Experimentation and Evaluation

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Eric Eaton (UPenn), David Kauchak (Pomona), and the many others who made their course materials freely available online.

Experimental Procedure

Learning Goals

- Describe how cross-validation (*k*-fold, leave-one-out) is used to evaluate model and optimize hyperparameters
- Describe how to compare models statistically using the *t*-test
- Describe how bootstrapping is used to evaluate test performance

Proper Evaluation?

Current plan

- Learn algorithm on training data (subset of full data)
- Evaluate on test data (subset of full data)
- Repeat until happy with results

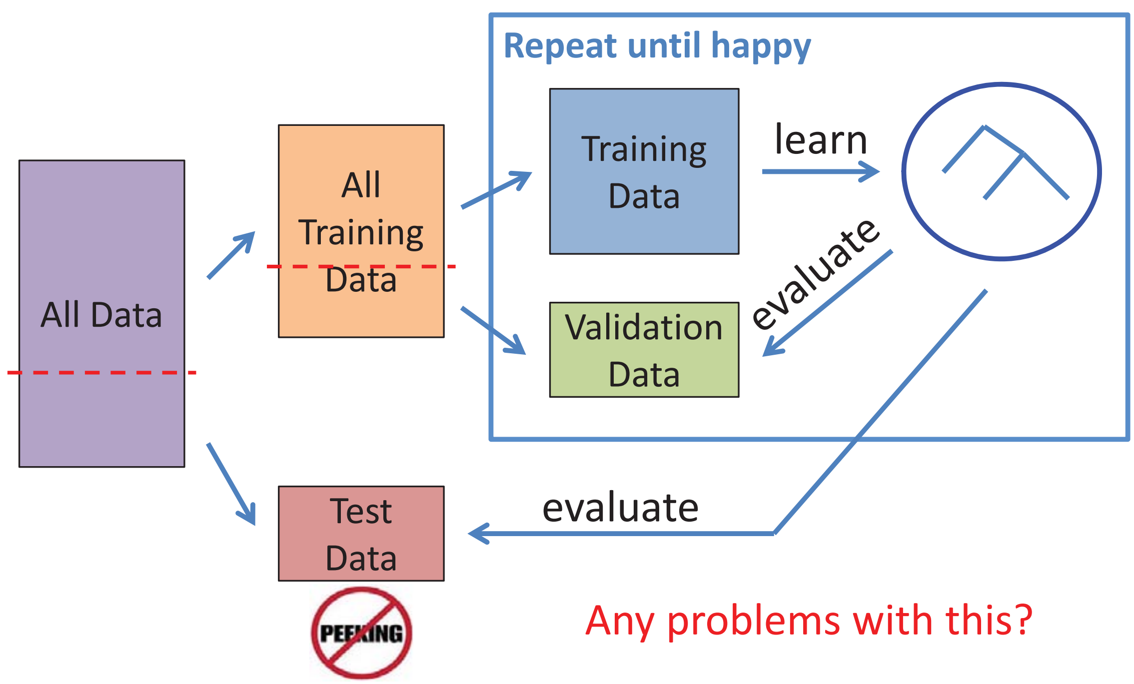
Is this okay?

- No! Although we are not explicitly looking at test data, we are still "cheating" by biasing our algorithm to test data
- **Once you look at / use test data, it is no longer test data!**

So, how can we evaluate our algorithm during development?

Based on slide by David Kauchak

Classification Evaluation



Any problems with this?

