Model Overfitting

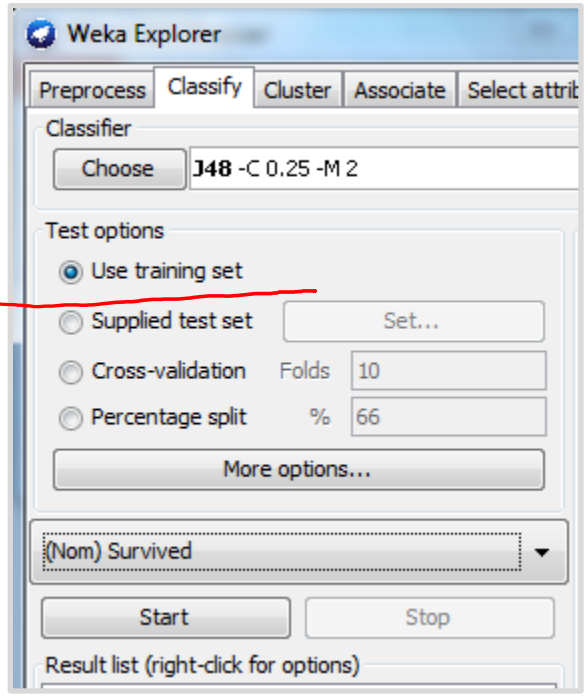
School of Information Studies
Syracuse University

Model Generalization

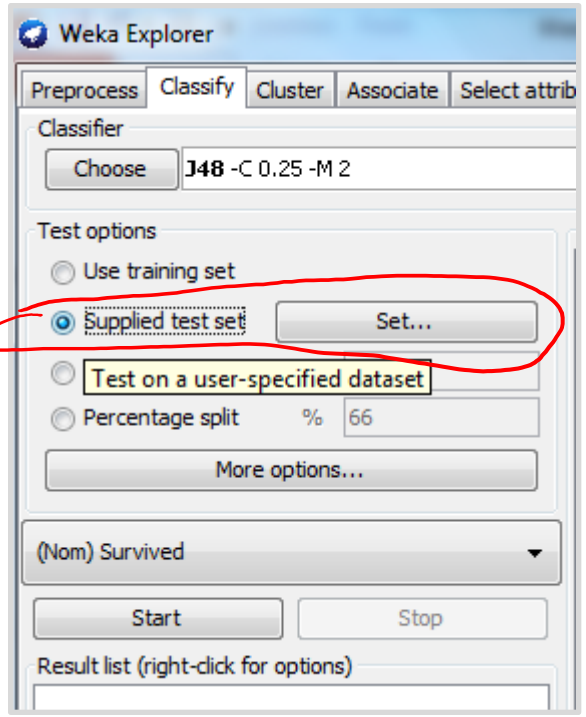
Two fundamental concepts:

- ***Training error**: train a model (e.g., a decision tree model) on the training data set, then test the model on the same training set. The error rate is called "training error," which evaluates how well the model **fits** the training data.
- ***Test error**: test the model on a test data set that is different from the training set. The error rate is called "test error," which evaluates how well the model **generalizes** to unseen data.

