assume that flow records are generated without any sampling applied and $x_{ij}$ are independent.
Probabilistic Graphical Models

- Diagrammatic representations of probability distributions
- Directed acyclic graphs represent conditional dependence among R.V.
- Easy to perform inference
  \[ p(x) = \prod_{v \in V} p(x_v \mid x_{pa(v)}) \]
  \[ P(a,b,c) = P(a) \ P(b \mid a) \ P(c \mid a,b) \]
- Simple graph manipulation can give us complex distributions.
- Advantages:
  - Modularity
  - Iterative design
  - Unifying framework
• $\varphi$ is the parameter of the class distribution and $\theta_{kj}$ is the parameter of the distribution of feature $j$ for class $k$.
• Graph model similar to supervised Naïve Bayes Model.
• Assume $\theta_{kj} \sim \text{Dir}(\alpha_\theta)$ and $\varphi \sim \text{Dir}(\alpha_\varphi)$.
• Use bayesian approach to calculate parameter distribution.
Semi supervised learning

- Hybrid approach of supervised and unsupervised learning
- Train using a labeled dataset and extend model by integrating newly labelled datapoints.

- Advantages:
  - Reduced training dataset.
  - Increased accuracy when the model is correct.
  - Highly configurable when used with Bayesian modeling.

- Disadvantages
  - Computationally complex.
• Calculating parameter increases exponentially as new unlabeled datapoint are added.

\[ p(\phi, \theta|X, Y, C_X) \propto \sum_{C_Y} p(C_X, X, C_Y, Y|\phi, \theta)p(\phi)p(\theta) \]

• **Hard assignment**: Add newly labelled datapoint to the Cx with the highest posterior probability.

• **Soft assignment**: Update the posterior for each parameter according to the predicted weight of the datapoint.

• Define class using:

\[ f(x^*) = \max_c(p(c|x^*)) \]
Outline

• Model Presentation
• **Results**
• Related work
• Further Development
Data

- 2 day trace from research facility [Li09]. Appr. 6 million tcp flows.
- Ground-truth using GTVS tool.
- Netflow records exported using nProbe. Settings similar to a Tier-1 ISP.
- Model implemented in C#. Also used the Naïve Bayes with kernel estimation implementation from the Weka Platform.

Feature set:

<table>
<thead>
<tr>
<th>srcIP/dstIP</th>
<th>srcPort/dstPort</th>
<th>ip tos</th>
<th>start/end time</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpFlags</td>
<td>bytes</td>
<td># packets</td>
<td>time length</td>
</tr>
<tr>
<td>avg. packet size</td>
<td>byte rate</td>
<td>packet rate</td>
<td>tcpF* (uniq. flag)</td>
</tr>
</tbody>
</table>
## Application statistics

<table>
<thead>
<tr>
<th>App</th>
<th>%</th>
<th>App</th>
<th>%</th>
<th>App</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>4.3</td>
<td>services</td>
<td>0.03</td>
<td>peer-to-peer</td>
<td>11.47</td>
</tr>
<tr>
<td>mail</td>
<td>2.5</td>
<td>Spam filter</td>
<td>0.48</td>
<td>web</td>
<td>72.33</td>
</tr>
<tr>
<td>ftp</td>
<td>6.25</td>
<td>streaming</td>
<td>0.31</td>
<td>vpn</td>
<td>0.1</td>
</tr>
<tr>
<td>im</td>
<td>0.6</td>
<td>voip</td>
<td>0.16</td>
<td>Remote access</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Baseline comparison

![Baseline comparison graph]

- Accuracy vs Time of Day
- Legend: hard, hard smallSet, NBK, soft, soft smallSet

