INFO 371
Introduction to Data Science

Spring 2018

Are algorithms good judges?

Catherine Matacic. (2018) *Science.* 359: 6373

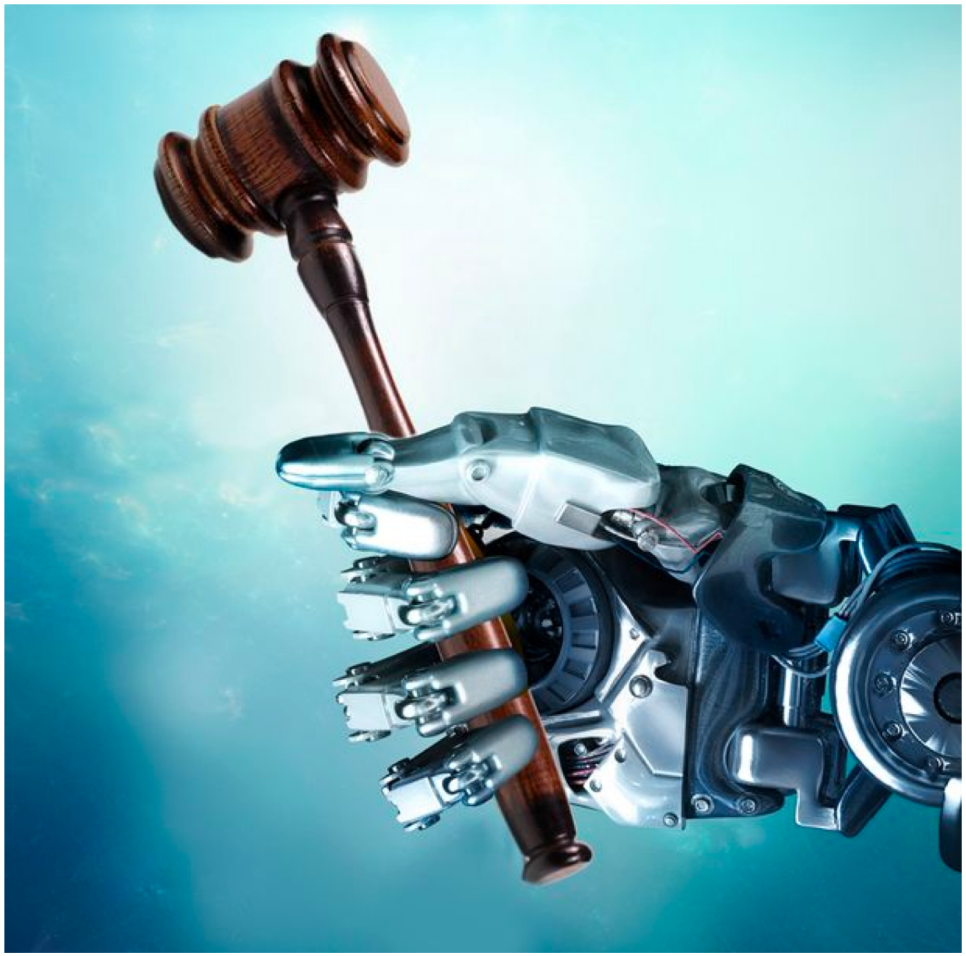


Image: https://www.mirror.co.uk/tech/artificially-intelligent-judge-could-help-9114133

CONFRONTING FAKE NEWS & MISINFORMATION

Lectures are open to members of the UW community.

MINI LECTURE SERIES

MUDDIED WATERS: ONLINE DISINFORMATION DURING CRISIS EVENTS

April 18 from 5-6:15 p.m.
in Bagley 131

Kate Starbird
Assistant Professor, Human Centered Design & Engineering, University of Washington

Public debate around "fake news" highlights the growing challenge of determining information veracity online. Our strategies for making sense of information make us vulnerable to absorbing and passing along misinformation. Certain actors exploit these vulnerabilities, spreading intentional misinformation—or disinformation—for various reasons, including geopolitical goals. Drawing on recent research, this talk explores what "conspiracy theories" of crisis events reveal about "fake news", political propaganda, and disinformation.

CLEANING UP OUR POLLUTED INFORMATION ENVIRONMENTS

April 24 from 5-6:15 p.m.
in Gowen 301

Jevin West
Assistant Professor, Information School, University of Washington

Pandering politicians, winking advertisers, startup soothsayers, television "experts", and even some scientists use the news media to promulgate half-truths, misrepresentations and outright lies. Technology and our legal system are of little help in solving this problem. Cleaning up our polluted information environments requires a digital citizenry that can spot and refute BS. This talk will provide strategies for combatting a particular kind of BS—BS cloaked in data, figures, statistics and algorithms.

THE NEW GLOBAL POLITICS OF WEAPONIZED AI PROPAGANDA

April 30 from 5-6:15 p.m.
in HUB Lyceum

Berit Anderson
CEO & Editor-in-Chief, Scout.ai, a media company covering the future of technology, its risks and rewards through investigative reporting, analysis, and science fiction.

Silicon Valley spent the last ten years building platforms whose natural end state is digital addiction. In 2016, a small group of powerful actors hijacked them. This talk explores how powerful individuals and companies are using technology to manipulate citizen behavior and shift the outcomes of elections around the world, and what policymakers, technologists, journalists, and individuals can do about it.

Assignment #2 Questions?

Quiz #1 (April 17, 2018)

- Nearest Neighbors (~1 questions)
- Designing ML experiments, training & test data (~1-2 questions)
- Supervised vs Unsupervised (~1 question)
- Cross Validation (~1 question)
- PageRank (~1 question)
- Linear Algebra (~1 question)
- Python and data structures (~1-2 questions)
- Readings (~2-3 questions)
  - Green (Linear Algebra Appendix, pg 803 - 819) and anything from lecture/lab
  - Page, Lawrence, et al. *The PageRank citation ranking: Bringing order to the web*.
  - P. Domingos "A Few Useful Things to Know about Machine Learning"
  - *Data Mining, Whitten et al. (Chpts. 5, especially 5.2-5.4)
  - Daume (Chpt. 3.1-3.3)
- Approximately 30 minutes to complete the quiz

Schedule

- April 17: Problem #2 DUE at midnight
- April 19: Quiz #1
- May 1: Problem Set #3 DUE

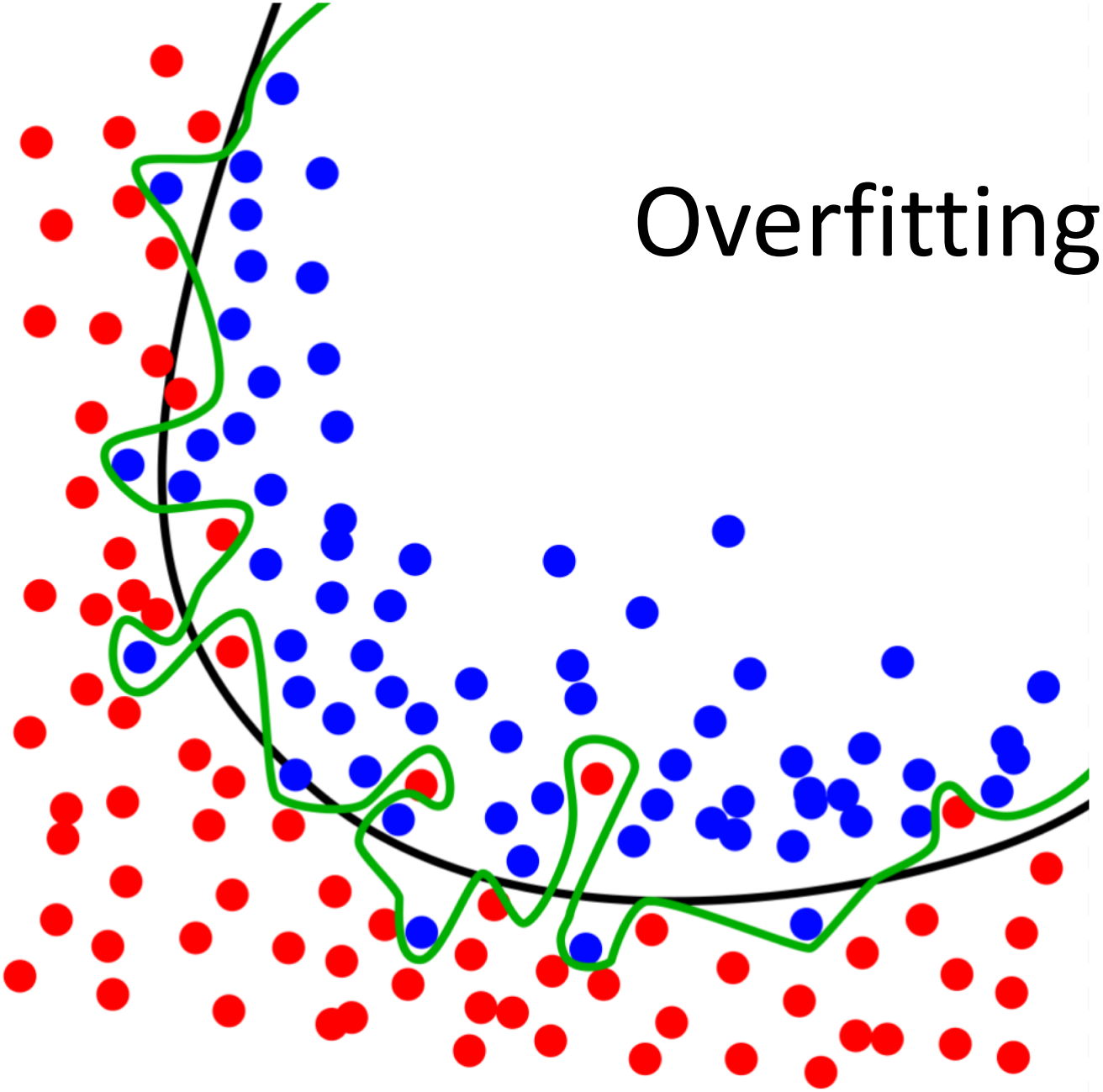
Key Issues in (Supervised) ML

- Don’t forget about Theory
  - Every learner must embody some knowledge or assumptions beyond the data it is given in order to generalize beyond it
  - One of the key criteria for choosing a representation is which kinds of knowledge are easily expressed in it