Training Error vs. Test Error



Weka: the evaluation option to obtain **training error**



Weka: the evaluation option to obtain **test error**

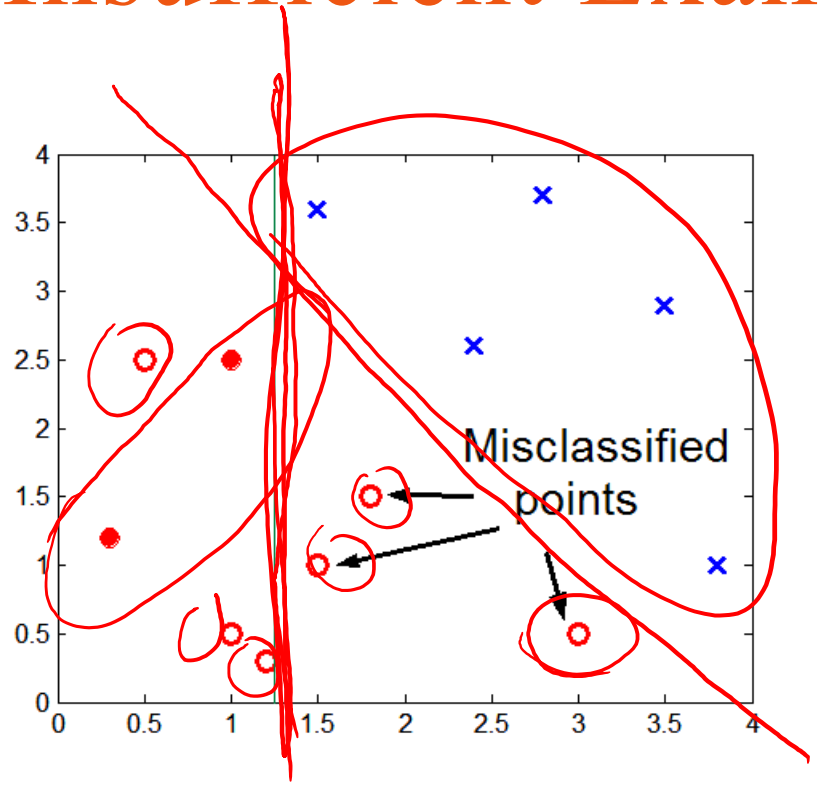
Model Overfitting

Overfitting means a model fits the training data very well, but generalizes to unseen data poorly.

How do I know if my model is overfitting?

- Your model is overfitting if its **training error is small** (fits well with training data), but **the test error is large** (generalizes poorly to unseen data).
- Did it happen to your Naïve Bayes model?

Overfitting Due to Insufficient Examples



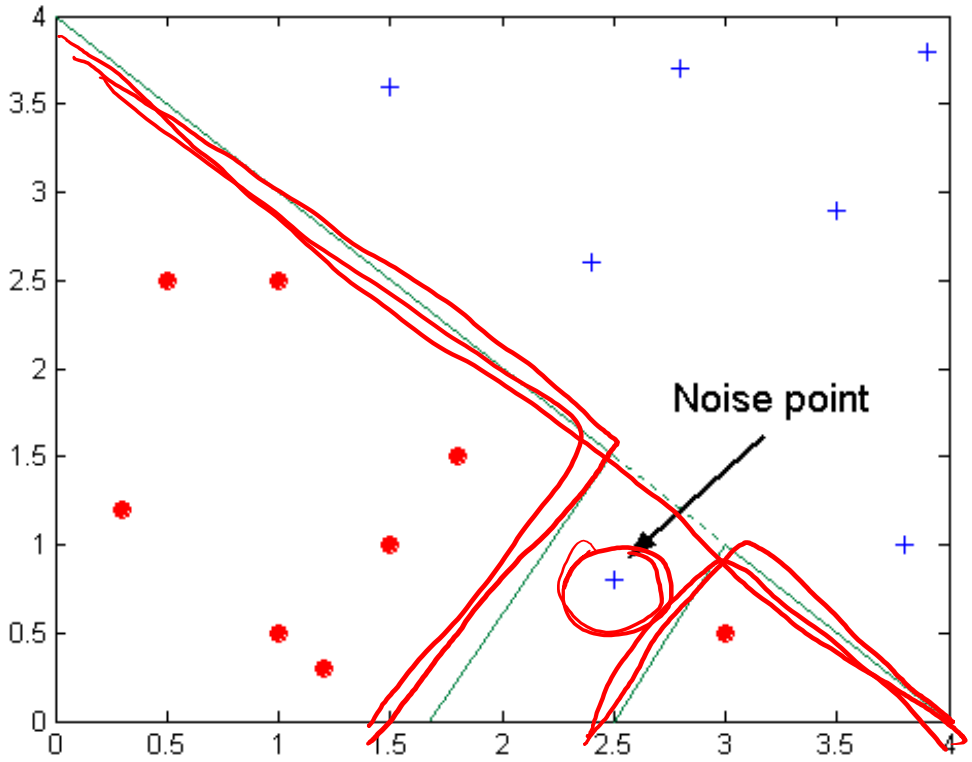
Blue crosses and solid red dots are training data.

