

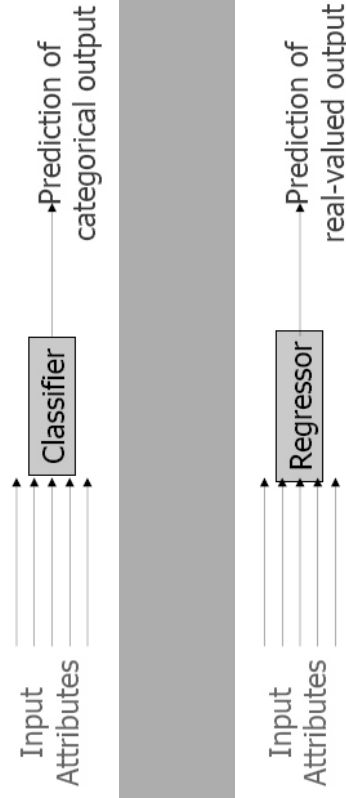
# Prediction (Classification, Regression)

Ryan Shaun Joazeiro de Baker

## Prediction

- Pretty much what it says
- A student is using a tutor right now. **Is he gaming the system or not?** (“attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material”)
- A student has used the tutor for the last half hour. **How likely is it that she knows the knowledge component in the next step?**
- A student has completed three years of high school. **What will be her score on the SAT-Math exam?**

## Two Key Types of Prediction



## Classification

- General Idea
- Canonical Methods
- Assessment
- Ways to do assessment wrong

## Classification

- There is something you want to predict (“the label”)
- The thing you want to predict is categorical
  - The answer is one of a set of categories, not a number
  - CORRECT/WRONG (sometimes expressed as 0,1)
  - HELP REQUEST/WORKED EXAMPLE REQUEST/ATTEMPT TO SOLVE
  - WILL DROP OUT/WON’T DROP OUT
  - WILL SELECT PROBLEM A,B,C,D,E,F, or G

## Classification

- Associated with each label are a set of “features”, which maybe you can use to predict the label

Skill	pknow	time	totalactions	right
ENTERINGGIVEN	0.704	9	1	WRONG
ENTERINGGIVEN	0.502	10	2	RIGHT
USEDIFFNUM	0.049	6	1	WRONG
ENTERINGGIVEN	0.967	7	3	RIGHT
REMOVECOEFF	0.792	16	1	WRONG
REMOVECOEFF	0.792	13	2	RIGHT
USEDIFFNUM	0.073	5	2	RIGHT
...				

## Classification

- The basic idea of a classifier is to determine which features, in which combination, can predict the label

Skill	pknow	time	totalactions	right
ENTERINGGIVEN	0.704	9	1	WRONG
ENTERINGGIVEN	0.502	10	2	RIGHT
USEDIFFNUM	0.049	6	1	WRONG
ENTERINGGIVEN	0.967	7	3	RIGHT
REMOVECOEFF	0.792	16	1	WRONG
REMOVECOEFF	0.792	13	2	RIGHT
USEDIFFNUM	0.073	5	2	RIGHT
...				

## Classification

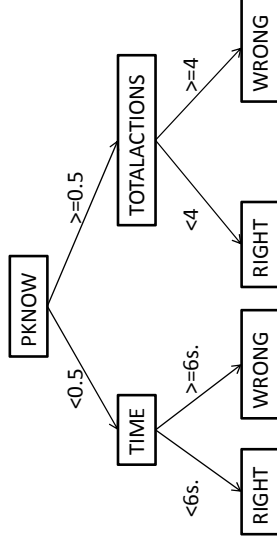
- Of course, usually there are more than 4 features
- And more than 7 actions/data points
- I’ve recently done analyses with 800,000 student actions, and 26 features
  - DataShop data

## Classification

- Of course, usually there are more than 4 features
- And more than 7 actions/data points
- I've recently done analyses with 800,000 student actions, and 26 features
  - DataShop data
- 5 years ago that would've been a lot of data
- These days, in the EDM world, it's just a medium-sized data set

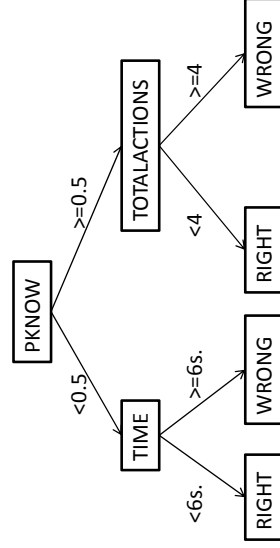
## Classification

- One way to classify is with a Decision Tree (like J48)



## Classification

- One way to classify is with a Decision Tree (like J48)



Skill	pknow	time	totalactions	right
COMPUTESLOPE	0.544	9	1	?