Key Issues in (Supervised) ML

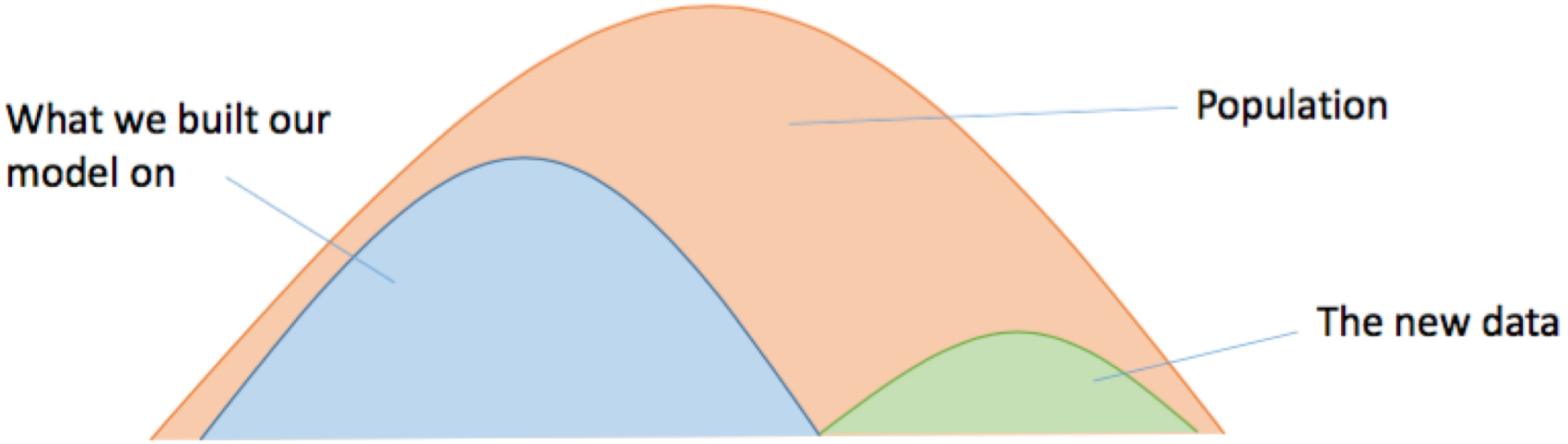
- Generalization
  - "The fundamental goal of machine learning is to generalize beyond examples in the training set. This is because, no matter how much data we have, it is very unlikely that we will see those exact examples again at test time."
  - Keep an uncontaminated training set
  - But note: Can still be very hard to measure how well your fitted model will generalize to new data!
- … and Overfitting
  - "The bugbear of machine learning"

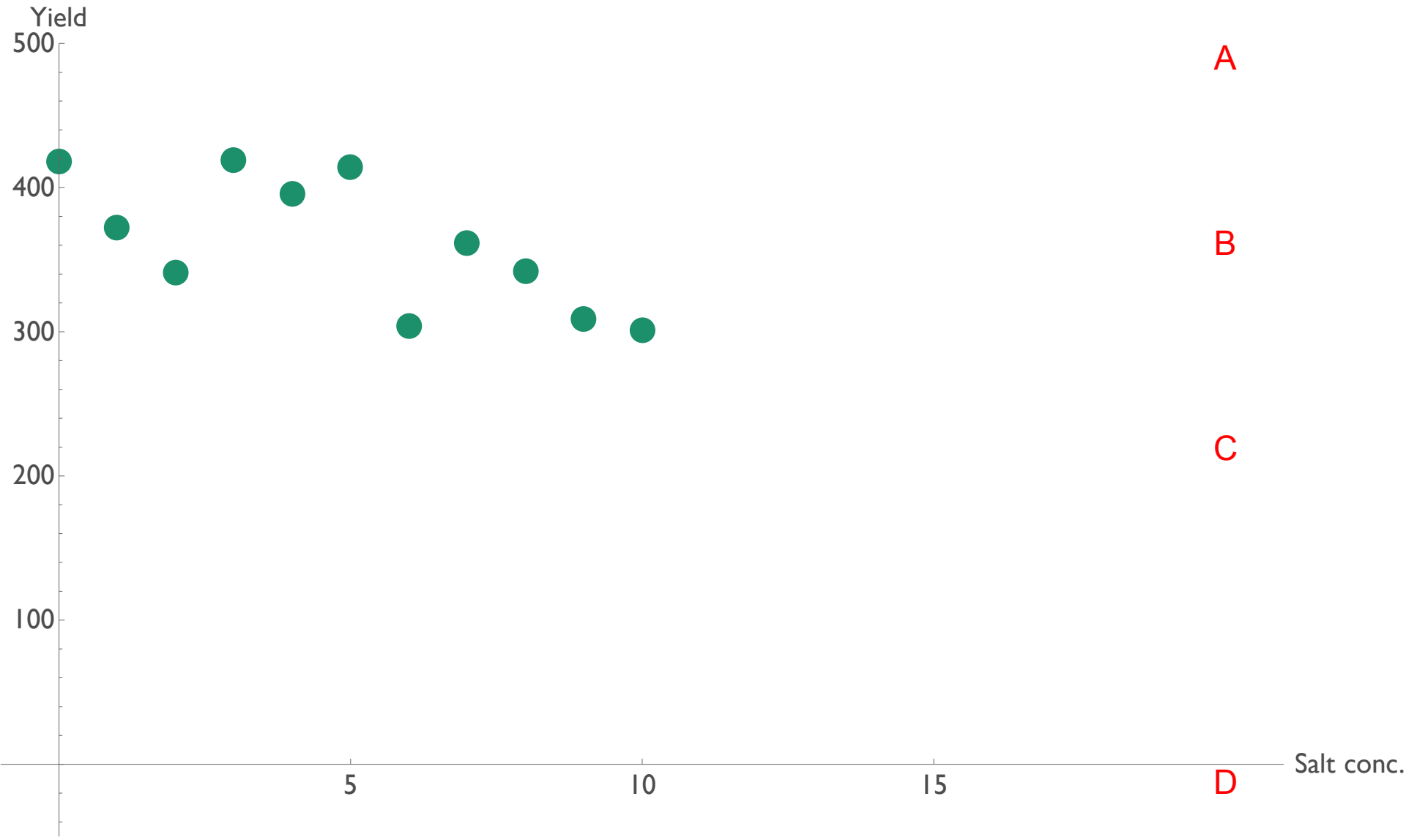
What is over-fitting, anyway?

