Robust Artificial Intelligence

Reading Seminar; Tsinghua University
Thomas G. Dietterich, Oregon State University
tgd@cs.orst.edu
Lecture 2: Rejection

• Given:
  • Training data \((x_1, y_1), \ldots, (x_N, y_N)\)
  • Target accuracy level \(1 - \epsilon\)
  • Learn a classifier \(f\) and a rejection rule \(r\)

• At run time
  • Given query \(x_q\)
  • If \(r(x_q) < 0\), REJECT
  • Else classify \(f(x_q)\)
Papers for Today

• Cortes, C., DeSalvo, G., & Mohri, M. (2016). Learning with rejection. *Lecture Notes in Artificial Intelligence, 9925 LNAI*, 67–82. [http://doi.org/10.1007/978-3-319-46379-7_5](http://doi.org/10.1007/978-3-319-46379-7_5)


Basic Theory

• Suppose $f^*(x, y) = P(y|x)$ is the optimal probabilistic classifier
• Best prediction is $\hat{y} = \arg\max_y f^*(x, y)$
• Then the optimal rejection rule is to REJECT if $f^*(x, \hat{y}) < 1 - \epsilon$
• (Chow 1970)
Non-Optimal Case

- If $f$ is not optimal, we can still determine a threshold with performance guarantees

- Let $(f(x_i, \hat{y}_i), I[\hat{y}_i = y_i])$ be a set of calibration data points $i = 1, \ldots, N$

- Sort them by $\hat{p}(\hat{y}_i | x_i) = f(x_i, \hat{y}_i)$

- Choose the smallest threshold $\tau$ such that if $f(x_i, \hat{y}_i) > \tau$ then the fraction of correct predictions is $1 - \epsilon$
Finite Sample (PAC) Guarantee

- $P \left( \sqrt{n} \sup_x |\hat{F}_n(x) - F(x)| > \lambda \right) \leq 2 \exp(-2\lambda^2)$ Massart (1990)
- Set $x := \tau$
- $P \left( \eta > \frac{\lambda}{\sqrt{n}} \right) = 2 \exp(-2\lambda^2)$
- Set $\frac{\lambda}{\sqrt{n}} = \eta$ and $\delta = 2 \exp(-2\lambda^2)$; solve for $n$
- $\lambda = \eta \sqrt{n}$
- $\delta = 2 \exp(-2\eta^2 n)$
- $\log_{\frac{2}{\delta}} = -\eta^2 n$
- $n = \frac{1}{\eta^2} \log \frac{2}{\delta}$
- If $n > \frac{1}{\eta^2} \log \frac{2}{\delta}$ then w.p. $1 - \delta$, the true error rate will be bounded by $1 - (\epsilon + \eta)$
Related Work

• Geifman & El Yaniv (2017)
  • Develop confidence scores based on either the softmax ("SR") or Monte Carlo dropout ("MC-dropout")
  • Binary search for the threshold
  • Use an exact Binomial confidence interval instead of Massart’s bound
  • Union bound over the binary search queries
Cost-Sensitive Rejection

- **Cost Matrix**

- **Optimal Classifier**
  - For \( \hat{p}(y = 1|x) \geq \tau_1 \), predict 1
  - For \( \hat{p}(y = 2|x) \geq \tau_2 \), predict 2
  - Else REJECT

- Search all pairs \((\tau_1, \tau_2)\) to minimize expected cost

- Pietraszek (2005) provides a fast algorithm based on (a) isotonic regression and (b) computing the slopes on the ROC curve corresponding to \(\tau_1\) and \(\tau_2\)

<table>
<thead>
<tr>
<th>Actions</th>
<th>Predict 1</th>
<th>Predict 2</th>
<th>Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(y = 1</td>
<td>x))</td>
<td>0</td>
<td>(c_{12})</td>
</tr>
<tr>
<td>(P(y = 2</td>
<td>x))</td>
<td>(c_{21})</td>
<td>0</td>
</tr>
</tbody>
</table>
Support Vector Machines

• Key insight: Maximize the Margin around the Decision Boundary

• Three strategies:
  • Fit standard SVM, then calibrate or threshold
  • Fit a double-hinge loss (DHL) SVM that maximizes margin around the rejection thresholds
  • Fit two separate functions (classifier and rejection function) that maximize margins around the rejection thresholds
Reminder: Standard SVM

• Linear classifier that maximizes the margin between positive and negative examples