## Classification

- Another way to classify is with logistic regression
- Where  $\pi$  = probability of right
- $\text{Ln} \frac{\pi}{1-\pi} = 0.2\text{Time} + 0.8\text{Pknow} - 0.33\text{Totalactions}$

## And of course...

- There are lots of other classification algorithms you can use...
- SMO (support vector machine)
- KStar
- In your favorite Machine Learning package
  - WEKA
  - RapidMiner
  - KEEL

## How does it work?

- The algorithm finds the best model for predicting the actual data
- This might be the model which is most likely given the data
- Or the model for which the data is the most likely
- These are not always the same thing
- Which is right? It is a matter of much debate. 😊

How can you tell if  
a classifier is any good?

How can you tell if  
a classifier is any good?

- What about accuracy?
- $\frac{\text{\# correct classifications}}{\text{total number of classifications}}$
- 9200 actions were classified correctly, out of 10000 actions = 92% accuracy, and we declare victory.

What are some limitations of accuracy?

Biased training set

- What if the underlying distribution that you were trying to predict was:
- 9200 correct actions, 800 wrong actions
- And your model predicts that every action is correct
- Your model will have an accuracy of 92%
- Is the model actually any good?

What are some alternate metrics you could use?

What are some alternate metrics you could use?

- Kappa

$$\frac{(\text{Accuracy} - \text{Expected Accuracy})}{(1 - \text{Expected Accuracy})}$$

## Kappa

- Expected accuracy computed from a table of the form

	"Gold Standard" Label Category 1	"Gold Standard" Label Category 2
ML Label Category 1	Count	Count
ML Label Category 2	Count	Count

- For actual formula, see your favorite stats package; or read a stats text; or I have an excel spreadsheet I can share with you

## Comparison

- Kappa
  - easier to compute
  - works for an unlimited number of categories
  - wacky behavior when things are worse than chance
  - difficult to compare two kappas in different data sets (K=0.6 is not always better than K=0.5)

What are some alternate metrics you could use?

- A'
- The probability that if the model is given an example from each category, it will accurately identify which is which
- Equivalent to area under ROC curve

## Comparison

- A'
  - more difficult to compute
  - only works for two categories (without complicated extensions)
  - meaning is invariant across data sets (A'=0.6 is always better than A'=0.55)
  - very easy to interpret statistically

What data set should you generally test on?

- A vote...

What data set should you generally test on?

- The data set you trained your classifier on
- A data set from a different tutor
- Split your data set in half, train on one half, test on the other half
- Split your data set in ten. Train on each set of 9 sets, test on the tenth. Do this ten times.
- Votes?

What data set should you generally test on?

- The data set you trained your classifier on
- A data set from a different tutor
- Split your data set in half, train on one half, test on the other half
- Split your data set in ten. Train on each set of 9 sets, test on the tenth. Do this ten times.

- What are the benefits and drawbacks of each?

The dangerous one (though still sometimes OK)

- The data set you trained your classifier on
- If you do this, there is serious danger of overfitting

The dangerous one  
(though still sometimes OK)

- You have ten thousand data points.
- You fit a parameter for each data point.
- “If data point 1, RIGHT. If data point 78, WRONG...”
- Your accuracy is 100%
- Your kappa is 1
- Your model will neither work on new data, nor will it tell you anything.

The dangerous one  
(though still sometimes OK)

- The data set you trained your classifier on
- When might this one still be OK?

K-fold cross validation

- Split your data set in half, train on one half, test on the other half
- Split your data set in ten. Train on each set of 9 sets, test on the tenth. Do this ten times.
- Generally preferred method, when possible

A data set from a different tutor

- The most stringent test
- When your model succeeds at this test, you know you have a good/general model
- When it fails, it’s sometimes hard to know why

## An interesting alternative

- Leave-out-one-tutor-cross-validation (cf. Baker, Corbett, & Koedinger, 2006)
  - Train on data from 3 or more tutors
  - Test on data from a different tutor
  - (Repeat for all possible combinations)
- Good for giving a picture of how well your model will perform in new lessons

## Statistical testing

## Statistical testing

- Let's say you have a classifier A. It gets kappa = 0.3. Is it actually better than chance?
- Let's say you have two classifiers, A and B. A gets kappa = 0.3. B gets kappa = 0.4. Is B actually better than A?