Red circles are test data.

The green vertical line is the decision boundary.

Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels in that region.

Overfitting Due to Noise



The decision boundary (supposedly a straight line) is distorted by the noise point. The overfitted decision boundary is the solid blue lines.

Occam's Razor

Given two models of similar generalization errors, the simpler model is preferred over the more complex model.

For complex models, there is a greater chance that it was overfitted accidentally by errors in data.

Therefore, model complexity should be considered when evaluating a model.

Evaluation Methods

School of Information Studies
Syracuse University

Evaluation Methods to Avoid Model Overfitting

But… we have to estimate how good a model is before using it in real predictions.

Some evaluation methods have been designed to test the model on training data while controlling model overfitting.

- Hold-out test
- Cross validation

Hold-Out Test

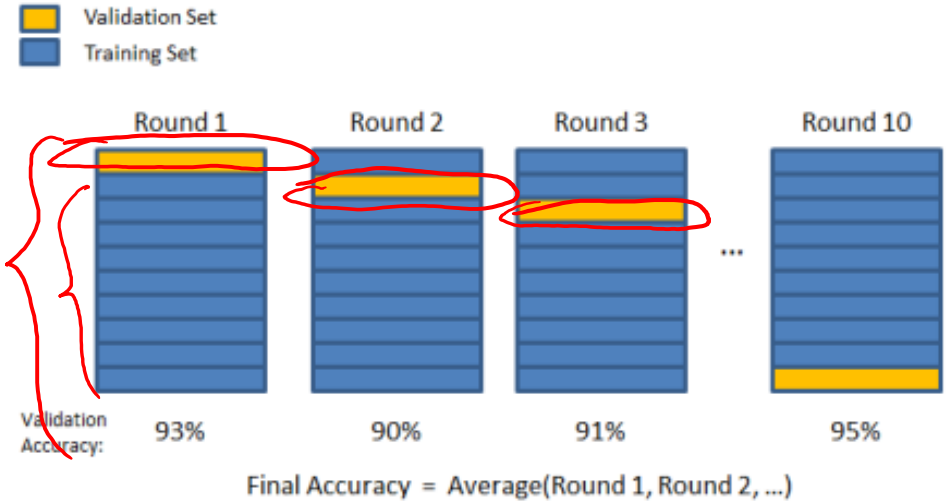
Hold-out test

- Split the training data to two subsets, using one subset for training and the other for testing.
- The splitting ratio is determined by the training set size in that both subsets cannot be too small.
- 50/50 or 2:1 are common splitting ratios.

Cross Validation (CV)

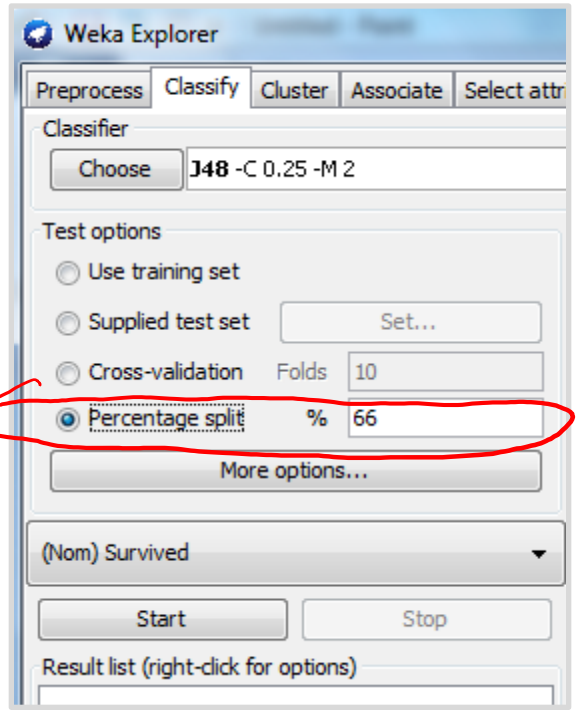
N-fold cross validation (CV)

- N is determined by the training set size. The larger the N, the longer it takes to run the experiment.
- 5 or 10 are common choices for N.