Based on slide by Eric Eaton

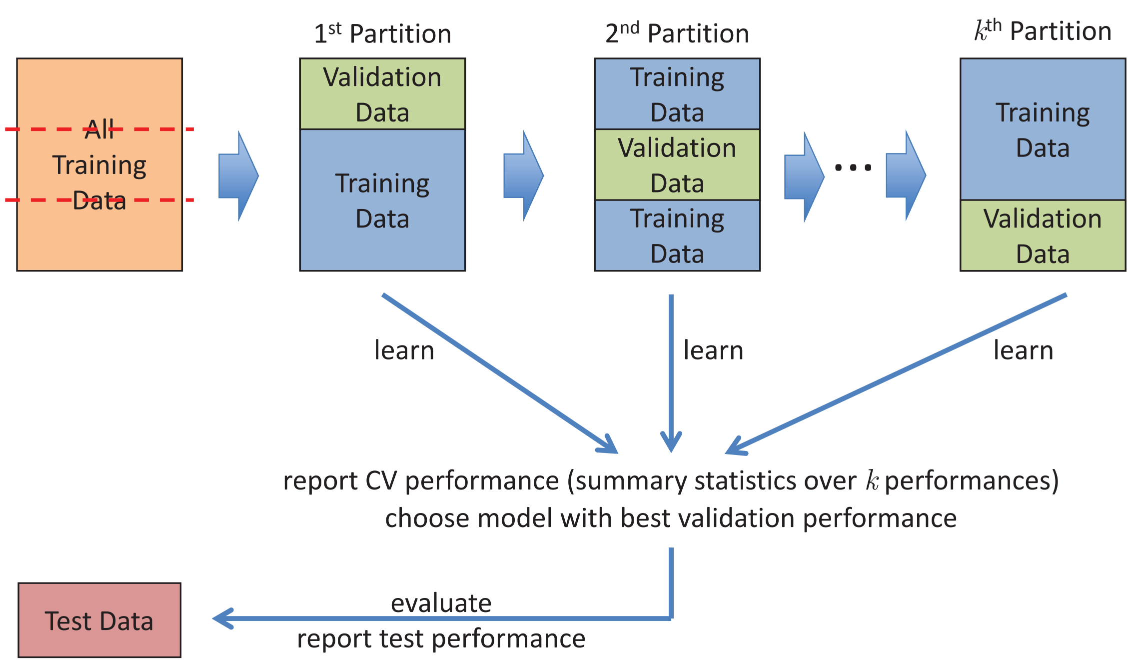
k-Fold Cross-Validation

- Why just choose one particular "split" of data?
  – in principle, we should do this multiple times since performance may be different for each split

- k-Fold Cross-Validation (e.g., $k = 10$)
  – randomly partition all training data of n instances into ***k* disjoint subsets** (each roughly of size n/k)
  – choose each fold in turn as validation set; train model on the other $k-1$ folds and evaluate
  – compute statistics over k test performances, or choose best of k models