## Statistical tests

- Kappa can generally be converted to a chi-squared test
  - Just plug in the same table you used to compute kappa, into a statistical package
  - Or I have an Excel spreadsheet I can share w/ you
- A' can generally be converted to a Z test
  - I also have an Excel spreadsheet for this (or see Fogarty, Baker, & Hudson, 2005)

## A quick example

- Let's say you have a classifier A. It gets kappa = 0.3. Is it actually better than chance?
- 10,000 data points from 50 students

## Example

- Kappa -> Chi-squared test
  - You plug in your 10,000 cases, and you get
- Chi-sq(1,df=10,000)=3.84, two-tailed p=0.05
- Time to declare victory?

## Example

- Kappa -> Chi-squared test
  - You plug in your 10,000 cases, and you get
- Chi-sq(1,df=10,000)=3.84, two-tailed p=0.05
- No, I did something wrong here

## Non-independence of the data

- If you have 50 students
- It is a violation of the statistical assumptions of the test to act like their 10,000 actions are independent from one another
- For student A, action 6 and 7 are not independent from one another (actions 6 and 48 aren't independent either)
- Why does this matter?
- Because treating the actions like they are independent is likely to make differences seem more statistically significant than they are

So what can you do?

So what can you do?

1. Compute % right for each student, actual and predicted, and compute the correlation (then test the statistical significance of the correlation)
  - Throws out some data (so it's overly conservative)
  - May miss systematic error
2. Set up a logistic regression  
Prob Right = Student + Model Prediction

## Regression

- General Idea
- Canonical Methods
- Assessment
- Ways to do assessment wrong

## Regression

- There is something you want to predict (“the label”)
- The thing you want to predict is numerical
  - Number of hints student requests
  - How long student takes to answer
  - What will the student’s test score be

## Regression

- Associated with each label are a set of “features”, which maybe you can use to predict the label

Skill	pknow	time	totalactions	numhints
ENTERINGGIVEN	0.704	9	1	0
ENTERINGGIVEN	0.502	10	2	0
USEDIFFNUM	0.049	6	1	3
ENTERINGGIVEN	0.967	7	3	0
REMOVECOEFF	0.792	16	1	1
REMOVECOEFF	0.792	13	2	0
USEDIFFNUM	0.073	5	2	0
....				

## Regression

- The basic idea of regression is to determine which features, in which combination, can predict the label’s value

Skill	pknow	time	totalactions	numhints
ENTERINGGIVEN	0.704	9	1	0
ENTERINGGIVEN	0.502	10	2	0
USEDIFFNUM	0.049	6	1	3
ENTERINGGIVEN	0.967	7	3	0
REMOVECOEFF	0.792	16	1	1
REMOVECOEFF	0.792	13	2	0
USEDIFFNUM	0.073	5	2	0
....				

## Linear Regression

- The most classic form of regression is linear regression

$$\text{Numhints} = 0.12 * \text{Pknow} + 0.932 * \text{Time} - 0.11 * \text{Totalactions}$$

Skill	pknow	time	totalactions	numhints
COMPUTESLOPE	0.544	9	1	?

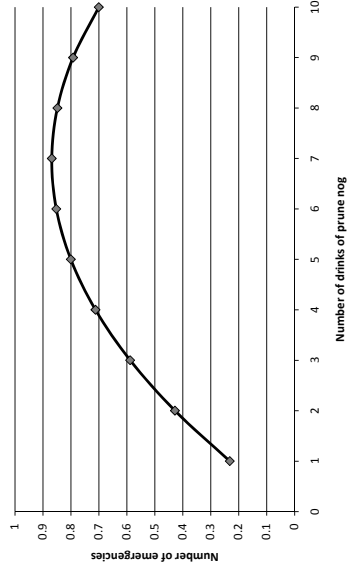
## Linear Regression

- Linear regression only fits linear functions (except when you apply transforms to the input variables... but this is more common in hand modeling in stats packages than in data mining/machine learning)

## Linear Regression

- However...
- It is blazing fast
- It is often more accurate than more complex models, particularly once you cross-validate
  - Machine Learning's "Dirty Little Secret"
- It is feasible to understand your model (with the caveat that the second feature in your model is in the context of the first feature, and so on)