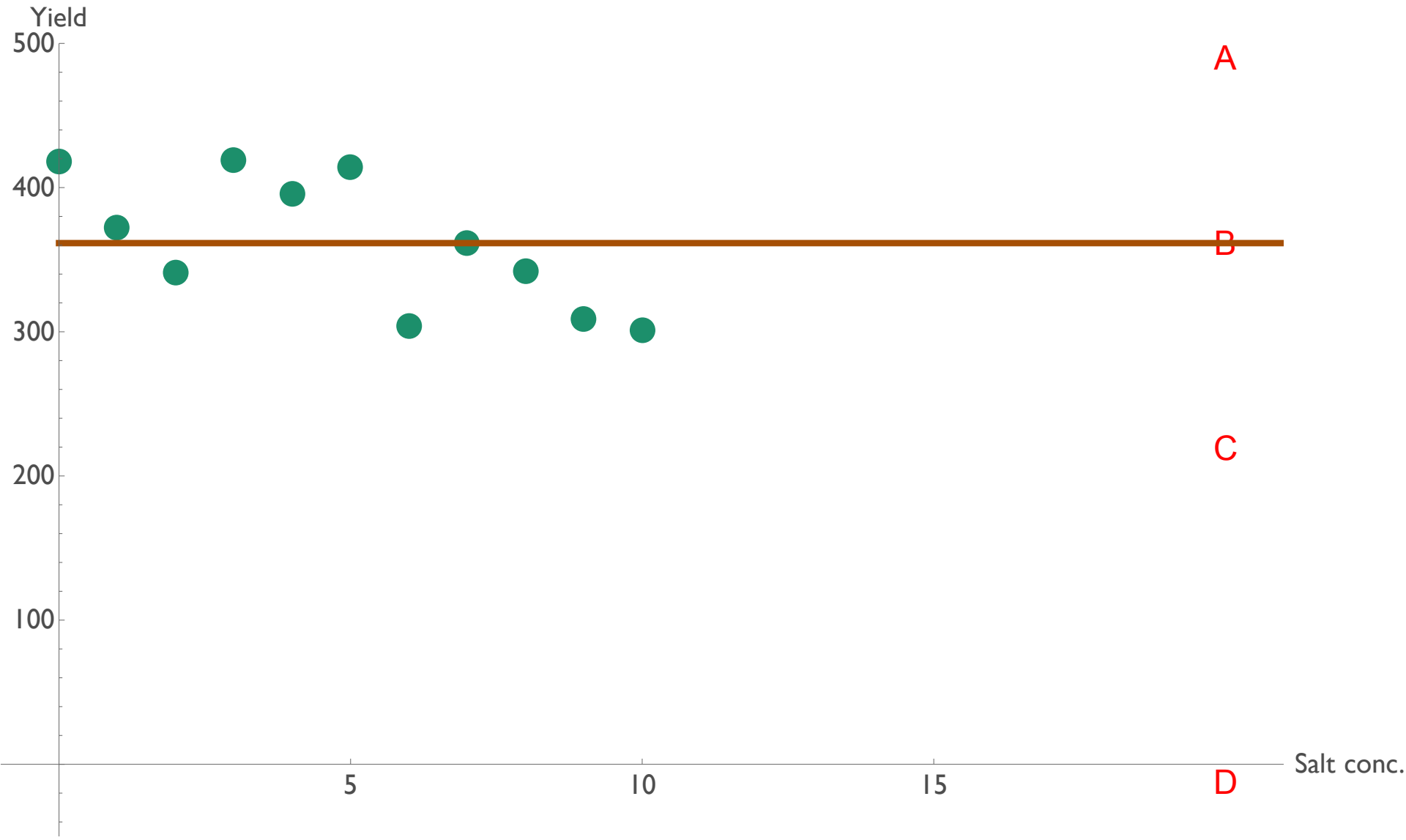

Overfitting

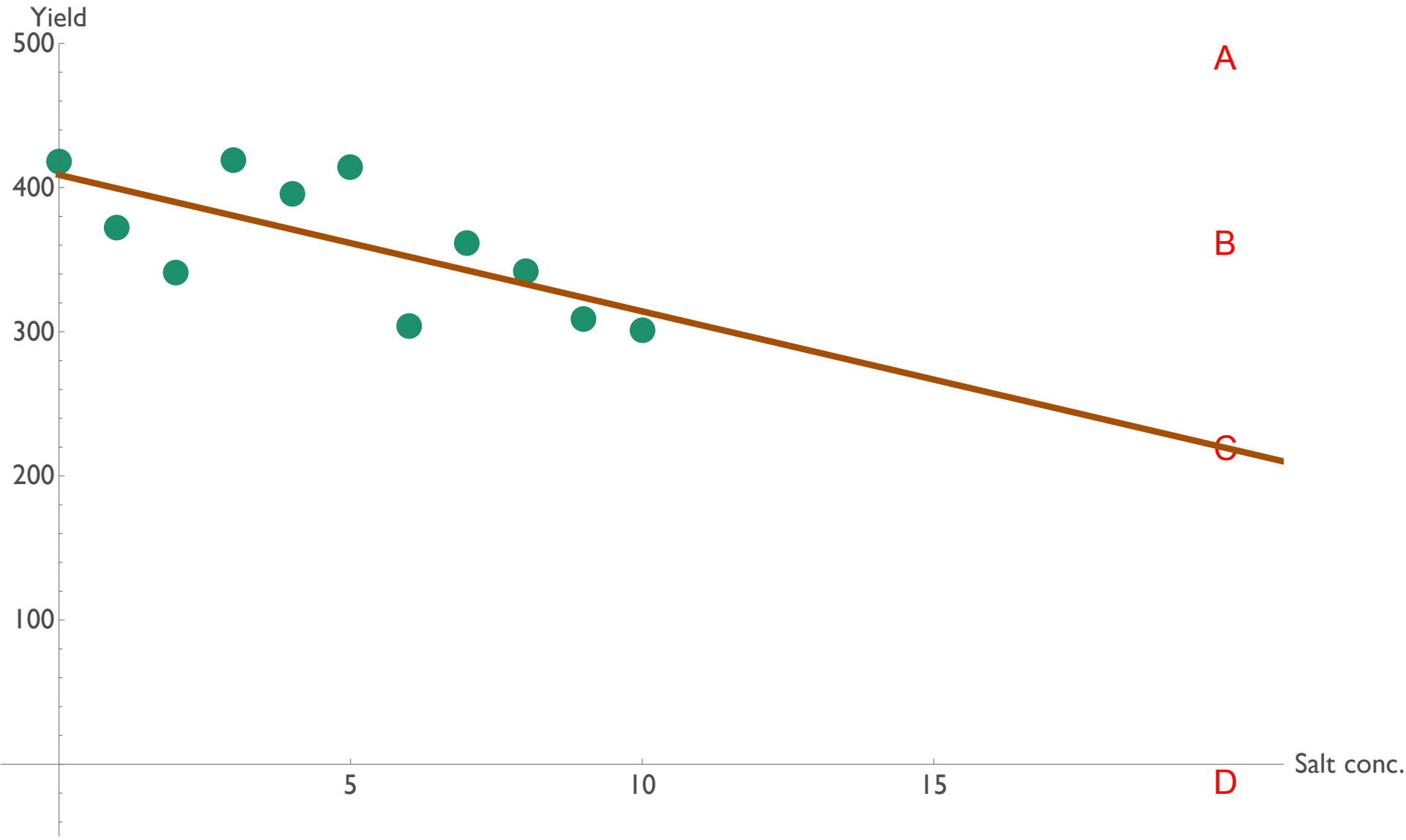
Overfitting

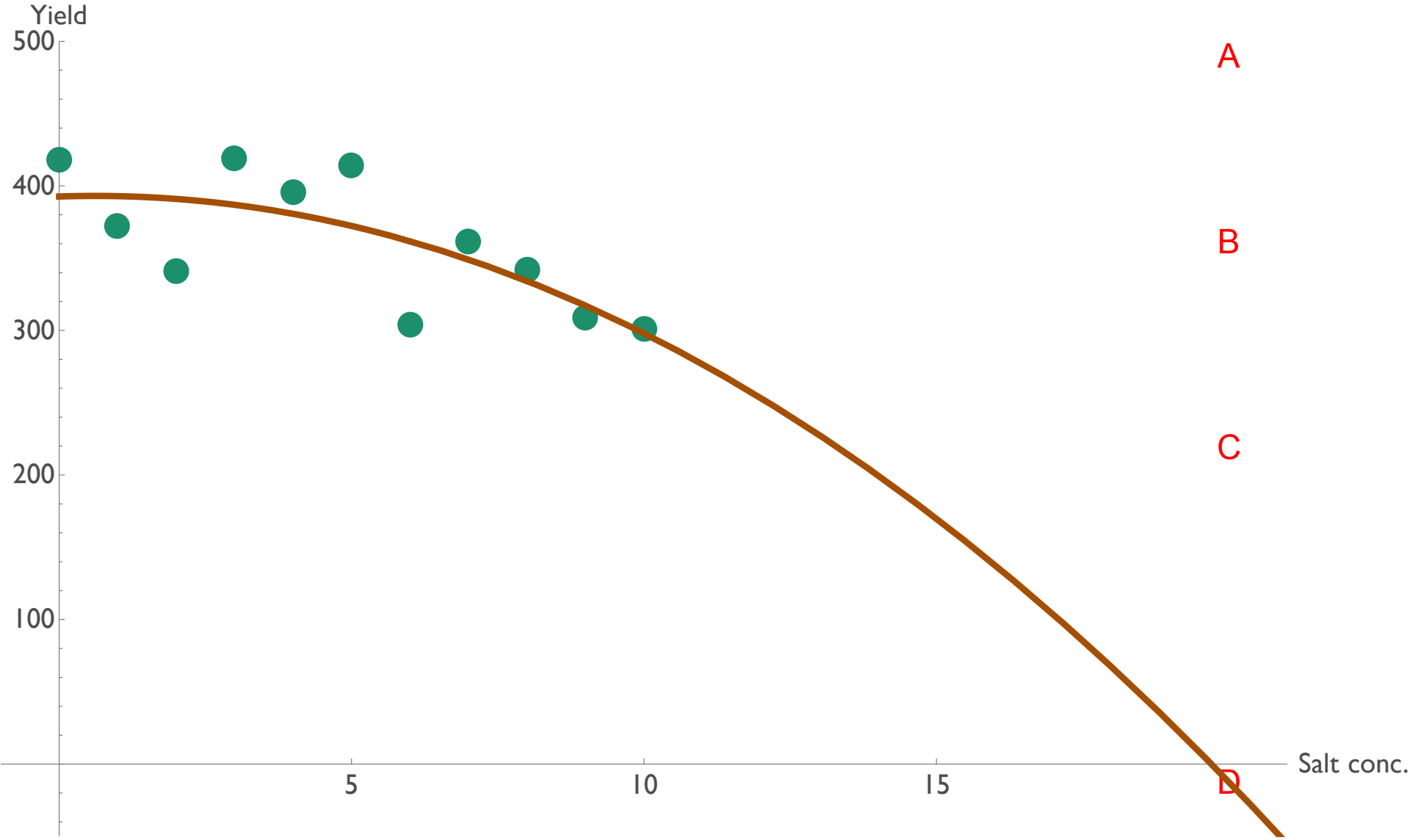
- If we have too many features, our model may fit some data really well but not handle other new examples very well at all
- Picture the situation below:

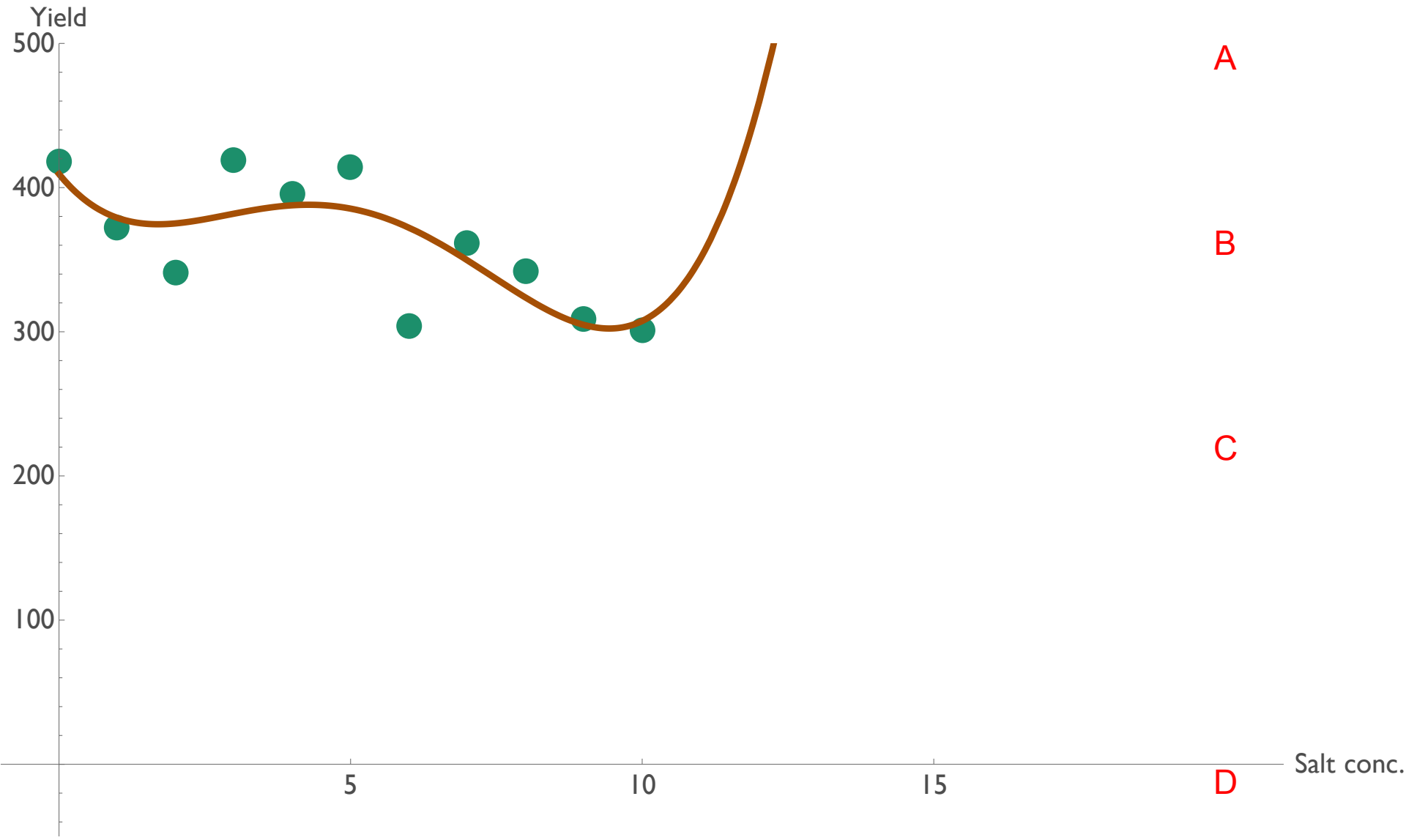


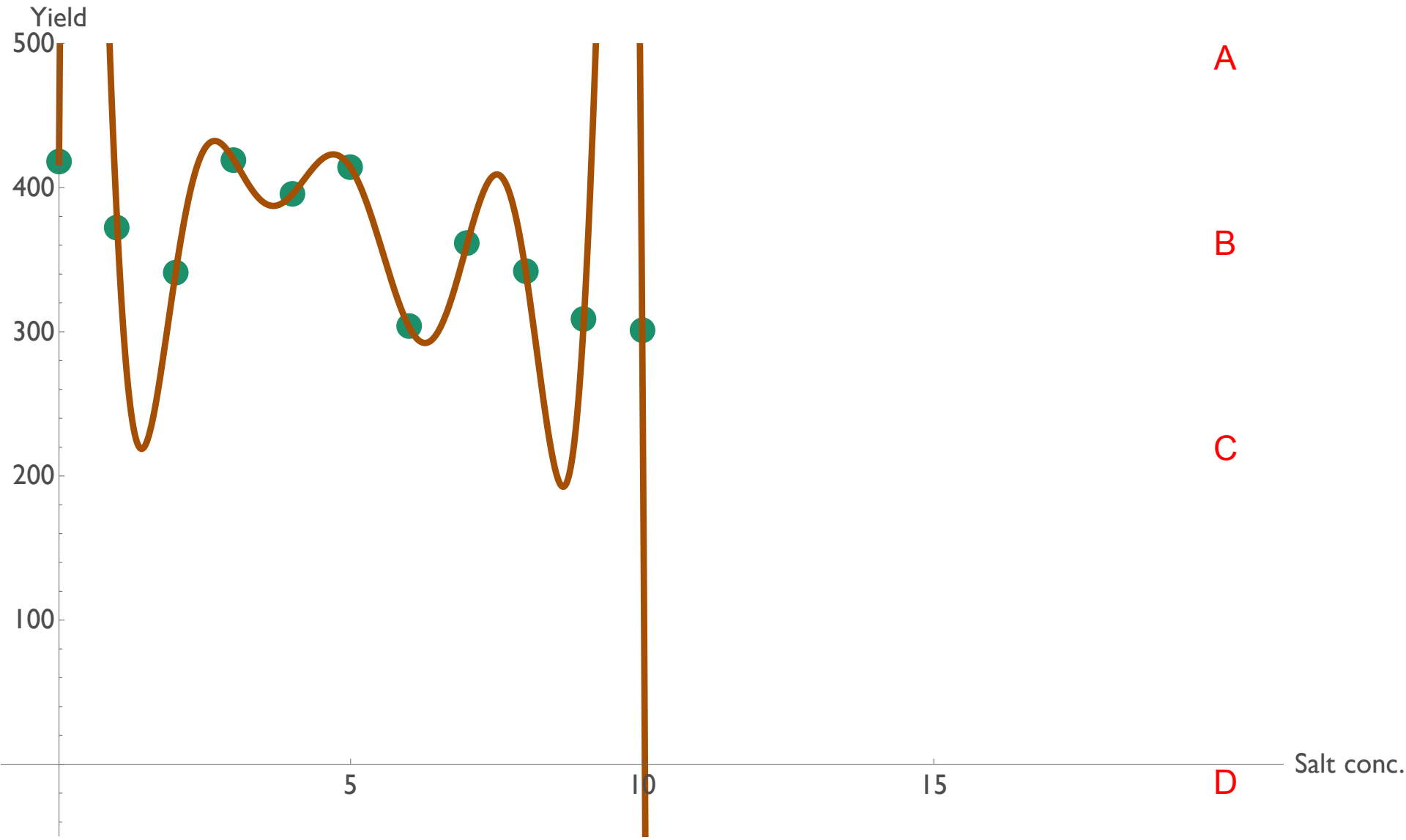

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

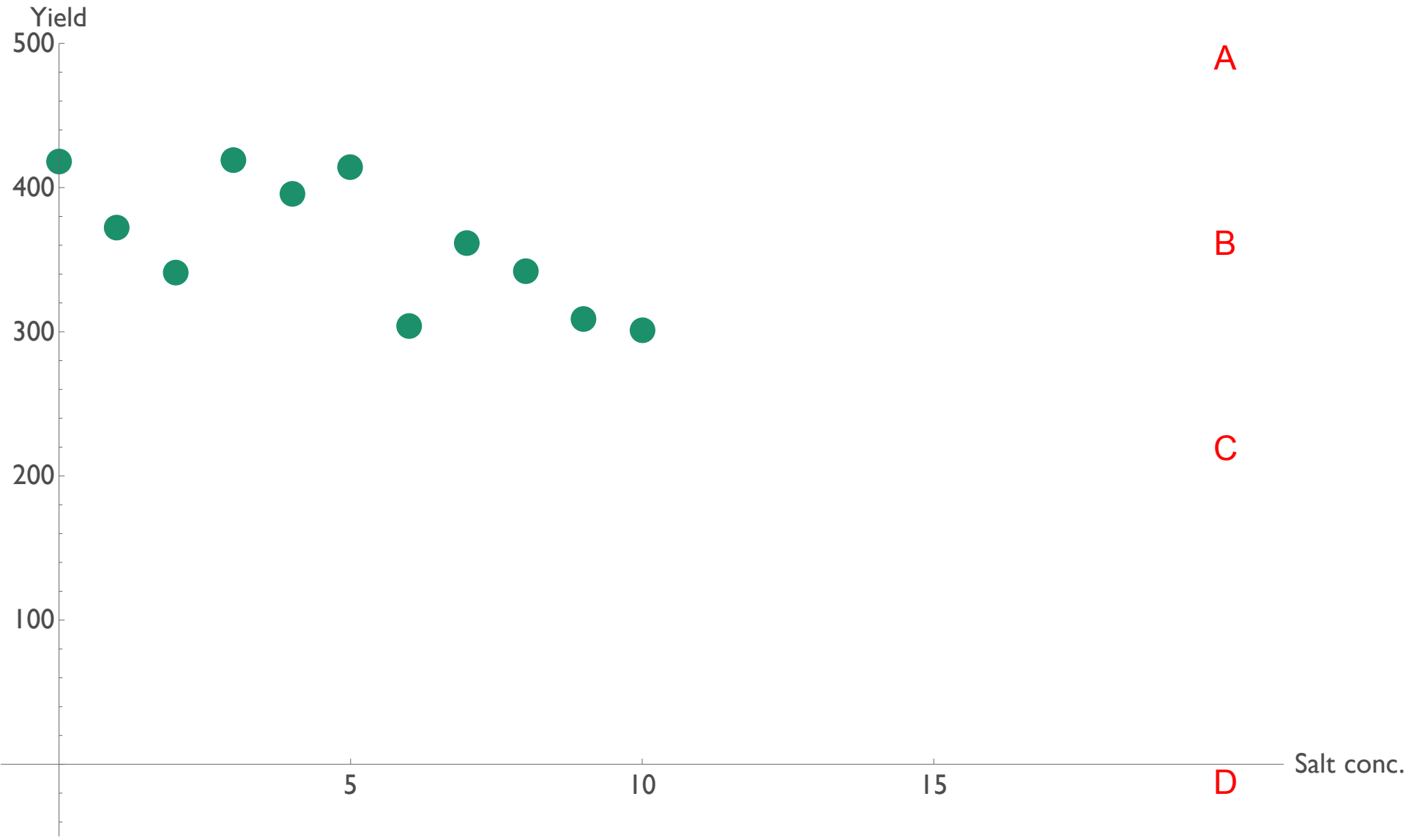

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

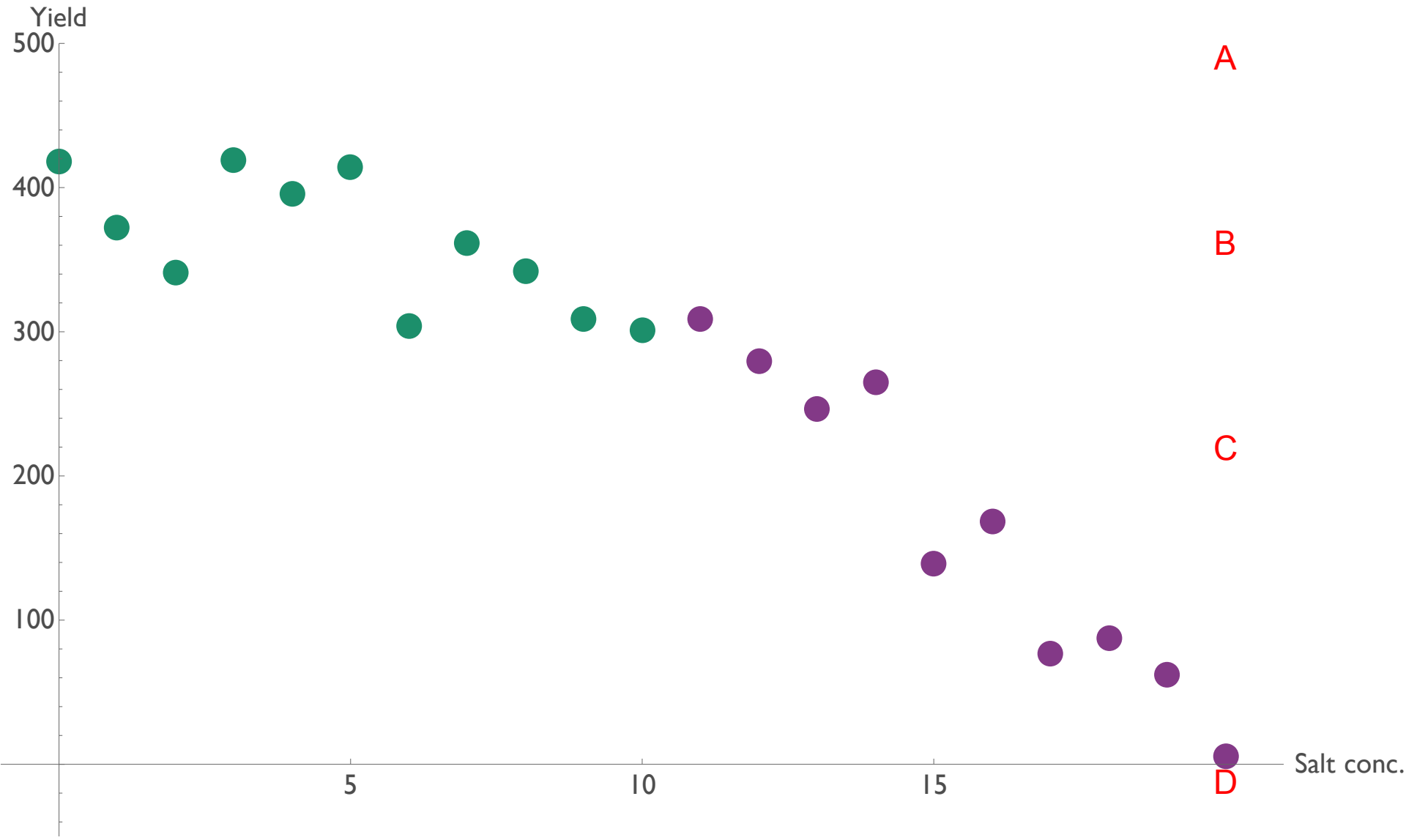

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

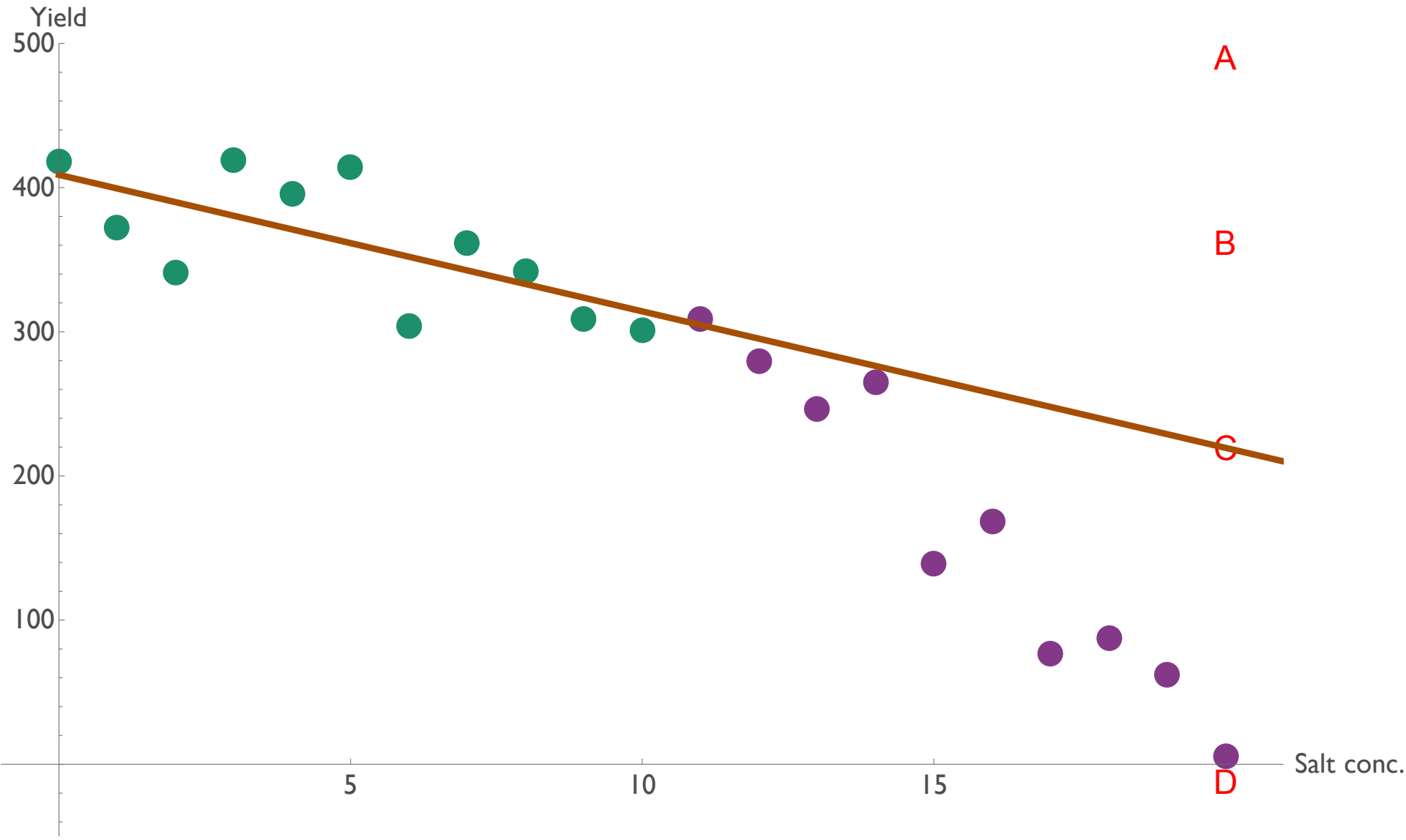

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

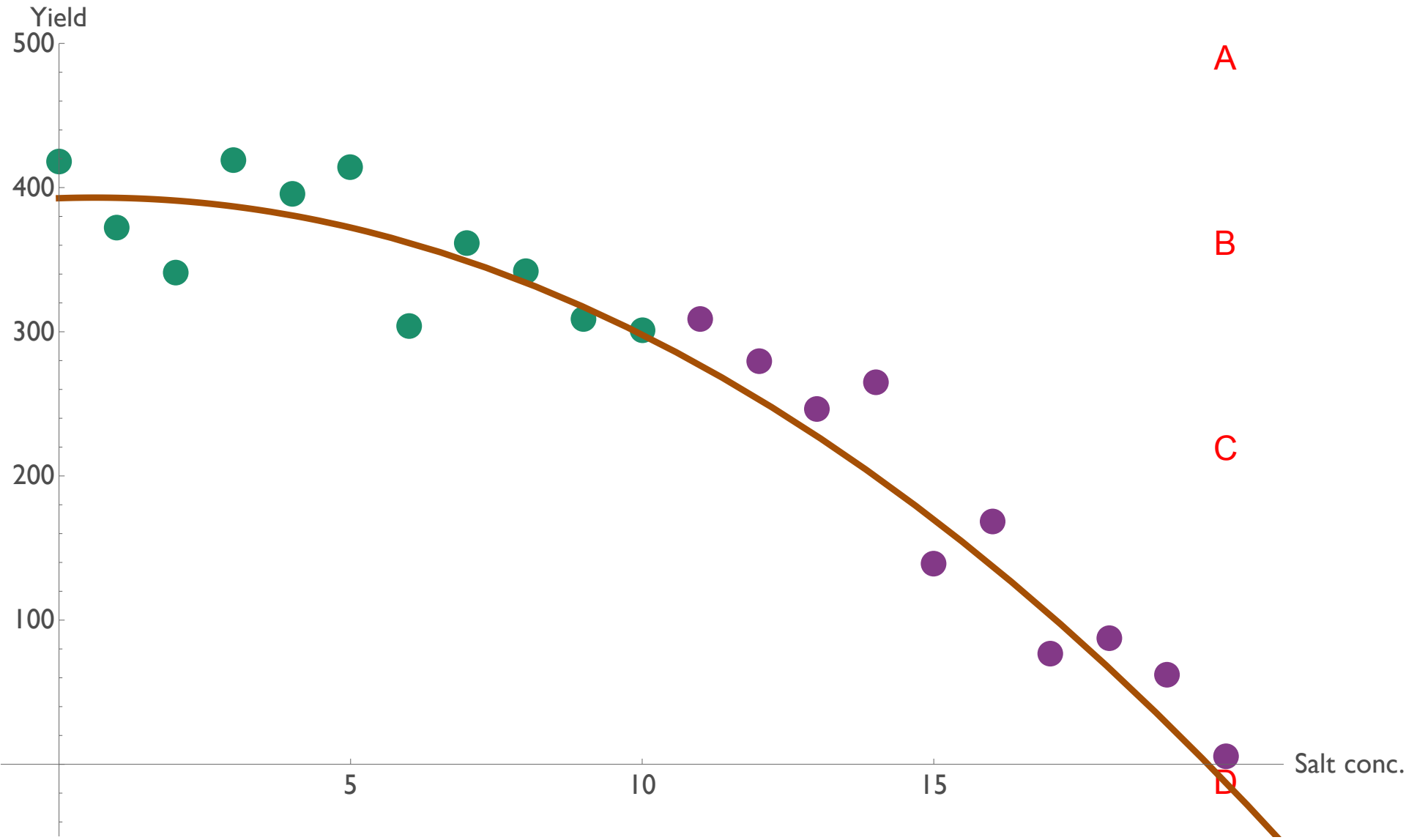

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D


Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

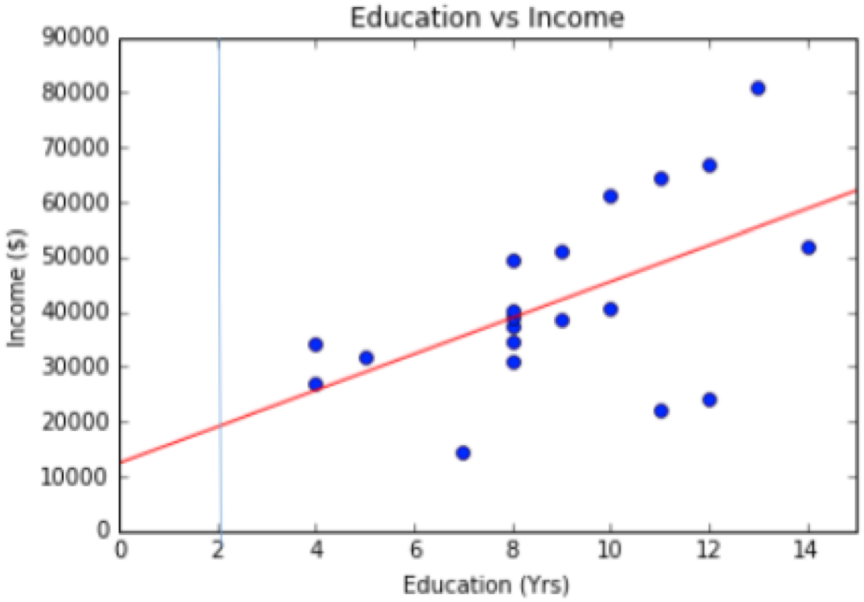

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

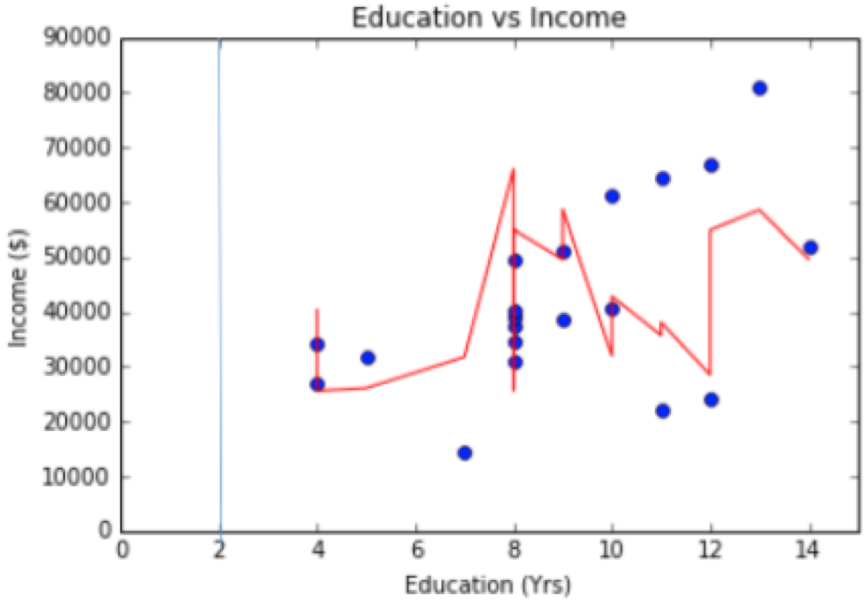

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D


Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

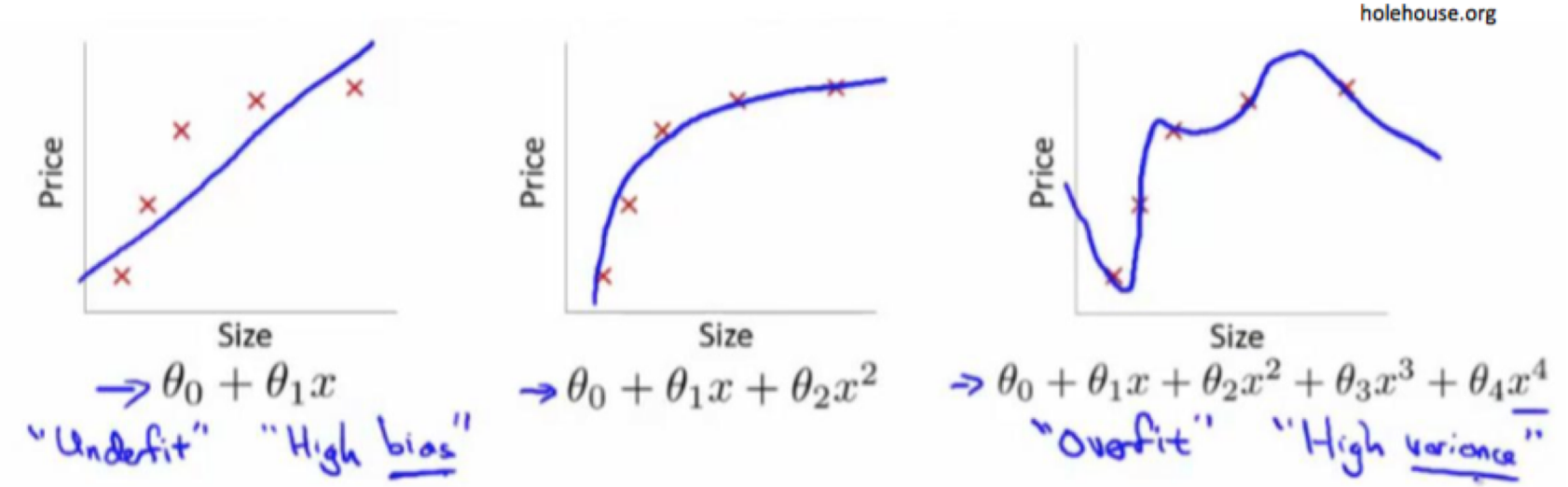

Yield
500
400
300
200
100
5
10
15
Salt conc.
A
B
C
D

Overfitting

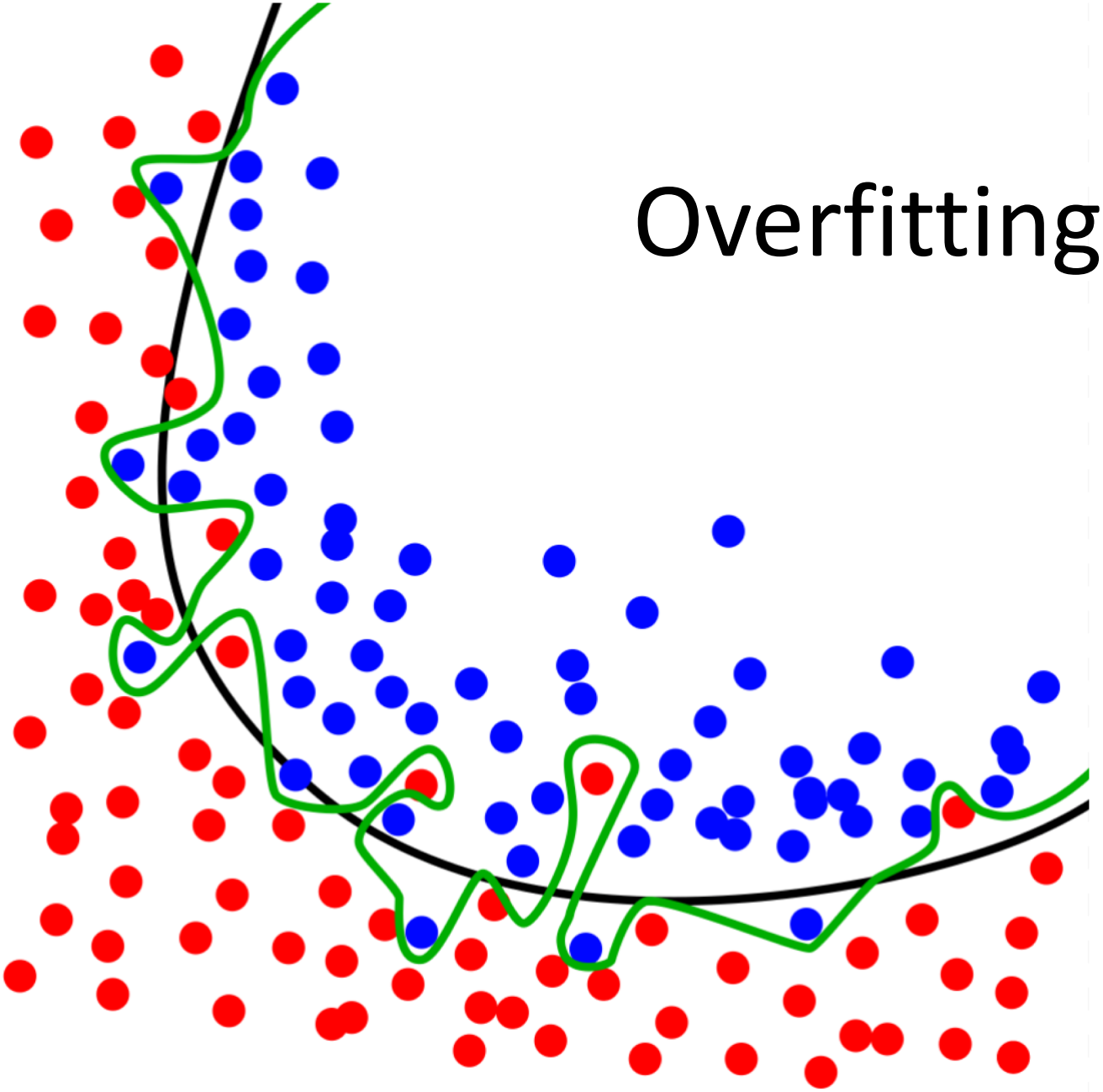




Overfitting (another example)