Based on slide by Eric Eaton

Example: 3-Fold CV

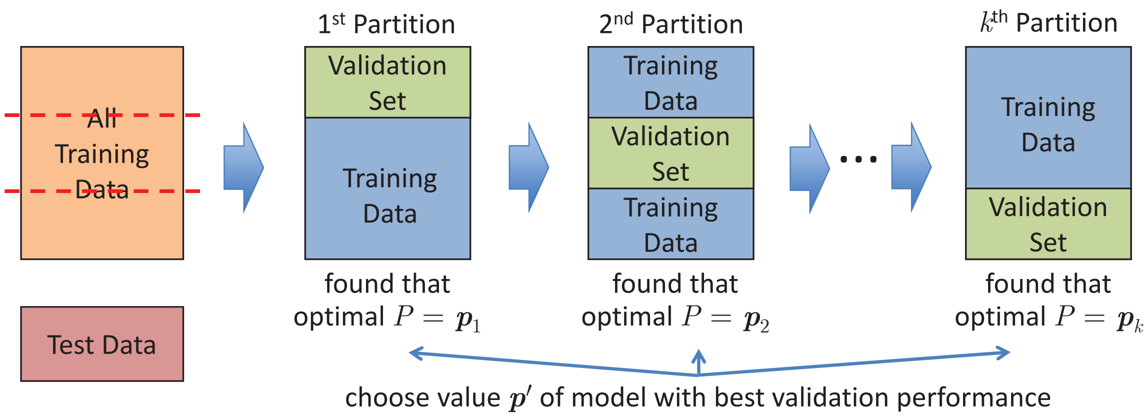


Based on slide by Eric Eaton

Optimizing Model Parameters

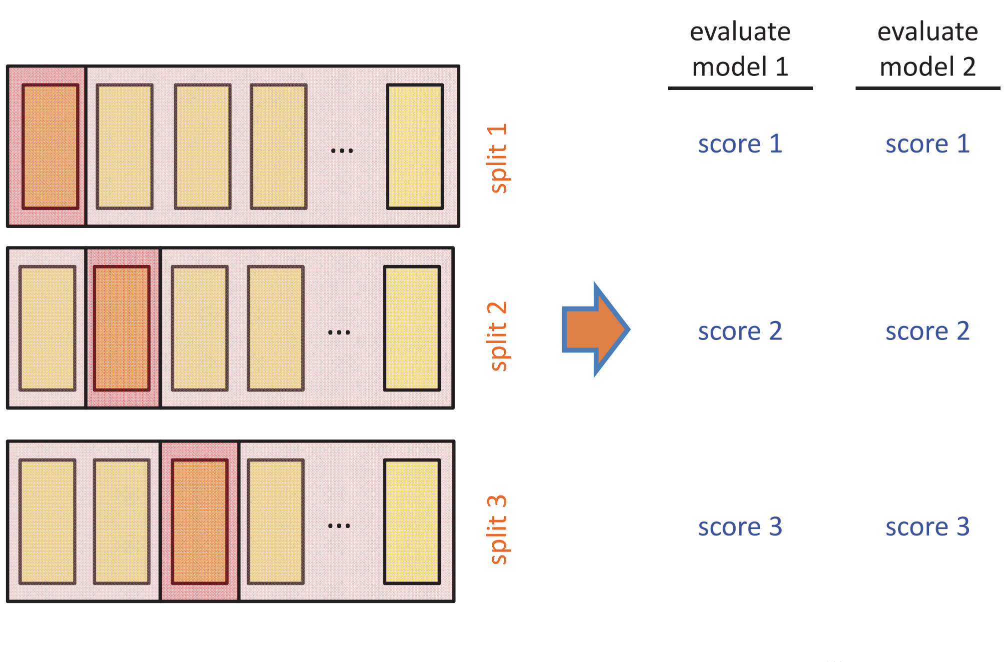
Can also use CV to choose value of model parameter P

- Search over space of parameter values $\boldsymbol{p} \in values(P)$
  - evaluate model with $P = \boldsymbol{p}$ on validation set
- Choose value $\boldsymbol{p}'$ with highest validation performance
- Learn model on full training set with $P = \boldsymbol{p}'$



Based on slide by Eric Eaton

Example : Comparing Models



Based on slide by David Kauchak

Comparing Models

Is model 2 better than model 1?

Sample 1

split	M1	M2
1	87	88
2	85	84
3	83	84
4	80	79
5	88	89
6	85	85
7	83	81
8	87	86
9	88	89
10	84	85
avg	**85**	**85**

Sample 2

split	M1	M2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
avg	**82**	**85**

Sample 3

split	M1	M2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
avg	**82**	**85**

Sample 4

split	M1	M2
1	80	82
2	84	87
3	89	90
4	78	82
5	90	91
6	81	83
7	80	80
8	88	89
9	76	77
10	86	88
avg	**83**	**85**

How do we decide if model 2 is better than model 1?

Based on slide by David Kauchak

Statistical tests

Setup

- assume some default hypothesis about data that you would like to *disprove*, called the null hypothesis
- e.g. model 1 and model 2 are not statistically different in performance

Test

- calculate test statistic from data (often assuming something about data)
- calculate *p*-value from test statistic
  - *p*-value = probability of seeing test statistic at least as extreme as one actually observed given null hypothesis is true
- compare p-value to threshold α (significance level)
- reject null hypothesis if $p < \alpha$
- note that statistically significant difference is not necessarily a large-magnitude difference

Based on slide by David Kauchak

t-test

Determines whether two samples come from same underlying distribution or not

Null hypothesis

- model 1 and model 2 accuracies are no different, i.e. come from **same** distribution

Assumptions

- there are a number that often are not completely true, but we are often not too far off

Our formulation

- do "paired *t*-test"
  - values can be thought of as pairs, calculated under same conditions (in our case, same train/test split)
  - gives more power than unpaired *t*-test (we have more information)
- for almost all experiments, do "two-tailed" version
  - no *a priori* knowledge of which model is better

Based on slide by David Kauchak

Comparing Models

Is model 2 better than model 1?

