Ill sampled problems:
Active learning,
one-class classification,
ranking

D.M.J. Tax

Contents

- Train and test datasets
- When (not enough) labels are given:
  Active learning
- When data from one class is given:
  One-class classification/Novelty detection
- Error estimation (ROC curves, AUC)
- When class priors are not given/unclear classes:
  Ranking optimization
- Application: Detection of lung diseases
- Conclusions

Standard classification

Train

Test

- Find the correct classifier complexity
- Compare the results on an independent test set

Representative datasets

- Training sets should be representative for the
  problem (for the test set, future data)
- Do not beautify them
- Do not delete outliers, unless similar outliers are
  improbable

How to select a representative train set?
- At random
- Systematically

Select such that the probability density function
  can be reconstructed. (Is that really necessary?)

Active learning

- Assumption: given a large, but finite, unlabeled
  training set $X_u$ or a density function
- Ask labels for a small set of objects (of given size)$X_l$
- Task: design a classifier, or label $X_u$

Selective sampling

$$\hat{f}_u(x) = \hat{f}_n(x|X_u)$$

$$\hat{f}_l(x) = \hat{f}_l(x|X_l)$$

$$\hat{d}(\hat{f}_u(x), \hat{f}_l(x)) = \int |\hat{f}_u(x|X_u) - \hat{f}_l(x|X_l)| \, dx$$

Choose $X_l$ such that $\hat{d}(\hat{f}_u(x), \hat{f}_l(x))$ is minimum (or some other probabilistic distance)
Active learning

- **Exploitation**: Add unlabeled objects close to the classifier to the training set
- **Exploration**: Add remote unlabeled objects that represent unvisited clusters

Active learning: Exploitation

**Uncertainty sampling**

Add objects close to the decision boundary

\[ x_t \rightarrow S(x), S(x) < 0 : \omega_A, S(x) > 0 : \omega_B \]

Classify \( x_u \) and select the most uncertain object

\[ x_u = \arg \min_x \{ |S(x)|, \forall x \in \mathcal{X}_u \setminus \mathcal{X}_l \}, \mathcal{X}_l \leftarrow \mathcal{X}_l \cup \{ x_u \} \]

Large \( \mathcal{X}_u : S(x) \approx 0 \) for many \( x \), improvement classifier dependent. Sometimes it is better to take random objects for which \( |S(x)| < d \)

Active learning: exploration

**Variation in label assignment**

Add objects that may change the classifier most.

Variation in class label

1. Initialize \( \mathcal{X}_l \) and compute \( S(x) \)
2. Select an object \( x' \) from \( \mathcal{X}_u \setminus \mathcal{X}_l \)
3. Add \( x' \) temporarily to \( \mathcal{X}_l \) with all possible labels
4. Compute for each of these labels a classifier
5. Classify \( \mathcal{X}_u \setminus \mathcal{X}_l \) by each of the classifiers
6. Count the number of objects \( N \) that are classified different by the classifiers
7. Repeat 2-6 for all \( x' \) from \( \mathcal{X}_u \setminus \mathcal{X}_l \)
8. Add \( x' \) to \( \mathcal{X}_l \) for which \( N \) is maximum
9. Repeat 2-8 as long as desired

Exploration, good for forgotten classes

- It is better to do some exploration and try to find samples in the ‘forgotten’ cluster, than to exploit the actual classifier by sampling around it.

Semi-supervised learning

- Can better classifiers be designed by using labeled and unlabeled objects simultaneously?
- Possible approaches:
  - Label propagation
  - Clustering
- Application: learn from the test set!
New classification problem

One-class classification

- Normal operating condition (target class) can be sampled well
- Abnormal conditions (outlier class) are rare, or are hard to sample reliably

One-class classification

- Many example target objects are present
- Few (reliable) outlier objects are available

Examples one-class

- Machine condition monitoring, (describe the normal operation condition of a running machine, and distinguish it from all outlier situations)
- Detection problems, (detect if a certain event is happening in a large set of unstructured events: ‘face detection’ in recorded videos)
- Inspection problems, (check industrial objects if they satisfy quality criteria)
- Outlier detection to detect suspicious objects in (supervised) classification problems.

How to detect outliers?

- No definition for ‘outlier’ exists
- ‘An outlier has a large distance to the bulk of the data’
- Define (1) bulk of the data, (2) the distance and (3) what is large

One-class classifiers

- Define a function $f$ that defines the distance to the bulk
- Thresholding $f$ gives the classification
- The threshold $\theta$ determines the error on the target class
Free variables for an OCC

Two essential choices of an OCC are:
1. The complexity of a model
2. The threshold of the given model

Complexity and threshold

Error minimization

<table>
<thead>
<tr>
<th>Estimated label</th>
<th>True label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>target</td>
</tr>
<tr>
<td>target</td>
<td>true positive</td>
</tr>
<tr>
<td>outlier</td>
<td>false negative</td>
</tr>
</tbody>
</table>

- Minimize the false positive and false negative fractions
- The fraction false positive cannot be estimated when no example outliers are present

How to estimate false pos.?

