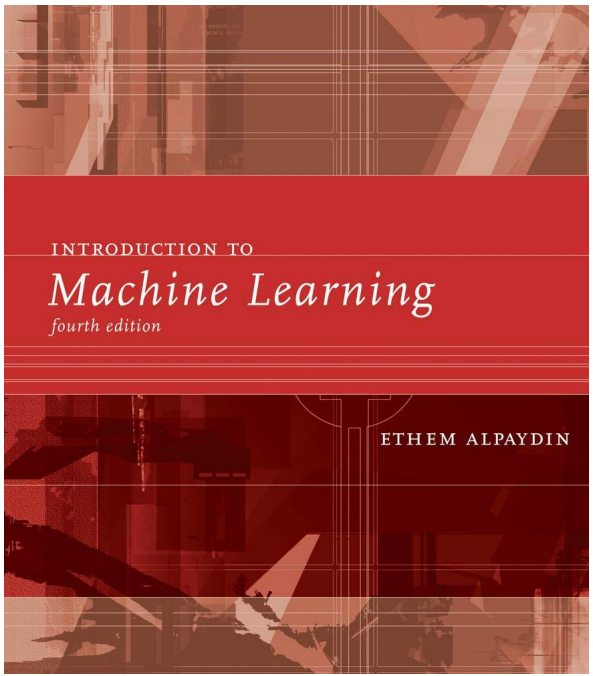
Doing Research

Research Concerns

- How do we assess a model?
- How do we compare two models?
  - Same method, different parameters
  - Different methods
- How do design experiments?

Resource Text



Design and Analysis of Machine Learning Experiments

Chapter 20 in

Introduction to Machine Learning
Ethem Alpaydin
The MIT Press, 2020
ISBN 9780262043793

CUR™
COUNCIL ON UNDERGRADUATE RESEARCH

VBC
VIRTUAL BIENNIAL CONFERENCE
2020

Segregating Data Sets

- Experimentation:
  - Training Set / Validation Set. (2/3^{rds})
- Testing:
  - Separate (never utilized!) test set (1/3rd)

Assessing Model Success

- Train/Test Once: one model, one error value

- Train Once, Multiple Tests: one model/many values
- Multiple Train/Test Runs: many models/values

  ⇒ Evaluate based on distribution of error values

Assessing Model Success

- Consider using
  - Error / Accuracy Rates
  - True Positives / True Negatives
  - Precision / Recall
  - Sensitivity / Specificity
- Receiver Operating Characteristics (ROC) curve
- Confusion Matrix

3.3. Metrics and scoring: quantifying the quality of predictions

There are 3 different APIs for evaluating the quality of a model's predictions:

- **Estimator score method**: Estimators have a `score` method providing a default evaluation criterion for the problem they are designed to solve. This is not discussed on this page, but in each estimator's documentation.
- **Scoring parameter**: Model-evaluation tools using cross-validation (such as `model_selection.cross_val_score` and `model_selection.GridSearchCV`) rely on an internal *scoring* strategy. This is discussed in the section The scoring parameter: defining model evaluation rules.
- **Metric functions**: The `metrics` module implements functions assessing prediction error for specific purposes. These metrics are detailed in sections on Classification metrics, Multilabel ranking metrics, Regression metrics and Clustering metrics.

