Machine Learning Pragmatics

Prepared by TAs
Revised by professors
Agenda

• Overview of what we have learned so far.
• Evaluating classifiers:
  – Which algorithm is the most accurate?
  – How confident we are in the results that we report?
• Advice for experimenting.
• Practical tips for large scale datasets.

Good News!
No heavy math today!
Resources

• Chapters from PP and ESL are provided in the syllabus.

• “A Course in Machine Learning” by Hal Daume II. Available from ciml.info. Chapters 4 and 5.

• Practical notes
  – From scikit-learn.org
  – Academic papers
  – From TAs’ own experience.
OVERVIEW
Supervised Learning

- If labels take categorical values, it is called *classification*.
- Otherwise, it is called *regression*.

\[(x_1, y_1) \quad \ldots \quad (x_n, y_n)\]

\[(x_{n+1}, ?) \quad \ldots \quad (x_{n+m}, ?)\]

Labeled Training Data \hspace{2cm} Predict the labels for the unlabeled test data

- If labels take categorical values, it is called *classification*.
- Otherwise, it is called *regression*. 
If our predictor is based on a parametric model, the algorithm is called parametric,

- Decision Tree, Linear SVM

Otherwise: Non-parametric

- kernelized SVM

Non-parametric does not mean parameter-less. It just means the number of parameters increases when the training dataset is increased.
Generative vs. Discriminative

Examples:

- **Discriminative:**
  - SVM, Linear Regression

- **Generative:**
  - Naïve Bayes.

\[ M_\theta : \begin{cases} P(Y|X) & \text{Discriminative} \\ P(Y, X) & \text{Generative} \end{cases} \]
Typical Learning Procedure

\[ M_\theta \]

\((x_i, y_i)_{1...n}\) \(\rightarrow\) Learning \(\theta\) \(\rightarrow\) \(\hat{\theta}\)

Testing
\[ x_{i}^{test} \overset{\hat{\theta}}{\rightarrow} y_{i}^{test} \]

* Not the case for online learning algorithms.
Cross-validation for hyperparameters

Hyperparameters are not part the model. They control the quality of the model that we learn.

- $M_\theta$
- $(x_i, y_i)_{1...n}$
- Cross-validation $\lambda$
- Learning $\theta$
- $\lambda$ in regularized linear regression
- $k$ in kNN.
- $C$ in SVM
- $x_i^{\text{test}} \rightarrow \hat{y}_i^{\text{test}}$
- Testing

Learning $\theta$
Reminder about Cross-validation

- We split the training data into $S$ equal parts.
- We use each part in turn as a validation dataset and use the others as a training dataset.
- We choose the hyperparameter such that on average, the model performing the best

Make sure you randomly shuffle the dataset before dividing it into fold.
COMPARING CLASSIFIERS
Class Imbalance

• 4 years ago, my advisor asked me to run SVM on Reuters’ dataset.
• Reuters’ dataset: My subtask was binary classification
  – 0.41% Positive labels.
• How did I do:
  – Accuracy: Yaaay 99.59%!!!
• Sad story:
  – All labels were 0s. Why I even used an algorithm 😞
• Questions:
  – Why SVM failed?
  – Seems accuracy doesn’t show the failure of SVM.
Why SVM fails on imbalanced datasets?

Margin violation = 
0.5 x 4 + 0.5 = 2.5
Why SVM fails on imbalanced datasets?

Margin violation = 1
Why SVM fails in imbalanced datasets?

Which decision boundary minimizes the SVM objective function?
What is the cure?

• **Class weighting:** if $\alpha_P$ fraction of samples are positive, reweight as follows:

$$\frac{1}{2} \|w\|_2^2 + \alpha_P \text{Loss}_N + (1 - \alpha_P) \text{Loss}_P.$$

$$= \frac{1}{2} \|w\|_2^2 + \alpha_P C \sum_{i \in N} \max\{0, 1 - y_i (w^T x_i)\}$$

$$+ (1 - \alpha_P) C \sum_{i \in P} \max\{0, 1 - y_i (w^T x_i)\}.$$

• **Balancing the dataset:**
  
  – Down-sample the negative samples,
  
  – Replicate the positive samples.
Cross-validation on Imbalanced Data

If you divide the data randomly into 5 folds, some folds might not have any positive label.

When you divide the data into folds, manually make sure that all folds have positive labels.

This is called *Stratified Sampling*. 
Classification accuracy measures

• When classes are too imbalanced, the 0-1 accuracy can be misleading.
• When one particular class has special importance, we need other types of performance metrics:
  ▪ A disease diagnosis algorithm should not identify healthy patients as infected. *(Low false positive)*
  ▪ At the same time, it should identify most of the truly infected patients. *(High true positive)*
  ▪ How can we define the performance in terms of false positive and true positive rates?
Using this table we can see that the accuracy can be defined as:

\[
Accuracy = \frac{\#TP + \#TN}{n}
\]

\[n\] denotes \# of data points: \[n = \#TP + \#TN + \#FP + \#FN\].