• $y \in \{+1, -1\}$ so $y_i f(x_i) > 0$ means $x_i$ is classified correctly

\[
\min_{w,b,\xi} C \|w\|^2 + \sum_i \xi_i \text{ subject to } y_i (w^T x_i + b) + \xi_i \geq 1 \ \forall i
\]

• The $\xi_i$ are “slack variables” the measure how “wrong” we are classifying $x_i$

• $C$ is the regularization parameter
Double Hinge Loss
(Herbei & Wegkamp, 2006; Bartlett & Wegkamp, 2008)

• Assume cost of rejection is $c$
• Reject if $|f(x)| < \delta$
• Loss function $\ell_c(yf(x))$
  • if $yf(x) < -\delta$ $\ell_c = 1$
  • if $yf(x) \in [-\delta, +\delta]$ $\ell_c = c$
  • if $yf(x) > +\delta$ $\ell_c = 0$
• Convex upper bound $\phi_c$
  • if $yf(x) < 0$ $\phi_c = 1 - ayf(x)$
  • if $yf(x) \in [0,1]$ $\phi_c = 1 - yf(x)$
  • if $yf(x) > 1$ $\phi_c = 0$
DHL Optimization Problem

• \( \min_{w,b,\xi,\gamma} \sum_i \xi_i + \frac{1-2c}{c} \gamma_i \) subject to
  
  • \( y_i (w^T x_i + b) + \xi_i \geq 1 \)
  
  • \( y_i (w^T x_i + b) + \gamma_i \geq 0 \)
  
  • \( \xi_i \geq 0; \gamma_i \geq 0 \)
  
  • \( \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \leq r^2 \) (regularization constraint)

• This is a quadratically-constrained quadratic program, so it can be solved, but it is not easy
Non-Optimal Case (2)

• Defining the rejection function in terms of $h$ assumes that the probability of error is monotonically related to $\hat{p}(y|x)$.
• We saw last lecture that this is not necessarily true
• We can try to fix $h$ or we can learn a more complex $r$ function
• Unlikely to be a problem for flexible models, but could be a problem for linear and SVM methods
Method 3: Learn \((f, r)\) pair
(Cortes, DeSalvo & Mohri, 2016)

- Two-dimensional loss function
- If \(r(x) \geq 0\) and \(yf(x) \geq 0\) loss = 0
- If \(r(x) \geq 0\) and \(yf(x) < 0\) loss = 1
- If \(r(x) < 0\) loss = \(c\)
Convex Upper Bound

\[ L_{MH}(r, f, x, y) = \max \left( 1 + \frac{1}{2} (r(x) - yf(x)), c \left( 1 - \frac{1}{1-2c} r(x) \right), 0 \right) \]
CHR Optimization Problem

• \( f(x) = w^\top x + b \)
• \( r(x) = u^\top x + b' \)

• \( \min_{w,u,\xi} \frac{\lambda}{2} \|w\|^2 + \frac{\lambda'}{2} \|u\|^2 + \sum_i \xi_i \) subject to

  • \( c \left( 1 - \frac{1}{1-2c} (u^\top x_i + b') \right) \leq \xi_i \)
  • \( \frac{1}{2} (u^\top x_i + b' - y_i w^\top x_i - b) \leq 1 + \xi_i \)
  • \( 0 \leq \xi_i \)

• By minimizing \( \xi_i \) we are minimize the max of these three terms
Experimental Tests

Total Loss (Misclassification + Rejection)

Data Set: liver, bank, skin, pima, australian, cod, haber

Graph showing the total loss for different datasets with error bars for DHL and CHR.
Note: DHL modified to reject the same number of points as CHR
Reject Option Conclusions

- Basic thresholding is easy and gives PAC guarantees
- 2-class thresholding with differential costs is easy
- $K$-class thresholding?
- Thresholding SVMs is interesting
  - Focus the “margin” on the reject boundaries
  - Learning a $(f, r)$ pair is better than optimizing the double hinge loss
- Open question: How to jointly train DNNs and a rejection function
Conformal Prediction (online version)

• Given:
  • Training data \([z_1, \ldots, z_{n-1}]\) where \(z_i = (x_i, y_i)\)
  • Classifier \(f\) trained on the training data
  • Nonconformity measure \(A_n : \mathcal{Z}^{n-1} \times \mathcal{Z} \mapsto \mathbb{R}\)
  • Query \(x_n\)
  • Accuracy level \(\delta\)

• Find:
  • A set \(C(x_q) \subseteq \{1, \ldots, K\}\) such that \(y_q \in C(x_q)\) with probability \(1 - \delta\)