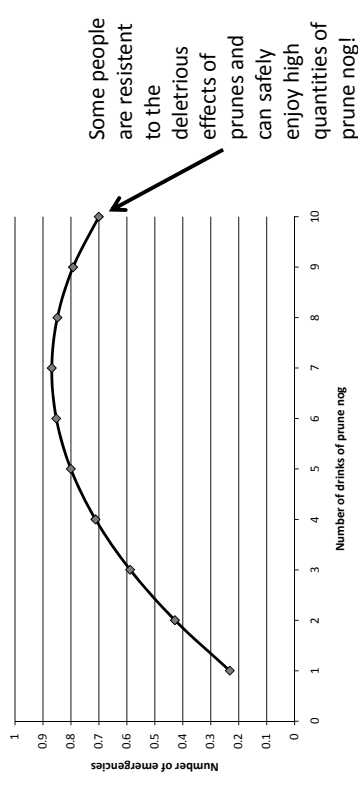
## Data



## Example of Caveat

- Let's study the classic example of drinking too much prune nog\*, and having an emergency trip to the washroom
  - \* Seen in English translation on restaurant menu
  - \*\* I tried it, ask offline if curious

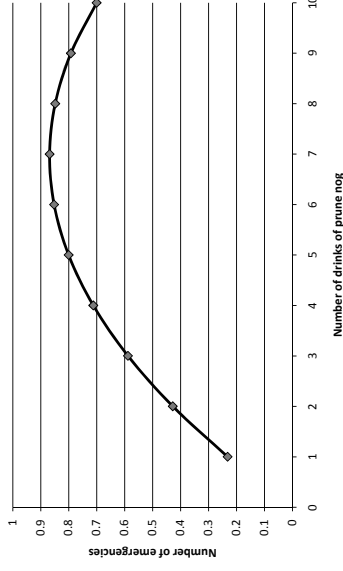
## Data



## Actual Function

- Probability of “emergency” =  
 $0.25 * \# \text{ Drinks of nog last 3 hours}$   
 $- 0.018 * (\text{Drinks of nog last 3 hours})^2$
- But does that actually mean that  
(Drinks of nog last 3 hours)<sup>2</sup> is associated with  
less “emergencies”?

## Example of Caveat

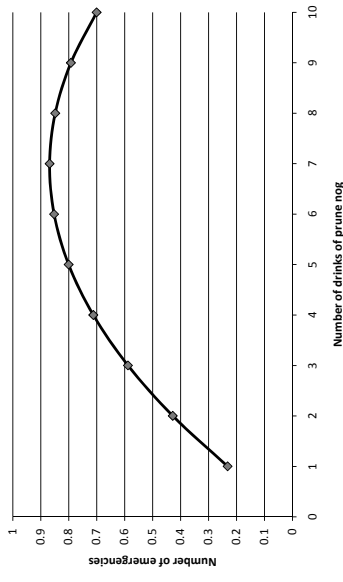


- (Drinks of nog last 3 hours)<sup>2</sup> is actually  
positively correlated with emergencies!  
—  $r=0.59$

## Actual Function

- Probability of “emergency” =  
 $0.25 * \# \text{ Drinks of nog last 3 hours}$   
 $- 0.018 * (\text{Drinks of nog last 3 hours})^2$
- But does that actually mean that  
(Drinks of nog last 3 hours)<sup>2</sup> is associated with  
less “emergencies”?
- No!

## Example of Caveat



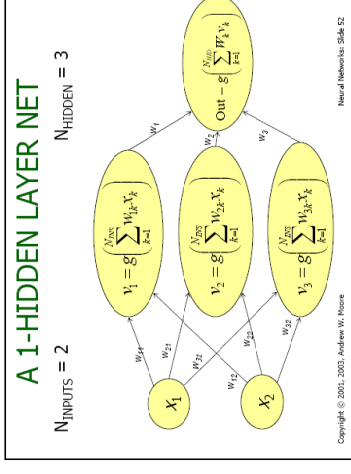
- The relationship is only in the negative  
direction when (Drinks of nog last 3 hours) is  
already in the model...

## Example of Caveat

- So be careful when interpreting linear regression models (or almost any other type of model)

## Neural Networks

- Another popular form of regression is neural networks (called Multilayer Perceptron in Weka)