- Find some outliers! Probably not according to the ‘true’ distribution...
- Generate artificial outliers. Uniformly around the target class.
- Minimize the captured volume directly.

Classification error?

- Even if we can find some outlier data, still:
  - class priors will be heavily skewed
  - misclassification costs will greatly differ
  - Standard classification error is sensitive to that, and using that carelessly will give very misleading results.

The ROC curve

- Receiver-Operating Characteristic curve: how does the true positive fraction vary after changing the false positive fraction?
Operating points

- A classifier is one point on the curve: the operating point.

Area under the ROC curve

- Comparing ROC curves is not so simple: for each threshold it is different
- An well-known overall measure is the AUC: Area under the ROC curve
- Integrate uniformly over all thresholds

AUC characteristics

- AUC is a performance measure: 1: the data is separable, 0.5: the data is randomly ordered
- AUC is independent of class priors or misclassification costs
- AUC can be estimated more reliably for small sample sizes
- AUC ignores the particular classification threshold

Free variables for an OCC

- Two essential choices of an OCC are:
  - The complexity of a model: compare different ROC curves
  - The threshold of the given model: compare different operating points on one ROC curve

One-class classifiers

- Density based classifiers: estimate the density of the target class
- Distance based classifiers (reconstruction based): compute the distance to (a model of) the target objects
- Decision function based classifiers: only optimize a closed boundary around the target data

Density based OCC

- Gaussian density, Mixture of Gaussians, Parzen density, k-nearest neighbor density
- Optimize the (log-) likelihood, threshold the density
- Estimating the density in high dimensional feature spaces is hard. Large trainingset is needed.
Distance based OCC

- k-means clustering, Self-organising Map, PCA-subspace, auto-associative neural networks, ...
- Fit a (simple) model $M$ to the data, threshold the reconstruction error $d(x, M(x))$
- choosing the right model is hard

Decision boundary OCC

- Fit only a boundary (inspired by the support vector classifier)
- Instead of a linear decision boundary, a hypersphere around the target class

Support Vector Data Description

<table>
<thead>
<tr>
<th></th>
<th>Support Vector classifier</th>
<th>Support vector DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>hyperplane $w, b$</td>
<td>hypersphere $a, R$</td>
</tr>
<tr>
<td>complexity</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>SVs</td>
<td>objects on the plane</td>
<td>objects on the sphere</td>
</tr>
<tr>
<td>slacks</td>
<td>objects on the wrong side of the plane</td>
<td>objects outside the sphere</td>
</tr>
</tbody>
</table>

SVDD (2)

- Classifiers can be expressed in terms of support vectors
- The classifiers and optimization are in terms of inner product (kernel) instead of features

SVM: $w^T x = \sum_i \alpha_i K(x, x_i)$

SVDD: $||x-a||^2 + K(x, x) - 2 \sum \alpha_i K(x_i, x) + \sum \alpha_i \alpha_j K(x_i, x_j)$

Different kernels

Gaussian kernel seems very fitting for SVDD

Clear classes: Traditional classification

- Objects of the two classes are physically very different: there is ground truth
- Confusion by noisy measurements and poor features
Pattern recognition is also applied to 'unclear' classes.
There is no 'ground truth' difference between the classes.

May 29, 2008

Intermediate question

So, when we can:
• define clear classes,
• define a good performance measure,
• sample a reliable test set (according to the true class distribution and class priors)
there is no problem in estimating the error.

In all other cases we have a problem...

Question:
Can we become more robust against un-representative test data?

or, simpler:
Can we become more robust against class imbalance/ unclear classes / misclassification costs?

Optimize the ordering!

• Avoid optimizing the classifier for one specific decision boundary
• Optimize the ordering of data

May 29, 2008

Area under the ROC curve

• The ROC curve shows the true positive fraction as function of the false positive fraction for varying threshold
• Independent of class priors and misclassification costs
• The AUC is identical to the chance that a random '+'-class object is ranked higher than a random '-'-class object

AUC for linear classifier

• Assume a linear classifier
  \( f(x) = \text{sign}(w^T x + b) \)
• Maximizing the AUC means maximizing the sum:
  \[ \sum_{i=+,-} I(w^T x_{i+} - w^T x_{i-} > 0) = \sum_{i=+,-} I(w^T (x_{i+} - x_{i-}) > 0) \]
• Optimize a linear classifier, AUC-LPM:
  \[ \min \|w\|_1 + C \sum_{k=+,-} \sum_{t=+,-} \xi_{k+t} \]
  s.t. \[ w^T (x_{k+} - x_{k-}) \geq 1 - \xi_{k+t} \]
  \[ \xi_{k+t} \geq 0 \]
• For each incorrectly ordered pair, an error \( \xi_{k+t} \) is introduced
Advantages and disadvantages