• Method:
  • For each \(k\), let \(z_n^k = (x_q, k)\)
    • \(\forall i \alpha_i^k := A([z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n], z_i)\) “how different is \(z_i\) from the rest of the \(z\) values?"
    • Let \(p^k = \text{fraction of } [\alpha_1^k, \ldots, \alpha_n^k] \text{ that are } \geq \alpha_n^k\)
  • \(C(x_q) = \{k | p^k \geq \delta\}\)
  • Output \(C(x_q)\)
Examples of Nonconformity Measures

• Conditional probability method:
  • Train a probabilistic classifier $f$ on $[z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n]$  
  • Then compute $A([z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n], z_i) = -\log f(z_i)$

• Nearest neighbor nonconformity
  • $A(B, z) = \frac{\text{distance to nearest } z' \in B \text{ in same class}}{\text{distance to nearest } z' \in B \text{ in different class}}$
Additional Information

• In addition to outputting $C(x_q)$, we can output
  • $\hat{y}_q = \arg\max_k p^k$ (the best prediction)
  • $p_q = \max_k p^k$ (the p-value of the best prediction)
  • $1 - \max_{k; k \neq \hat{y}_q} p^k$ (the “confidence”. We have more confidence if the second-best p-value is small)
Batch ("inductive") Conformal Prediction

• Divide data into training and calibration
• Train $f$ on the training data
• Let $Z_1, \ldots, Z_n$ be the validation data
• Let $\alpha_1, \ldots, \alpha_n$ be the non-conformity scores of the validation data
  • $\alpha_i := A(\{Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_n\}, Z_i)$
• Given query $x_q$
  • For $k = 1, \ldots, K$
    • Let $Z_q^k = (x_q, k)$
    • Let $\alpha_q^k = A(\{Z_1, \ldots, Z_n\}, Z_q^k)$
    • Let $p^k = \text{fraction of } \{\alpha_1, \ldots, \alpha_n, \alpha_q^k\} \text{ that are } \geq \alpha_q^k$
  • $C(x_q) = \{k \mid p^k \geq \delta\}$
• Key difference: $Z_q^k$ does not affect the other non-conformity scores
Almost Equivalent to Learning a Threshold

• Let $\tau = \delta$ quantile of $[\alpha_1, ..., \alpha_n]$  

• Given query $x_q$
  
  • For $k = 1, ..., K$
    
    • Let $z^k_q = (x_q, k)$
    
    • let $\alpha^k_q = A([z_1, ..., z_n], z^k_q)$
  
  • $C(x_q) = \{k | \alpha^k_q \geq \tau\}$

• Additional difference: $\tau$ is computed without considering $\alpha^k_q$

• IF $n$ is large enough, this does not matter
Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Satellite</th>
<th>Shuttle</th>
<th>Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Units</td>
<td>23</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>Hidden Learning Rate</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>Output Learning Rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Momentum Rate</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- **Non-conformity Measures**
  - **Resubstitution:**
    - Train $f$ on all data
    - Let $\hat{y}_i = f(x_i)$
    - $A(\{z_1, ..., z_{i-1}, z_i, ..., z_N\}, (x_i, k)) = I[\hat{y}_i = k]$  
  - **Leave One Out:**
    - Train $f$ on $\{z_1, ..., z_{i-1}, z_i, ..., z_N\}$
    - Let $\hat{y}_i = f(x_i)$
    - $A(\{z_1, ..., z_{i-1}, z_i, ..., z_N\}, (x_i, k)) = I[\hat{y}_i = k]$
## Satellite

Table 3. Results of the second mode of the Neural Networks ICP for the Satellite data set.

<table>
<thead>
<tr>
<th>Nonconformity Measure</th>
<th>Confidence Level</th>
<th>Only one Label (%)</th>
<th>More than one label (%)</th>
<th>No Label (%)</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resubstitution</strong></td>
<td>99%</td>
<td>60.72</td>
<td>39.28</td>
<td>0.00</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>84.42</td>
<td>15.58</td>
<td>0.00</td>
<td>4.67</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>96.16</td>
<td>3.02</td>
<td>0.82</td>
<td>9.59</td>
</tr>
<tr>
<td><strong>Leave one out</strong></td>
<td>99%</td>
<td>61.69</td>
<td>38.31</td>
<td>0.00</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>85.70</td>
<td>14.30</td>
<td>0.00</td>
<td>4.86</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>96.11</td>
<td>3.10</td>
<td>0.79</td>
<td>9.43</td>
</tr>
</tbody>
</table>