- Bias-Variance Tradeoff: “The problem of simultaneously minimizing two sources of error that prevent generalization.”


Overfitting

Key Issues in (Supervised) Machine Learning

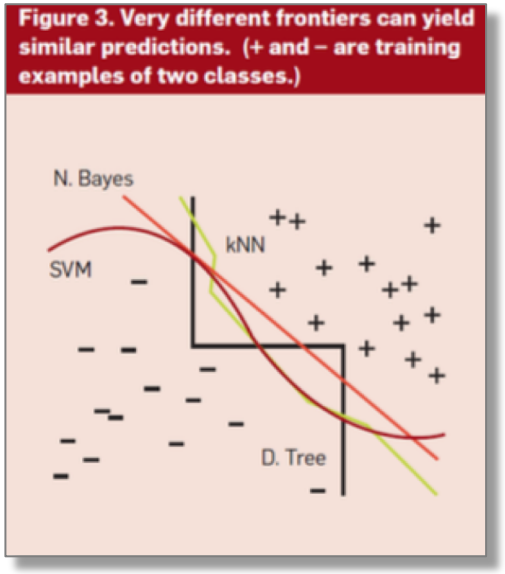
- The curse of dimensionality
  - Our intuition works well in up to three dimensions, but at that…?
  - ”Generalizing correctly becomes exponentially harder as the dimensionality (number of features) of the examples grows, because a fixed-size training set covers a dwindling fraction of the input space.”

Key Issues in (Supervised) Machine Learning

- Fast vs. exact solutions
- Feature engineering
  - ”Easily the most important factor is the features used.”
  - ”This is typically where most of the effort in a ML project goes. It is often also one of the most interesting parts, where intuition, creativity and “black art” are as important as the technical stuff.”

Key Issues in (Supervised) Machine Learning

- More Data Matters
  - "As a rule of thumb, a dumb algorithm with lots and lots of data beats a clever one with modest amounts of data.."