But we are interested in defining new evaluation metrics.
Precision and Recall

Precision = \frac{\#TP}{\#TP + \#FP}.

Meaning: What ratio of positive predictions are truly positive.

Recall = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP + \#FN}.

Meaning: What ratio of true positive samples are truly discovered.

Truth: filled samples (left) are truly positive.
Prediction: inside the circle are identified as positive.
Error regions shown in red.
From Wikimedia
Precision and Recall: Exercise

\[
Precision = \frac{\#TP}{(\#TP + \#FP)}.
\]

\[
Recall = \frac{\#TP}{\#P} = \frac{\#TP}{(\#TP + \#FN)}.
\]

<table>
<thead>
<tr>
<th>True Labels</th>
<th>(\hat{y} = 1)</th>
<th>(\hat{y} = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y = 1)</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>(y = 0)</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

**Exercise 1**
A dataset base contains 100 records on “machine learning”
A search was conducted on “machine learning” and 40 records were retrieved
Of the 40 records, 30 were relevant.

What is precision and recall?
Precision and Recall: Exercise

\[
\text{Precision} = \frac{\#TP}{\#TP + \#FP}.
\]

\[
\text{Recall} = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP + \#FN}.
\]

**Exercise 2**: Apply these metrics to the medical diagnosis example.

- A disease diagnosis algorithm should not identify healthy patients as infected.
- At the same time, it should identify most of the truly infected patients.

**Exercise 3**: In the SVM failure in imbalanced datasets example, which one becomes zero?

**Question**: Is high recall difficult to achieve?

\[
\text{Recall} \uparrow \quad \text{Precision} \downarrow \\
\text{Recall} \downarrow \quad \text{Precision} \uparrow
\]
F Score

• Combines precision and recall to report a single number as performance measure.

• It is the harmonic mean of precision and recall:

\[ F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \]

• Properties:
  – The F₁ score lies between recall and precision.
  – It tends to lie closer to the smaller of the two.
  – High values for the F₁ score are only possible if both the precision and recall are large.
ROC Curves

• Most algorithms produce soft-labels which should be thresholded to obtain 0-1 labels.

\[(x_i, y_i) \rightarrow w^T x_i \rightarrow \text{Soft Labels} \rightarrow \hat{y}_i\]

Data Learning Thresholding

• Example: in logistic regression, we use the sign function to threshold our decision \(w^T x_i\) and convert it to 0-1 predicted label.

• ROC (Receiver operating characteristic) curve: vary the threshold and plot the (TP, FP) rates for each value of the threshold.
ROC Curves

- ROC (Receiver operating characteristic) curve illustrates the performance of a binary classifier system as its discrimination threshold is varied.
AUC Measure

- Area under ROC curve is called AUC.
- **Meaning**: the value of AUC is the probability that an algorithm predicts a higher value for a randomly selected positive sample than negative sample.
- **Questions**:
  - What is the possible range for AUC?
  - Is the diagonal line in the plot important?
  - Can a curve be possibly under this line?

Sensitivity and Specificity

Often used in bio and medical applications

• Sensitivity == Recall:

\[
\text{Sensitivity} = \frac{\#TP}{\#P} = \frac{\#TP}{(\#TP + \#FN)}.
\]

– e.g. the percentage of sick people who are correctly identified as having the disease.

• Specificity: measures the proportion of negatives which are correctly identified as such.

\[
\text{Specificity} = \frac{\#TN}{\#N} = \frac{\#TN}{(\#TN + \#FP)}.
\]

– e.g. the percentage of healthy people who are correctly identified as not having the disease.
Summary of Accuracy Measures

<table>
<thead>
<tr>
<th>True Labels</th>
<th>Our Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y = 1 )</td>
<td>( \hat{y} = 1 )</td>
</tr>
<tr>
<td>( y = 0 )</td>
<td>( \hat{y} = 0 )</td>
</tr>
</tbody>
</table>

- **Precision**
  \[
  \text{Precision} = \frac{\#TP}{\#TP + \#FP}.
  \]

- **Recall**
  \[
  \text{Recall} = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP + \#FN}.
  \]

- **Specificity**
  \[
  \text{Specificity} = \frac{\#TN}{\#N} = \frac{\#TN}{\#TN + \#FP}.
  \]

- **Sensitivity**
  \[
  \text{Sensitivity} = \text{Recall}.
  \]

- **F1 Score**
  \[
  F_1 = 2 \frac{\text{precision} \times \text{recall}}{(\text{precision} + \text{recall})}.
  \]

**ROC curve:** Vary the threshold and plot the (TP, FP) pair for each value of the threshold. Area under this curve is called AUC.

