Machine Learning Methods: ML Diagnostics

Zdeněk Žabokrtský, Ondřej Bojar
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague

Tue Mar 27, 2018
Outline

- Motivation for principled analysis.
- First step: Visualization.
- Bias vs. Variance.
- Optimizer vs. Objective function issue.
  - a.k.a. Search error vs. Modelling error.
- Error analysis, Ablative analysis.

Slides based on:
- the Stanford ML Lecture 11 [http://www.youtube.com/watch?v=sQ8T9b-uGVE](http://www.youtube.com/watch?v=sQ8T9b-uGVE)
- [http://scott.fortmann-roe.com/docs/BiasVariance.html](http://scott.fortmann-roe.com/docs/BiasVariance.html)
- All errors are Ondřej’s fault.
Motivation for Principled Analysis

Some ML does not perform sufficiently well. You can consider random improvements:

- Getting more training examples.
- Reduce the set of features.
- Enlarge the set of features.
- Use different features.
- Run the optimizer for some more iterations.
- Choose a different optimization algorithm.
- Use a different regularization term or constant value.
- Try another learning algorithm (SVM).

... some may be fixing problems you don’t have.
Principled Analysis

First figure out what’s going on.

- Overfitting vs. Underfitting?
- Search error vs. Modelling error?
- Complex system: Find the most problematic component.

Trivial but vital:

- Visualize the data. (Plot or view frequent patterns.)
- Start with simple things.
First Step: Visualization

Always visualize your data, you get:

- Formal check of the dataset.
  - Bad characters, bad number format, bad number of columns.
- Sense of ranges.
- Sense of outliers.
- Sense of any disbalance of the datapoints.

- Subsample (at random) to get an image quickly
- Also process the full dataset, for sanity check.
Scatter Plot of Activity – Heart Rate
Box Plot of Activity – Heart Rate
Highly-Dimensional Datasets

- Plot projections to individual axes.
- Project to pairs of axes, all pairs:
A Correlations Plot

- Plot a corrplot, e.g.:
  http://thomas-cokelaer.info/blog/2014/10/corrplot-function-in-python/
Bias vs. Variance

High Variance = Overfitting:
- the model has too many parameters.

High Bias = Underfitting:
- the model is too rigid.
Fit $\sin(x)$ with poly, orders: 0,1,2,3
Fit \( \sin(x) \) with poly, orders: 3, 5, 7
Fit $\sin(x)$ with poly, orders: 7, 9, 10
Fit \( \sin(x) \) with poly, orders: 10, 11, 12
Fit $\sin(x)$ with poly, orders: 12, 13, 14
Bias-Variance Trade-off

\[ Err(x) = E[(Y - \hat{f}(x))^2] \]

Expected error \( Err(x) \) of learning \( \hat{f} \) on fixed test set \( x \) can be decomposed as:

\[
Err(x) = (E\hat{f}(x) - f(x))^2 + E(\hat{f}(x) - E\hat{f}(x))^2 + \sigma_e^2
\]

\[
Err(x) = \text{Bias}^2 + \text{Variance} + \text{Noise}
\]

- **Bias**: how much the average predicted value \( E\hat{f}(x) \) differs from the ideal value \( f(x) \).
- **Variance**: how much a particular prediction \( \hat{f}(x) \) differs from the average prediction \( E\hat{f}(x) \), on average.


Derivation: see slides by Cohen
Bias-Variance Trade-off

Picture from: http://scott.fortmann-roe.com/docs/BiasVariance.html
Bias vs. Variance

High Variance = Overfitting:
- the model has too many parameters.

High Bias = Underfitting:
- the model is too rigid.

Consider:
- What is the effect of each of those on training error?
- Will more training data help?
- Sketch the shape of learning curves for each of those:
  - for the test error.
  - for the training error.
Search vs. Modelling Error

Search Error:
- the optimizer fails to find the best parameters
- ... a problem with the optimizer.

Modelling Error:
- the best parameters do not lead to the best performance.
- ... a problem with the objective function.

Consider:
- Will more iterations help?
- When can two learners help to diagnose the problem?
Complex Systems

Error Analysis:

▶ Compares the best possible vs. current accuracy.
▶ Provide more and more golden truth data as part of the input.
▶ Find the component where the jump in accuracy is the highest.

Ablative Analysis:

▶ Compares some baseline vs. current accuracy.
▶ Switch off more and more components.
▶ Find the component where the loss in accuracy is the highest.
Work for Today and at Home

- We will now assemble all datasets to one place.
- Apply scikit-learn classification modules on all the datasets collected by you and your colleagues.
  - Run at least 4 different scikit classifier:
    - e.g. svm.SVC, KNeighborsClassifier, BernoulliNB, tree.DecisionTreeClassifier, LogisticRegression, ...
- Recommended:
  - numpy.loadtxt
  - from sklearn.feature_extraction import DictVectorizer