## Baseline comparison – Class accuracy

<table>
<thead>
<tr>
<th></th>
<th>DB</th>
<th>MAIL</th>
<th>FTP</th>
<th>IM</th>
<th>P2P</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard</td>
<td>1</td>
<td>0.58</td>
<td>1</td>
<td>0.39</td>
<td>1</td>
<td>0.95</td>
</tr>
<tr>
<td>Hard-ss</td>
<td>1</td>
<td>0.59</td>
<td>1</td>
<td>0.82</td>
<td>1</td>
<td>0.77</td>
</tr>
<tr>
<td>Soft</td>
<td>1</td>
<td>0.55</td>
<td>1</td>
<td>0.42</td>
<td>1</td>
<td>0.96</td>
</tr>
<tr>
<td>Soft-ss</td>
<td>1</td>
<td>0.61</td>
<td>1</td>
<td>0.42</td>
<td>1</td>
<td>0.81</td>
</tr>
<tr>
<td>NBK</td>
<td>0.84</td>
<td>0.26</td>
<td>0.42</td>
<td>0.76</td>
<td>0.91</td>
<td>0.11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SERV</th>
<th>SPAM</th>
<th>STREAM</th>
<th>WEB</th>
<th>VPN</th>
<th>VOIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard</td>
<td>0</td>
<td>1</td>
<td>0.97</td>
<td>0.99</td>
<td>0.82</td>
<td>0.24</td>
</tr>
<tr>
<td>Hard-ss</td>
<td>0</td>
<td>1</td>
<td>0.91</td>
<td>0.99</td>
<td>0</td>
<td>0.44</td>
</tr>
<tr>
<td>Soft</td>
<td>0</td>
<td>1</td>
<td>0.96</td>
<td>0.99</td>
<td>1</td>
<td>0.77</td>
</tr>
<tr>
<td>Soft-ss</td>
<td>0</td>
<td>1</td>
<td>0.96</td>
<td>0.99</td>
<td>0.03</td>
<td>0.21</td>
</tr>
<tr>
<td>NBK</td>
<td>0.24</td>
<td>0.95</td>
<td>0.1</td>
<td>0.89</td>
<td>0.35</td>
<td>0.12</td>
</tr>
</tbody>
</table>
Dataset size

The graph shows the training dataset size over time of day. Different lines represent different datasets categorized by hardness and size:

- hard-30min
- hard-10000
- hard-20000
- NBK-30min
- soft-30min
- soft-10000
- soft-20000

The x-axis represents the time of day, while the y-axis shows accuracy. The graph indicates fluctuations in accuracy throughout the day for each dataset category.
Model parameters

![Graph showing model parameters for mail and web]

Accuracy vs. time of day for different models.

- **Mail:**
  - hard-20000
  - hard-1

- **Web:**
  - hard-20000
  - hard-1
Outline

- Model Presentation
- Results
- Related work
- Further Development
Related work

- Lots of work on traffic classification using machine learning
  - Survey paper [Ngyen et al, IEEE CST 2008] and method comparison [Kim et al, Connext08]
  - Semi-supervised learning used on packet-level measurements in [Erman et al, Sigmetrics07]
  - Traffic classification using NetFlow data is quite recent
    - First attempt using a Naïve Bayes classifier introduced in [Jiang et al, INM07]
    - Approach to the problem using C4.5 classifier in [Carela-Espanol et al, Technical report 09]
Outline

• Model Presentation
• Results
• Related work
• Further Development
Further development

- Packet sampling
  - Difficult problem – multi view points could simplify the problem
- Adapt model for host characterization problem
  - Aggregate traffic on the host level and enrich data dimensions
- Incorporate graph level information in the model
  - Computer networks bares similarities with social networks
Conclusion

- Flow records may be a good data primitive for traffic classification.
- Modeling using probabilistic graphical model is not very difficult.
- Semi supervised learning is an effective concept, but is not a one-solves-all solution.
- Our model achieves 5-10% better performance than generic classifier and exhibits a good stability in short scale.
- Bayesian modeling and graphical models allow easy integration of domain knowledge and adaptation to the requirements of the user.
- Model can be extended to achieve better results.

Thank you!!!!