\[ y_i \notin C(x_i) \]
Shuttle

<table>
<thead>
<tr>
<th>Nonconformity Measure</th>
<th>Confidence Level</th>
<th>Only one Label (%)</th>
<th>More than one label (%)</th>
<th>No Label (%)</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resubstitution</strong></td>
<td>99%</td>
<td>99.23</td>
<td>0.00</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>93.52</td>
<td>0.00</td>
<td>6.48</td>
<td>6.48</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>89.08</td>
<td>0.00</td>
<td>10.92</td>
<td>10.92</td>
</tr>
<tr>
<td><strong>Leave one out</strong></td>
<td>99%</td>
<td>99.30</td>
<td>0.00</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>93.86</td>
<td>0.00</td>
<td>6.14</td>
<td>6.14</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>88.72</td>
<td>0.00</td>
<td>11.28</td>
<td>11.28</td>
</tr>
</tbody>
</table>

Table 4. Results of the second mode of the Neural Networks ICP for the Shuttle data set.
Segmentation

<table>
<thead>
<tr>
<th>Nonconformity Measure</th>
<th>Confidence Level</th>
<th>Only one Label (%)</th>
<th>More than one label (%)</th>
<th>No Label (%)</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resubstitution</strong></td>
<td>99%</td>
<td>90.69</td>
<td>9.31</td>
<td>0.00</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>97.71</td>
<td>1.25</td>
<td>1.04</td>
<td>3.68</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>94.68</td>
<td>0.00</td>
<td>5.32</td>
<td>6.71</td>
</tr>
<tr>
<td><strong>Leave one out</strong></td>
<td>99%</td>
<td>91.73</td>
<td>8.27</td>
<td>0.00</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>97.79</td>
<td>1.21</td>
<td>1.00</td>
<td>3.55</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>94.76</td>
<td>0.00</td>
<td>5.24</td>
<td>6.67</td>
</tr>
</tbody>
</table>

Table 5. Results of the second mode of the Neural Networks ICP for the Segment data set.
Pendigits + Random Forest
(Dietterich, unpublished)

• Train a random forest on half of UCI Training Set
• Use the predicted class probability $P(y = k| x)$ as the (non)conformity score
• Compute $\tau$ values using other half of Training Set
• Compute $C$ on the Test Set
Cumulative Distribution Function for Class “9”

Empirical CDF \( \ell_{10} \)
Pendigits Results

- All $\tau$ values were 0 (for $\epsilon = 0.001$)
- Probability $y \in \Gamma(x) = 0.9997$
- Abstention rate = 0.72
- Sizes of prediction sets $\Gamma$
Simple Thresholding of $\max_k \hat{p}(y = k | x)$
Zoomed In: $\tau = 0.87$ for $\delta = 0.05$
Test Set Results

• Probability of correct classification: 0.9987
• Rejection rate: 33.4%
  • [Conformal prediction was 72%]
Another Use Case: Lexicon Reduction

• US Postal Service Address Reading Task
  • (Madhavanath, Kleinberg, Govindaraju, 1997)

• Two classifiers
  • Method 1: Fast but not always accurate
  • Method 2: Slower but more accurate
    • Can only afford to run on 1/3 of envelopes
    • Faster if it can be focused on a subset of the classes

• Apply conformal prediction using Method 1
  • Eliminate as many classes as possible
  • Apply Method 2 if $|C(x)| > 1$
Summary

• Lecture 1: Calibration

• Lecture 2: Rejection
  • Method 1: Threshold $f$ with single or multiple thresholds
    • Multiple thresholds requires a change in the SVM methodology
  • Method 2: Learn a separate rejection function and threshold it
  • Method 3: Conformal: Use thresholding to construct a confidence set
    • Reject if $|C(x_q)| \neq 1$
    • Can perform “lexicon reduction”
  • In my experience, Conformal Prediction is not good for Rejection, but more experiments are needed
Next Lecture

• All of these methods assume a closed world
• What happens when queries may belong to “alien” classes not observed during training?

• Papers:
Citations

- Cortes, C., DeSalvo, G., & Mohri, M. (2016). Learning with rejection. *Lecture Notes in Artificial Intelligence, 9925 LNAI*, 67–82. [http://doi.org/10.1007/978-3-319-46379-7_5](http://doi.org/10.1007/978-3-319-46379-7_5)