- **Advantages**
  - It can deal with heavily imbalanced datasets
  - The linear classifier is optimized for all thresholds simultaneously
  - It can deal with high dimensional feature spaces (it performs an automatic feature selection)
  - AUC optimization is more stable for small sample sizes

- **Disadvantages**
  - Still a decision threshold (operating point) has to be chosen
  - The number of constraints explodes for larger datasets: all combinations of positive and negative objects have to be considered (use a subsampling approach)
  - A suitable value for C has to be chosen
  - The model is linear

Other models

1. Other names that are used are:
   - ordinal regression,
   - learning to rank, ranking,
   - ordered classification
2. Linear model using gradient descent (iterative scheme)
3. Support vector machines that optimize the AUC (this uses the $L_2$-norm instead of the $L_1$-norm, but it makes subsampling of the constraints impossible)
4. There is a boosting algorithm that optimizes the ranking: rank-boost
5. Bayesian approach using Gaussian Processes,
6. ...

Some results

<table>
<thead>
<tr>
<th>classifier</th>
<th>Biomed</th>
<th>Cleveland</th>
<th>Car Imports</th>
<th>Sonar</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>86.5 (14.4)</td>
<td>60.4 (13.8)</td>
<td>47.2 (21.2)</td>
<td>51.8 (32.0)</td>
</tr>
<tr>
<td>QP</td>
<td>90.3 (12.8)</td>
<td>62.5 (13.2)</td>
<td>51.5 (26.8)</td>
<td>56.8 (27.7)</td>
</tr>
<tr>
<td>Parzen</td>
<td>88.4 (11.6)</td>
<td>63.7 (11.8)</td>
<td>59.5 (24.4)</td>
<td>65.9 (27.7)</td>
</tr>
<tr>
<td>Logistic</td>
<td>94.9 (5.7)</td>
<td>90.3 (4.1)</td>
<td>74.8 (18.8)</td>
<td>64.8 (22.0)</td>
</tr>
<tr>
<td>SVM RBF-kernel</td>
<td>93.3 (7.7)</td>
<td>88.6 (5.4)</td>
<td>83.0 (17.9)</td>
<td>80.7 (19.3)</td>
</tr>
<tr>
<td>Rankboost (T=100)</td>
<td>89.3 (15.7)</td>
<td>88.4 (5.4)</td>
<td>88.9 (9.8)</td>
<td>84.9 (15.5)</td>
</tr>
<tr>
<td>AUC-LP subs (1000)</td>
<td>95.0 (5.7)</td>
<td>90.1 (4.6)</td>
<td>86.7 (13.2)</td>
<td>73.2 (21.7)</td>
</tr>
</tbody>
</table>

- For nice, balanced, separable datasets the difference between standard and AUC optimization is small
- For small (low sample size), overlapping, imbalanced datasets, AUC optimization may be preferred
- (Inherent feature selection may also save your life)

Application: lung diseases

- Detect interstitial lung diseases (tuberculosis) in X-ray images

Interstitial lung disease

- Interstitial disease (ID) is often characterized by abnormal textures
- Automatic detection is hindered because normal anatomical structures overlap the textures
- The exact outline of diseased tissue is very hard (examples are hard to give)
- Try to detect the deviations of the normal textures
- Textures are characterized by Gaussian derivative filters in local patches. Also some position features, and rib-presence features are included. In total we have 158 features.

Pixel-based classification
New results ID

- New results more reasonable
- Best results around AUC=0.95
- Supervised classifiers appear to be confused by the overlap between the classes

<table>
<thead>
<tr>
<th>Method</th>
<th>10% ill</th>
<th>50% ill</th>
<th>90% ill</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>70.2 (17.6)</td>
<td>61.0 (18.2)</td>
<td>59.4 (15.4)</td>
</tr>
<tr>
<td>QC</td>
<td>68.6 (18.6)</td>
<td>61.4 (17.1)</td>
<td>59.0 (14.8)</td>
</tr>
<tr>
<td>Average</td>
<td>52.6 (16.4)</td>
<td>52.4 (17.3)</td>
<td>52.3 (17.3)</td>
</tr>
<tr>
<td>Product</td>
<td>53.8 (16.8)</td>
<td>52.2 (16.7)</td>
<td>51.8 (17.1)</td>
</tr>
<tr>
<td>L1-SVM</td>
<td>68.1 (18.8)</td>
<td>61.3 (17.7)</td>
<td>58.9 (15.6)</td>
</tr>
<tr>
<td>AUC-LPC</td>
<td>90.4 (6.7)</td>
<td>93.9 (5.6)</td>
<td>94.1 (6.2)</td>
</tr>
</tbody>
</table>

Results 2-class

- healthy:
- ill:

Conclusions

- When one of the classes cannot be sampled reliably: try one-class classifiers
- Standard classification approach does not suffice (other classifiers, other errors)
- 'Strange' data is directly detected
- Without example outliers the evaluation is still problematic
- Area Under the ROC curve (AUC) is a good evaluation criterion
- There are ways to optimize AUC directly, but you will not get an operating point in that case