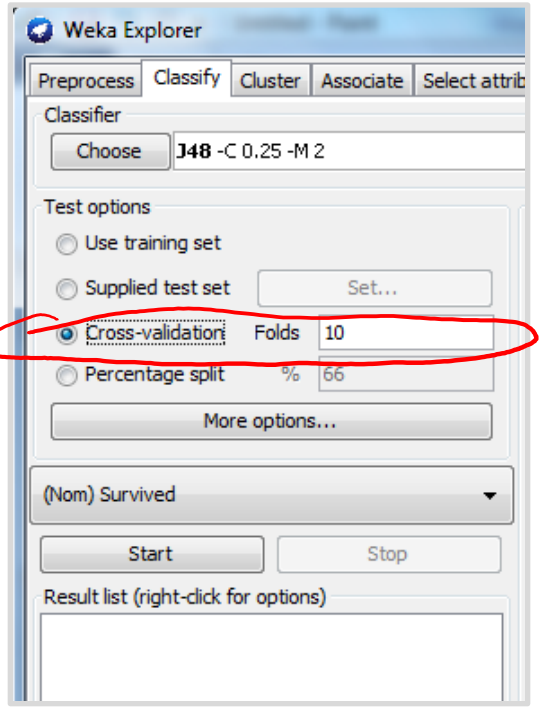


http://chrisjmccormick.wordpress.com/2013/07/31/k-fold-cross-validation-with-matlab-code/

Hold-Out Test vs. Cross Validation



Weka test option for **hold-out test**



Weka test option for **cross validation**

Hold-Out Test vs. Cross Validation

Hold-out test

- Pros: fast
- Cons: high variability in the result depending on the split

Cross validation

- Pros: less variability and thus more reliable error estimation
- Cons: takes longer

Limitation of Cross Validation

Since cross validation result is still an estimation of the real test error, even if you get good cross validation accuracy on the training data, your final test accuracy may still be different from it, either higher or lower, but the difference should not be large.

Aspects of Model Performance

School of Information Studies
Syracuse University

Aspects of Model Performance

Accuracy: total correct predictions/total

Speed

- Time to construct model (training time)
- Time to use the model (prediction time)

Robustness: handling noise and missing values

Scalability: efficiency in handling large data set

Interpretability

- Understanding and insight provided by the model

Compare DT and MNB Time

Time for training model

Time for prediction

```
Size of the tree :      411

Time taken to build model: 17.5 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.17 seconds
```

```
Time taken to build model: 1.13 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.16 seconds
```

MNB Robustness

Robustness: handling noise and missing values

- Noise?
  - If you change an example's label from "pos" to "neg," how would that affect the model's performance?
- Missing value?
  - Does it exist in text vectors?

MNB Scalability

Scalability: efficiency in handling large data set

Can the probabilities be calculated using parallel processing?

MNB Interpretability

MNB is a linear model:

$$\hat{P}(c|d) \propto \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

$$\log \frac{\hat{P}(c|d)}{\hat{P}(\bar{c}|d)} = \log \frac{\hat{P}(c)}{\hat{P}(\bar{c})} + \sum_{1 \le k \le n_d} \log \frac{\hat{P}(t_k|c)}{\hat{P}(t_k|\bar{c})}$$

Feature weight

odds ratio

Evaluation Baselines

School of Information Studies
Syracuse University

Metrics of Model Performance

Most common measure

- Accuracy

Accuracy

Accuracy: total correct/total

- So you get 90% accuracy; isn't that great?!
- Maybe not, it depends on your baseline.