  - Different models may produce similar results – so start simple!



Key Issues in (Supervised) Machine Learning

- Ensembles work
  - Bagging: resample the training data to generate multiple data sets, and train classifiers on each one
  - Boosting: Focus on examples that are hard to learn
  - Stacking: Use models to learn from the outputs of other models

Key Issues in (Supervised) Machine Learning

- Interpretability is important
  - There is beauty in simplicity, and not just the abstract kind!
  - Interpretability is hard to measure, but often trumps other measures of performance

Key Issues in (Supervised) Machine Learning

- Summary
  - Don’t forget the theory
  - Generalization and overfitting
  - Feature Engineering
  - More Data Matters
  - Ensembles work
  - Interpretability is important

Outline going forward

- Design of ML Experiments
  - Generalization and overfitting
  - Training, testing, and validation
  - Cross-validation and bootstrap
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

Design of ML Experiments: Key Concepts

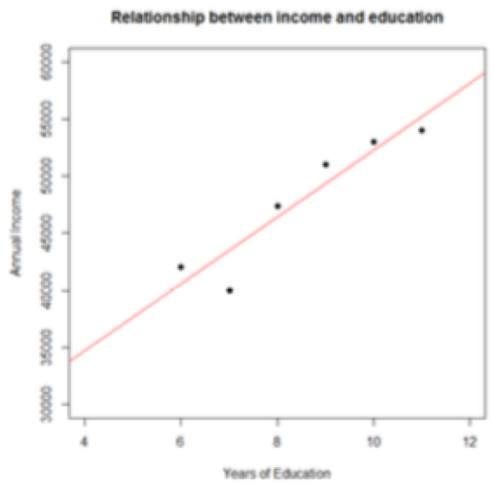