Choice of measure: Problem Dependent!
An Empirical Comparison of Supervised Learning Algorithms

- “Learning methods such as boosting, random forests, bagging, and SVMs achieve excellent performance that would have been difficult to obtain just 15 years ago.”
- “Of the earlier learning methods, feedforward neural nets have the best performance and are competitive with some of the newer methods, ...”

Variance of Algorithms

Two practical scenarios:

• You have tested several algorithms
  – Algorithm A: accuracy 90%,
  – Algorithm B: accuracy 92%,
  – Is B significantly better than A?

• How confident you are that the estimated value is close to the true value?

• We need the concepts of “Significance test” and “Confidence Intervals”.
Confidence Intervals

• The $(1 - \alpha)$-level confidence interval for a parameter $\theta$ is computed based on data $X$ such that:
  $$P[L(X) \leq \theta \leq U(X)] > 1 - \alpha.$$  

• Theoretically, use asymptotic distributions.  
  – Not the goal of this class.

• **Significance test**: we report p-values which are the probability that the observed difference in performance was by chance. (lower the better)
How can we get the confidence Intervals?

- What can we do computationally?
  - Find the standard deviation using these multiple copies

- But we do not have multiple copies of our dataset 😞
Bootstrapping

• Idea: Create multiple copies of the dataset by resampling.

\[
\begin{align*}
(x_1, y_1) \\
(x_1, y_1) \\
(x_2, y_2) \\
(x_4, y_4) \\
. & . \\
\end{align*}
\]

\[
\begin{align*}
(x_2, y_2) \\
(x_3, y_3) \\
(x_3, y_3) \\
. & . \\
\end{align*}
\]

\[
\begin{align*}
(x_1, y_1) \\
(x_3, y_3) \\
(x_4, y_4) \\
(x_4, y_4) \\
. & . \\
\end{align*}
\]

\[\hat{\theta}_1 \quad \hat{\theta}_2 \quad \hat{\theta}_K\]

• Resampling should be with replacement.
  – Data instances can be repeated.
Significance Test with Bootstrapping

Create $K$ datasets by resampling and test on them.

• **P-values:**

$$p \text{ value} = 1 - \frac{\#(\text{acc}_A > \text{acc}_B)}{K}$$

• **Confidence intervals:**

$$\text{mean}({\hat{\theta}_i}) \pm \text{std}({\hat{\theta}_i})$$

as 1-std confidence intervals.
Bootstrapping vs Cross-validation

- Bootstrapping is used to find the confidence intervals by testing our trained algorithm on each resampled test dataset.
- Cross-validation is used for tuning the hyper-parameters by training our algorithm for each folding combination.
McNemar's test

- How can we claim that our algorithm significantly outperforms a competitive algorithm?
- A hypothesis test in the following form:
  \[ \mathcal{H}_0 : p_A = p_B, \]
  \[ \mathcal{H}_1 : p_A \neq p_B. \]
- Apply both algorithms to the test set and obtain the following contingency table:

<table>
<thead>
<tr>
<th></th>
<th>B predicts positive</th>
<th>B predicts negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>A predicts positive</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>A predicts negative</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

- McNemar’s test statistic:
  \[ \chi^2 = \frac{(b - c)^2}{(b + c)}, \]
- When \( b + c > 25 \), the test statistic is distributed according to \( \chi^2 \) distribution with one degrees of freedom.
- If \( \chi^2 \) value is large, we can reject the null hypothesis. The corresponding p-value can be computed, as well.
Perplexity

• How do you measure confidence of an algorithm in prediction?
• We can use perplexity of the distribution \( p(y|x) \) implied by our algorithm.
• Perplexity of a discrete distribution \( p(z) \) is defined as:
\[
2^H(p(z)) = 2^{-\sum_z p(z) \log_2 p(z)}.
\]
smaller the better.
DEBUGGING

CIML 4.8
Debugging Optimization Algorithms

• Check the gradients numerically:

\[ f'(x) = \frac{f(x + h) - f(x)}{h} \text{ as } h \to 0^+. \]

• Compare sub-modules with the existing software packages.

• Monitor convergence.

• If you are maximizing the log-likelihood, do not optimize too much! Only \( O \left( \frac{1}{\sqrt{n}} \right) \) accuracy is enough. (mostly of theoretical insight, \( n \) is the number of instances.)

Do you know why?
Debugging Learning Algorithms

• Design a simple synthetic dataset on which the algorithm should work and make sure that your algorithm works, indeed.
• Visualize on a tiny 2D dataset.
• Try to overfit: if your algorithm cannot overfit, then that is a problem.
  – Sometimes you need to add a fake very informative feature to make sure that your algorithm really works.
• Do not debug your algorithm on large datasets
  – Sub-sample them!