Random Guess Baseline

50% for binary classification

1/n for n-class classification

Majority Vote Baseline

- What is majority vote baseline?
  - It's a trivial classifier that classifies all examples to the majority class.
  - Example: A spam data set includes 90% spams and 10% regular mails, so the majority vote baseline is 90%.
  - It means this trivial classifier predicts everything as spam and gets 90% accuracy.
  - A good classifier has to beat this baseline to claim effectiveness.

Which Baseline to Choose?

Accuracy: total correct/total

- So you get 90% accuracy; isn't that great?!
- If your data set is balanced with equal number of examples in each category, then **yes**!
- But maybe not, if you have a very skewed data set, in which case you need to compare your classifier's accuracy against the majority vote baseline.
- Or compare your classifier to the state-of-the-art classifier.

Evaluation Metrics

School of Information Studies
Syracuse University

More Evaluation Measures

Precision

Recall

F-measure

Confusion matrix

Confusion Matrix

Confusion matrix for two classes

- Examines results of the classifier on the test set. Compare the predicted labels against the ground truth.

		Predicted sentiment	
		Positive	Negative
Actual sentiment	Positive	a	b
	Negative	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Precision, Recall, and f-Measure

Recall and precision for each class:

- Recall $_{Class=Yes}$ = a / a + b , Recall $_{Class=No}$ = d / c + d
- Precision $_{Class=Yes}$ = a / a + c , Precision $_{Class=No}$ = d / b + d
- F_1-measure = 2/(1/P+1/R)=2PR/(P+R)

		Predicted sentiment		
		Positive	Negative	Recall
Actual sentiment	Positive	a	b	a/(a+b)
	Negative	c	d	d/(c+d)
	Precision	a/(a+c)	d/(b+d)	

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Confusion Matrix for Multi-Class

		Predicted sentiment		
		Positive	Negative	Neutral
Ground truth	Positive	a	b	e
	Negative	c	d	f
	Neutral	g	h	i

Manual Annotation Reliability

School of Information Studies
Syracuse University

Where Does the Ground Truth Come From?

Convenience sample

- Star rating from customer reviews

Manual annotation

How Trustworthy Is Manual Annotation?

Reliability test

- If asking two or more people to mark the sentiment of a collection of tweets, to what extent will they agree with each other?

Inter-Coder Agreement

Raw agreement:

- a = count(agreed_items)/total_items

Problem with raw agreement:

- Skewed categories: 90% raw agreement in both tables

		Coder A	
		Positive	Negative
Coder B	Positive	45	5
	Negative	5	45

		Coder A	
		Positive	Negative
Coder B	Positive	90	10
	Negative	0	0

Cohen's Kappa

a = raw_agreement

c = chance_agreement

K = (a-c)/(1-c)

		Coder A	
		Positive	Negative
Coder B	Positive	45	5
	Negative	5	45

		Coder A	
		Positive	Negative
Coder B	Positive	90	10
	Negative	0	0

Cohen's Kappa

a = raw_agreement

c = chance_agreement

K = (a-c)/(1-c)

K=0.8

		Coder A	
		Positive	Negative
Coder B	Positive	45	5
	Negative	5	45

K=0

		Coder A	
		Positive	Negative
Coder B	Positive	90	10
	Negative	0	0

How to Calculate Kappa?

Given a confusion matrix of two coders

		Coder A	
		Positive	Negative
Coder B	Positive	45	5
	Negative	5	45

How to Calculate Kappa?

Calculate marginal distribution

		Coder A		
		Positive	Negative	
Coder B	Positive	45	5	50%
	Negative	5	45	50%
		50%	50%	

How to Calculate Kappa?

Calculate raw agreement (a = 0.9)

Calculate

- P (both A and B gives “positive” label) = 0.25
- P (both A and B gives “negative” label) = 0.25
- Chance_agreement: c = 0.25+0.25 = 0.5
- Kappa = (a-c)/(1-c) = (0.9-0.5)/(1-0.5) = 0.4/0.5 = 0.8

What Is Kappa for This?