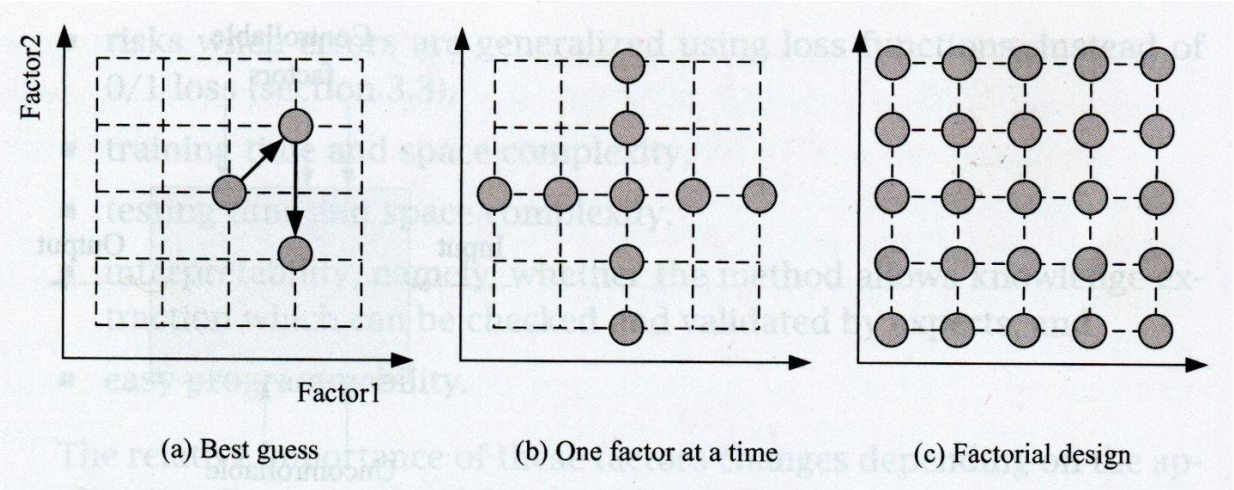
Caveats

Other assessment criteria?

- Ease of Implementation (but, SciKit Learn)
- Training & Testing time/space complexity
- Interpretability

Experimental Design

From [Alpaydin 2020]:



Hypothesis Testing

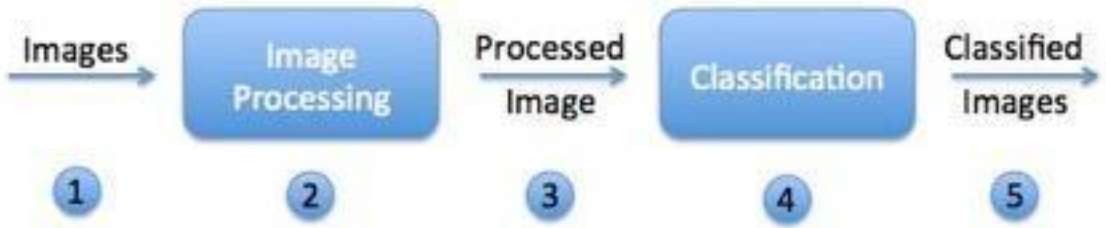
- Standard method: Null Hypothesis
- Binomial Test
- Approximate Normal Test
- (on multiple training/validation sets) t-Test

Because

Results are not domain independent
but a function of our specific dataset

A good result means the inductive bias
for our model corresponds to our
dataset's chacteristics

There is a lot to try



2. Change Data (via preprocessing)
4. Change ML Algorithm
   Change Algorithm Parameters
5. Change Classification Goal

Experimental Basics

- Randomization
- Cross-validation
- Blocking
  - Strategy to reduce/eliminate specific (uninteresting) variabilities
- Establish workflow first

Comparing Classifiers

- (two) McNemar's Test

A-B- # Both Misclassify	A-B+ # Only A Misclassifies
A+B- # Only B Misclassifies	A+B+ # Both Classify Correctly

If same error, expect: A-B+ = A+B- = (A-B+ + A+B-) / 2

$$X^2 \sim (|\text{A-B+} - \text{A+B-}| - 1)^2 / (\text{A-B+} + \text{A+B-})$$

- (two) Paired t-Test
- (multiple) ANOVA

CUR™
COUNCIL ON UNDERGRADUATE RESEARCH

VBC
VIRTUAL BIENNIAL CONFERENCE
2020

Caveats

- Previous tests assumed error values were normally distributed
- Not so across multiple datasets – use nonparamentric tests

Not covered today!

Live Session V

(end of video)

COUNCIL ON UNDERGRADUATE RESEARCH

VBC
VIRTUAL BIENNIAL CONFERENCE
2020