- Overfitting
- Train, test, held-out data
- Hyperparameters
- Accuracy, ROC, AUC, F-scores
- Lift curves
- Baselines
- Error Analysis
- Ablative Analysis

Generalization and Overfitting

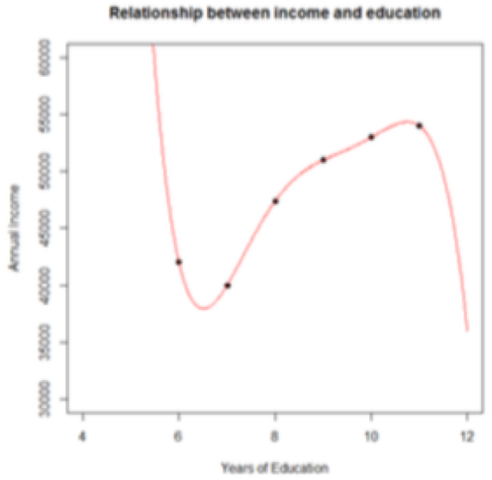
- Adding more features generally improves the "fit"of your model to your data
  - $Wagesi = \alpha + \beta Educationi + \epsilon i$
  - $Wages = \alpha + \beta Education + \beta\ Age + \beta\ Age2 + \beta\ Location + \cdots + \beta\ Hairy \epsilon$I
- In the limit, ask(# features) approaches or evenexceedsN(# observations)..
  - Our model can fit our data really, really well
  - If we have 4 observations, how about: $Wagesi = \pi 0i = 1 + \pi 1i = 2 + \pi 3i = 3 + \pi 4(i = 4)$

Generalization and Overfitting

- Overfitting: If we have too many features, our model may fit the training set very well, but fail to generalize to new examples



$wages_i = \alpha + \beta * educ_i + error_i$



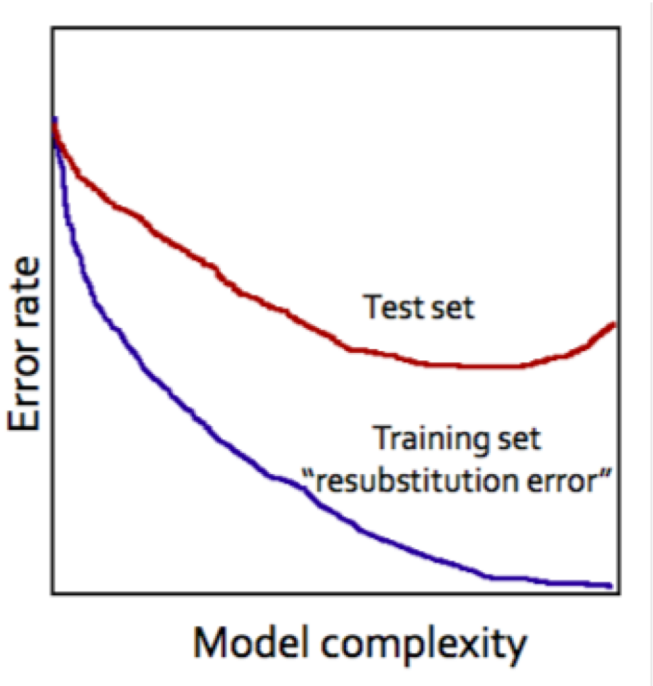
$wages_i = \alpha + \beta_1 * educ_i + \ldots + \beta_5 * educ_i^5 + error_i$