		Coder A	
		Positive	Negative
Coder B	Positive	89	9
	Negative	1	1

$K = .139$

Tools to Calculate Kappa

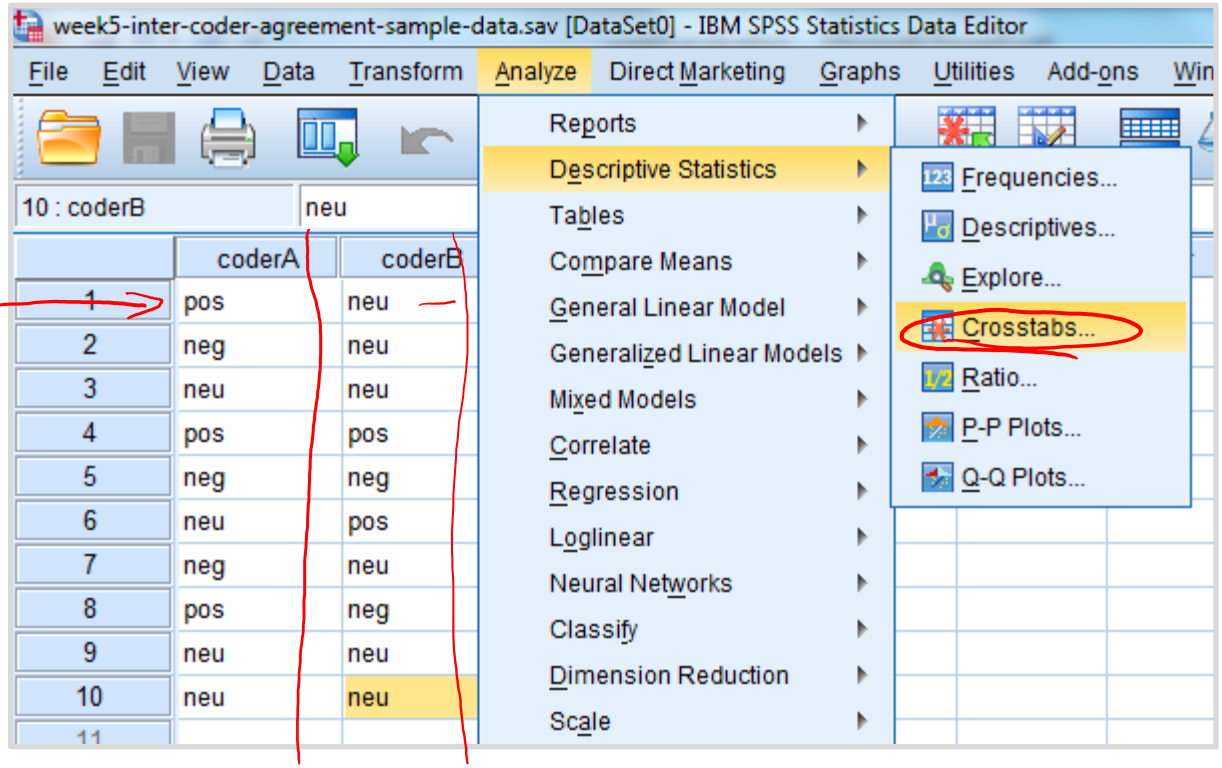
Online tool

- http://vassarstats.net/kappa.html

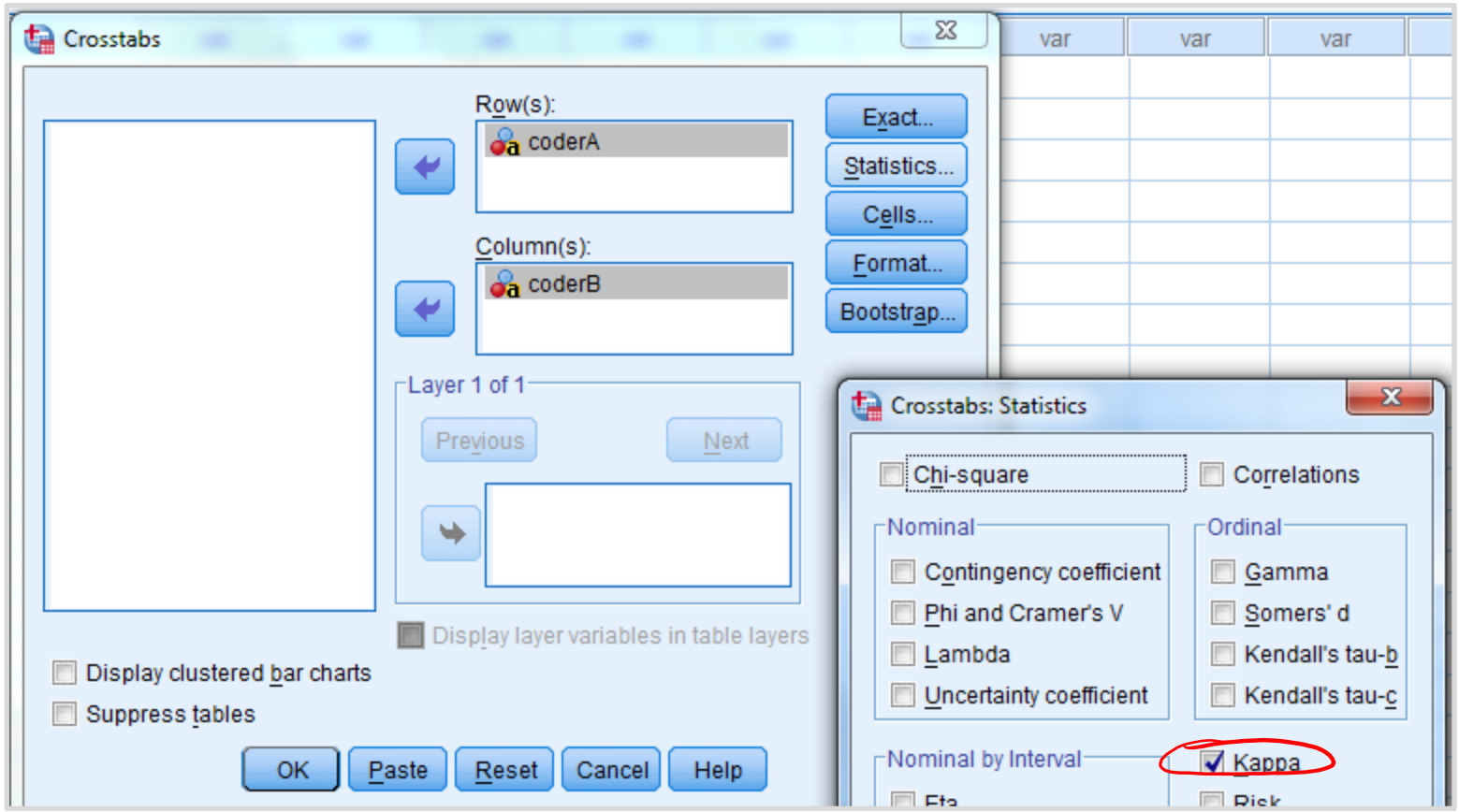
Statistical tools

- R package “irr”
- SPSS: crosstab function
- sklearn.metrics.cohen_kappa_score

Kappa Calculation in SPSS



Kappa Calculation in SPSS



Kappa Calculation in SPSS

coderA * coderB Crosstabulation

Count

		coderB			Total
		neg	neu	pos	
coderA	neg	1	2	0	3
	neu	0	3	1	4
	pos	1	1	1	3
Total		2	6	2	10

Symmetric Measures

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Measure of Agreement	Kappa	.219	.229	1.017	.309
N of Valid Cases		10			

a. Not assuming the null hypothesis.

b. Using the asymptotic standard error assuming the null hypothesis.