Sample 1

split	M1	M2
1	87	88
2	85	84
3	83	84
4	80	79
5	88	89
6	85	85
7	83	81
8	87	86
9	88	89
10	84	85
avg	**85**	**85**

$p = 1$

Sample 2

split	M1	M2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
avg	**82**	**85**
sdev	**5.9**	**3.9**

$p = 0.15$

Sample 3

split	M1	M2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
avg	**82**	**85**
sdev	**2.3**	**1.7**

$p = 0.007$

Sample 4

split	M1	M2
1	80	82
2	84	87
3	89	90
4	78	82
5	90	91
6	81	83
7	80	80
8	88	89
9	76	77
10	86	88
avg	**83**	**85**
sdev	**4.9**	**4.7**

$p = 0.001$

Based on slide by David Kauchak

Leave-One-Out CV (LOOCV)

- Special case where $k = n$
  - Each partition now one example
  - Train using $n - 1$ examples, validate on remaining example
  - Repeat n times, each with different validation example
  - Finally, choose model with smallest average validation error
- When is it used?
  - **Can be expensive** for large n, so typically used when n is small
  - Useful in domains with limited training data (**maximizes data used for training**)

Based on slide by Piyush Rai

Summary : Cross-Validation

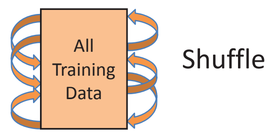
- Cross-validation generates an approximate estimate of how well the classifier will do on "unseen" data
  - as $k \rightarrow n$, model becomes more accurate (more training data)
  - … but, CV becomes more computationally expensive (have to train k models)
  - choosing $k < n$ is a compromise
- It is an even better idea to do CV repeatedly!

Based on slide by Eric Eaton

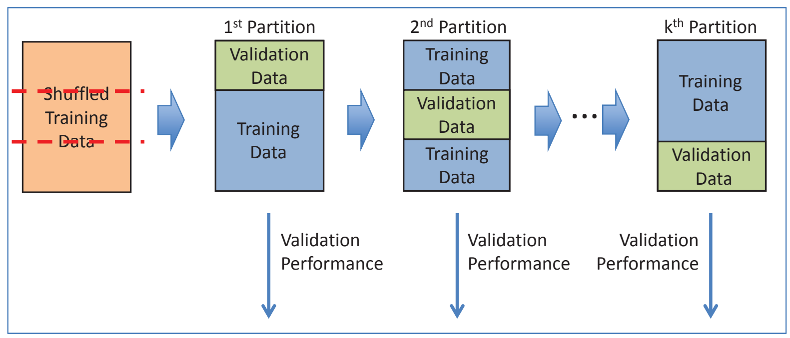
Multiple Trials of k-Fold CV

1) Loop for t trials:

a.) Randomize Data Set



b.) Perform k-fold CV



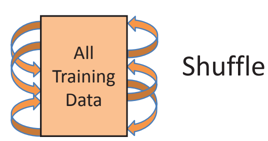
2) Compute statistics over $t \times k$ validation performances

Based on slide by Eric Eaton

Comparing Multiple Classifiers

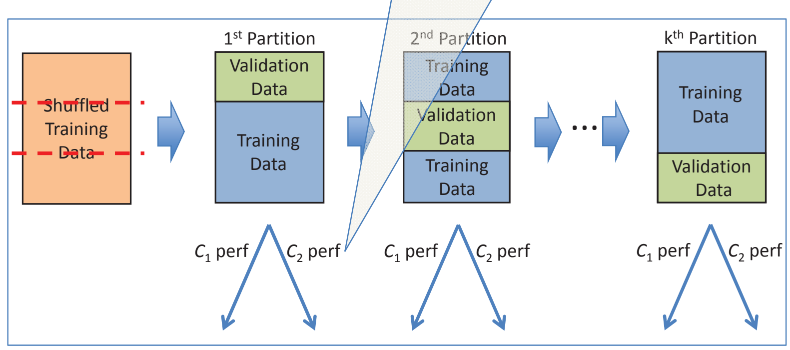
1) Loop for t trials:

a.) Randomize Data Set



Test each candidate learner on same training/testing splits

b.) Perform k-fold CV

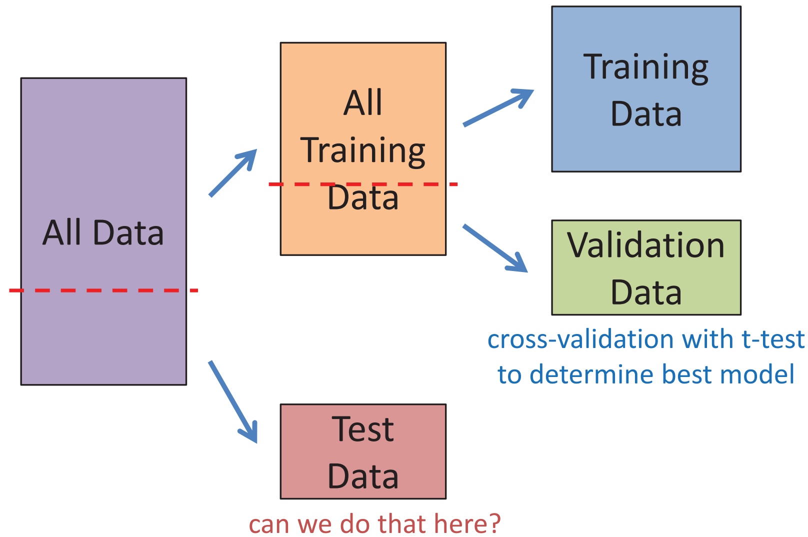


2) Compute statistics over $t \times k$ validation performances

Allows us to do paired summary statistics (e.g., paired t-test)

Based on slide by Eric Eaton

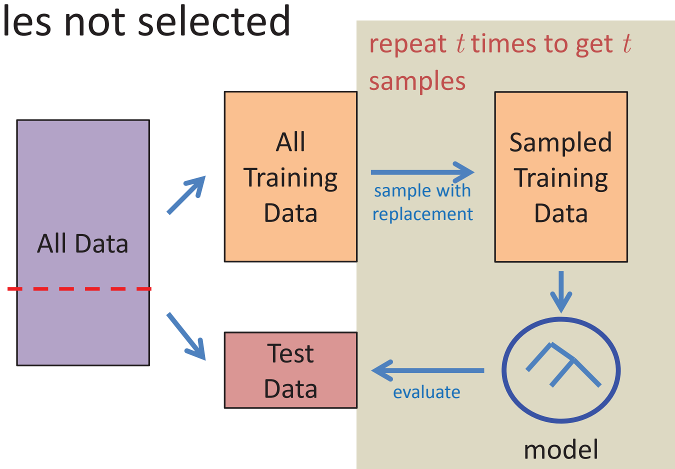
Statistical Tests on Test Data



Based on slide by David Kauchak

Bootstrapping

- Given set of n examples
- Sample n elements from this set **with replacement** to create new training set
- Use set of examples not selected as validation set
- Repeat t times



Based on slide by Piyush Rai

Experimentation Good Practices

Never look at your test data!

During development

- compare different models / hyperparameters on development data
- use cross-validation to get more consistent results
- if you want to be confident with results, use t-test

For final evaluation, use bootstrap resampling combined with t-test to compare

Based on slide by David Kauchak

Avoiding Pitfalls

- Is my held-aside test data really representative of going out to collect new data?
- Did I repeat my entire data processing procedure on every fold of cross-validation, using only training data for that fold?
- Have I modified my algorithm so many times, or tried so many approaches, on this same data set that I (human) am overfitting it?

Based on slide by David Page

The Short Way

(that Many People Actually Use)

- Split into only training data + validation data
- Train on training data, evaluate on validation data
- Report cross-validation performance
  - possibly also training performance

- Why is this used?
  - might not be enough data to create held-out test set
  - you cannot trust that authors did not peek at test data anyway =P