Outline going forward

- Design of ML Experiments
  - Generalization and overfitting
  - **Training, testing, and validation**
  - Cross-validation and bootstrap
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

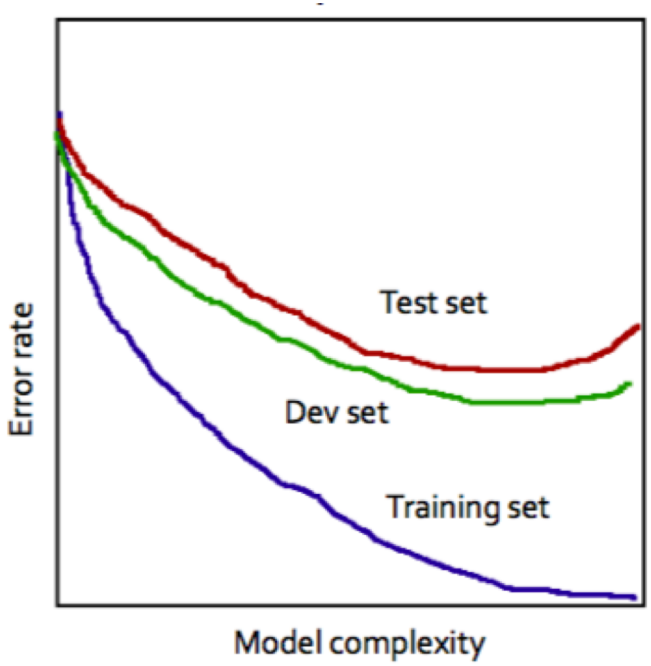
Training and Testing

- ML experiments typically separate data into a training set and a testing set
  - Model is fit on training set
  - Performance is measured on test set
- The test set helps avoid overfitting



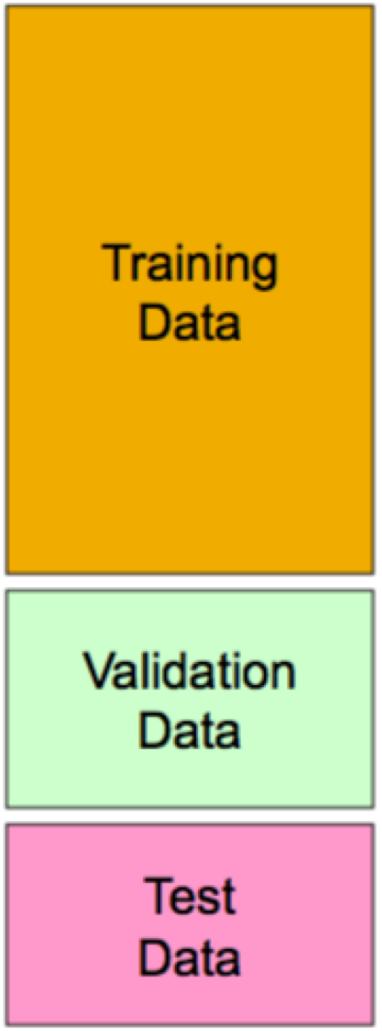
Validation (development) data

- Splitting into training + testing is often not enough
  - Hyperparameters must be chosen
  - Model selection, feature selection, etc.
  - Each time you look at the test set, you introduce bias (in yourself!)
- Validation data
  - A third split of the data
  - Aka “development” data
  - Used to fit model
- Measure and report performance on test set
  - Unseen until very end



A typical ML Experiment

- Data: labeled instances
  - Training set
  - Validation set
  - Test set
- Training
  - Estimate parameters on training set
  - Tune hyperparameters on validation set
  - Report results on test set
  - Anything short of this yields over-optimistic claims
- Evaluation
  - Many different metrics
  - Ideally, the criteria used to train the model should be closely related to those used to evaluate the model
- Statistical issues
  - Want a model which does well on test data
  - Overfitting: fits the training data closely, but doesn't generalize well
  - Error bars: want realistic (conservative) estimates of accuracy



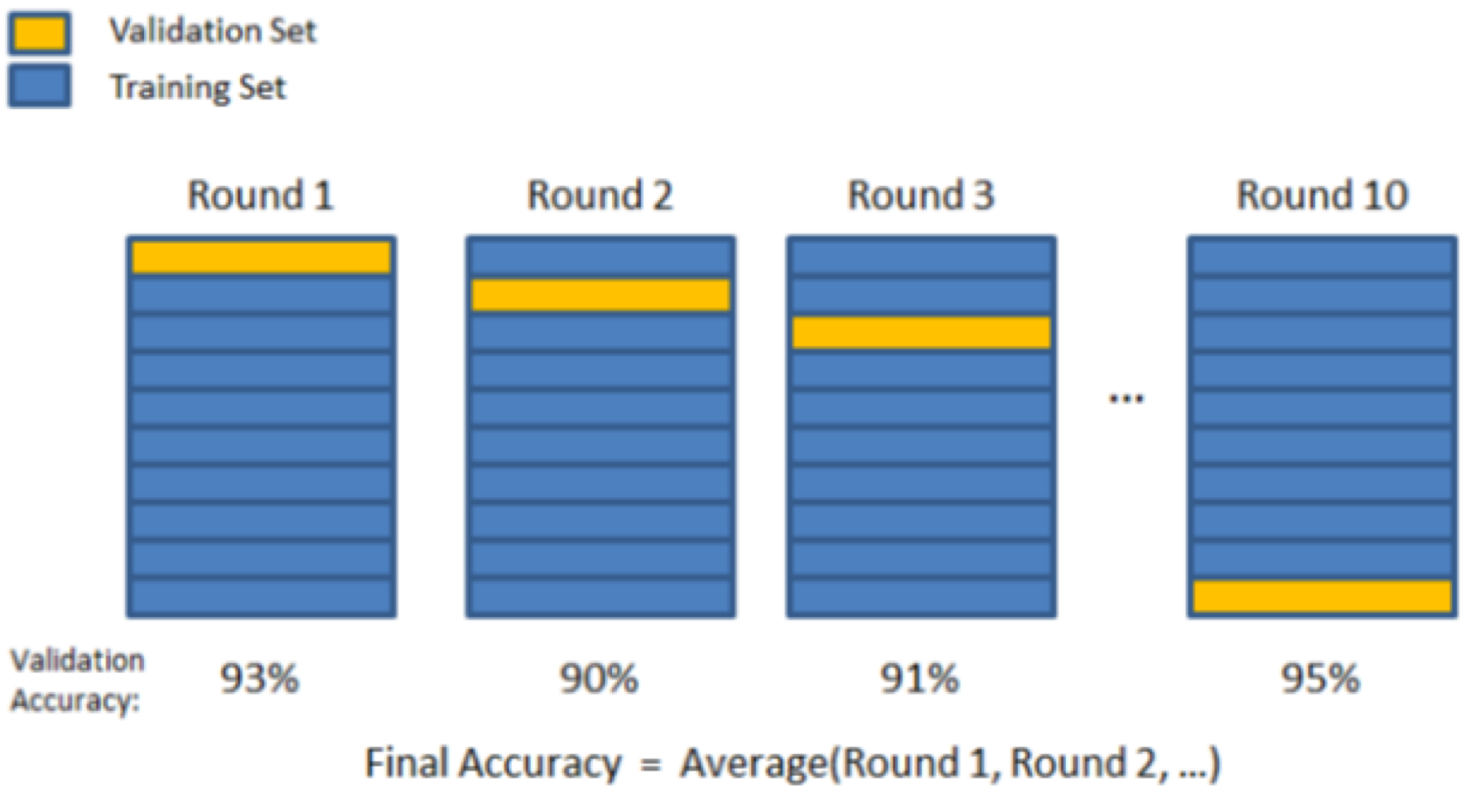
Outline going forward

- Design of ML Experiments
  - Generalization and overfitting
  - Training, testing, and validation
  - **Cross-validation and bootstrap**
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

Cross-validation

- Given unlimited data, it's easy to find new test data. But what if you have limited data?
  - Your "random" sample of training data may not be representative
  - GIGO: Garbage in, garbage out
- k-fold cross-validation
  - Randomly partition data into *k* equal size subsamples
  - Use each of *k* folds as validation set once
  - Average performance across *k* test runs

Cross-Validation



Cross-Validation

- Stratified cross-validation
  - Select folds so that the mean response value is approximately equal in all the folds, or so that some other parameter is balanced across folds
- Leave-one-out
  - Special case where K = N
  - Pro: deterministic, almost all data used each fold
  - Con: computationally intensive
  - Con: Can't stratify, can overfit
    - What is estimated error rate on truly random data?

Bootstrap

- Cross-Validation
  - Partitioning of data into *k* folds means each instance is used exactly once (either as train or test)
- Bootstrap
  - Instead, sample with replacement from data
  - Unsampled data become the validation set
  - Training data: 63.2% unique; validation: 36.8%
    - Probability that an instance is not picked = 1-(1/n)

$$\left(1-\frac{1}{n}\right)^{n} \cong e^{-1} = 0.368$$

Cross-Validation: a word of caution

- All of the above assume you **never, ever touch your test data**
- Even so, you stand the risk of overfitting
  - Over-use of cross-validation is another form of overfitting
  - Be careful about informal hyperparameters
    - Ng, A.Y.“Preventing “overfitting” ofcross-validation data.” ICML’97

Schedule

- April 17: Problem #2 DUE at midnight
- April 19: Quiz #1
- May 1: Problem Set #3 DUE