Multiple Coders

Average Kappa

Kripendorff's Alpha

- http://afhayes.com/spss-sas-and-mplus-macros-and-code.html
- Search "kalpha"

Bias in Human Annotations

School of Information Studies
Syracuse University

Manual Annotation

Develop the codebook

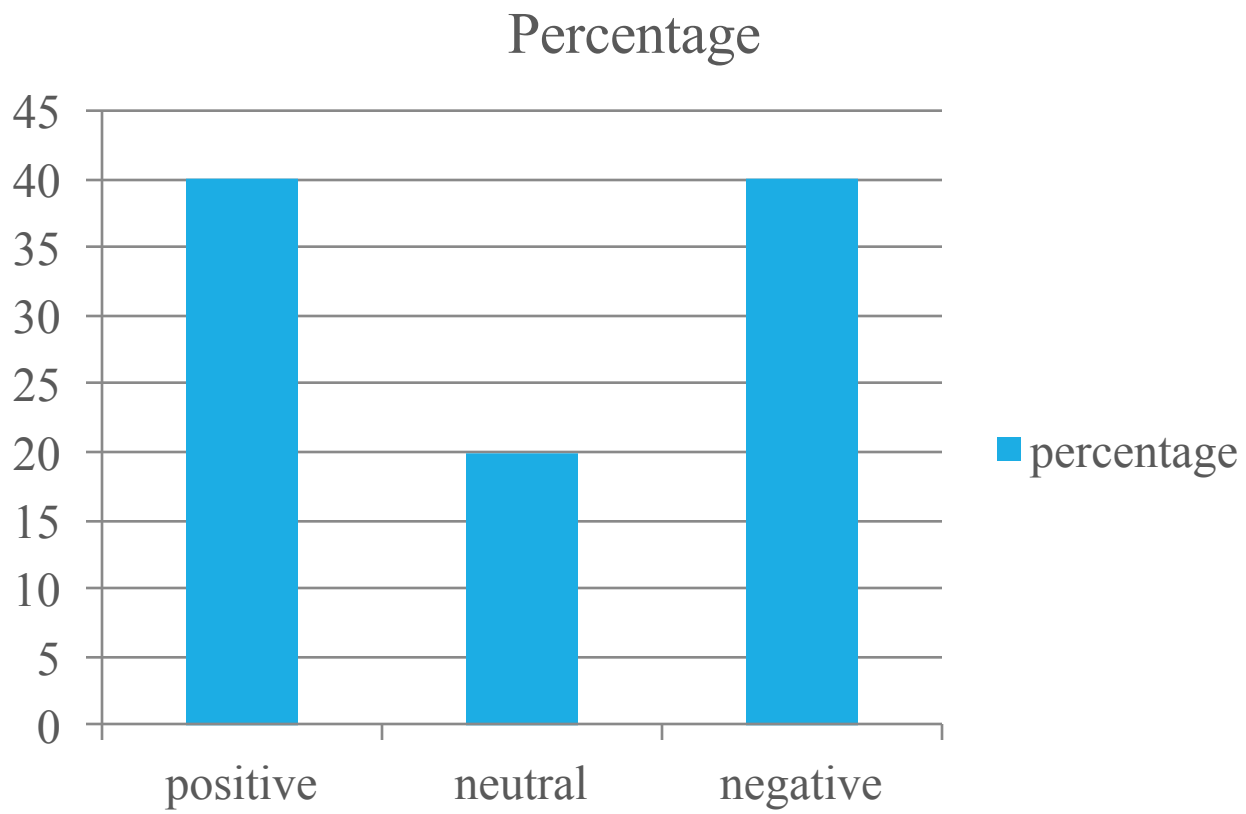
- Define the labels
  - What is positive, negative, and neutral?
  - Example: “I like Target better than Walmart”
    - Is it positive or negative?

Identify Systematic Bias

Marginal distribution of (positive, negative, neutral) labels:

- Some coders may favor the positive label.
- Some favor the negative label.
- Some favor the neutral label.
- The marginal distribution would be very much different.

A “Polarized” Coder



A “Neutral” Coder