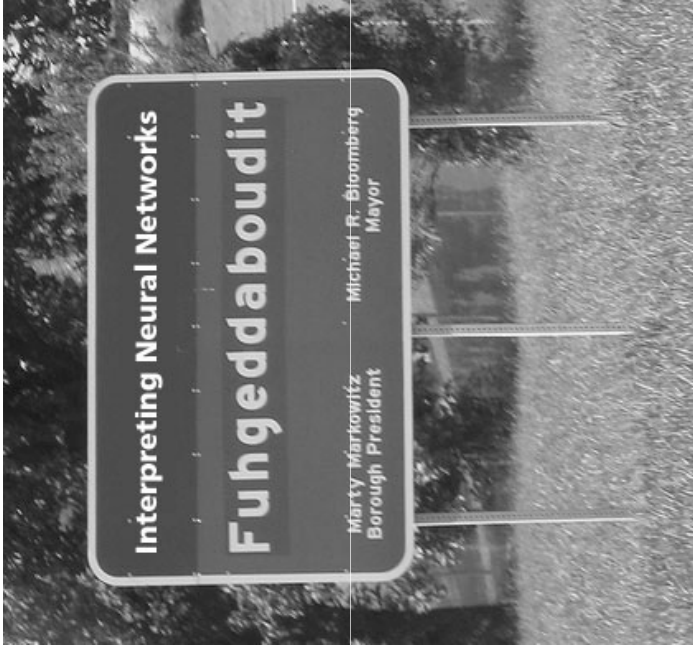
This image courtesy of Andrew W. Moore, Google  
<http://www.cs.cmu.edu/~awm/tutorials>

## Neural Networks

- Neural networks can fit more complex functions than linear regression
- It is usually near-to-impossible to understand what the heck is going on inside one

## In fact

- The difficulty of interpreting non-linear models is so well known, that New York City put up a road sign about it



## And of course...

- There are lots of fancy regressors in any Machine Learning package (like Weka)
- SMOReg (support vector machine)
- PaceRegression
- And so on

How can you tell if  
a regression model is any good?

- How can you tell if  
a regression model is any good?
- Correlation is a classic method
  - (Or its cousin  $r^2$ )

What data set should you generally test on?

- The data set you trained your classifier on
- A data set from a different tutor
- Split your data set in half, train on one half, test on the other half
- Split your data set in ten. Train on each set of 9 sets, test on the tenth. Do this ten times.
- **Any differences from classifiers?**

What are some stat tests you could use?

What about?

- Take the correlation between your prediction and your label
- Run an F test
- So  
 $F(1,9998)=50.00, p<0.0000000000001$

What about?

- Take the correlation between your prediction and your label
- Run an F test
- So  
 $F(1,9998)=50.00, p<0.0000000000001$
- All cool, right?

## As before...

- You want to make sure to account for the non-independence between students when you test significance
- An F test is fine, just include a student term

## As before....

- You want to make sure to account for the non-independence between students when you test significance
- An F test is fine, just include a student term (but note, your regressor itself should not predict using student as a variable... unless you want it to only work in your original population)

## Alternatives

- Bayesian Information Criterion (Raftery, 1995)
- Makes trade-off between goodness of fit and flexibility of fit (number of parameters)
- i.e. Can control for the number of parameters you used and thus adjust for overfitting
- Said to be statistically equivalent to k-fold cross-validation
  - Under common conditions, such as data set of infinite size

## How can you make your detector better?

- Let's say you create a detector
- But its goodness is “not good enough”

## Towards a better detector

- Try different algorithms
- Try different algorithm options
- Create new features

## The most popular choice

- **Try different algorithms**
- Try different algorithm options
- Create new features

## The most popular choice

- **Try different algorithms**
- Try different algorithm options
- Create new features
- EDM regularly gets submissions where the author tried 30 algorithms in Weka and presents the “best” one
  - Usually messing up statistical independence in the process
  - This is also known as “overfitting”

## My preferred choice

- Try different algorithms
- Try different algorithm options
- **Create new features**

## Repeatedly makes a bigger difference

- Baker et al, 2004 to Baker et al, 2008
- D’Mello et al, 2008 to D’Mello et al, 2009
- Baker, 2007 to Centinas et al, 2009

## Which features should you create?

- An art
- Many EDM researchers have a “favorite” set of features they re-use across analyses

## My favorite features

- Used to model
- Gaming the system (Baker, Corbett, & Koedinger, 2004; Baker et al, 2008)
- Off-task behavior (Baker, 2007)
- Careless slipping (Baker, Corbett, & Aleven, 2008)

## Details about the transaction

- The tutoring software’s assessment of the action
  - Correct
  - Incorrect and indicating a known bug (procedural misconception)
  - Incorrect but not indicating a known bug
  - Help request
- The type of interface widget involved in the action
  - pull-down menu, typing in a string, typing in a number, plotting a point, selecting a checkbox
- Was this the student’s first attempt to answer (or obtain help) on this problem step?

## Knowledge assessment

- Probability student knew the skill (Bayesian Knowledge Tracing)
- Did students know this skill before starting the lesson? (Bayesian Knowledge Tracing)
- Was the skill largely not learned by anyone? (Bayesian Knowledge Tracing)
- Was this the student's first attempt at the skill?

## Time

- How many seconds the action took.
- The time taken for the action, expressed in terms of the number of standard deviations this action's time was faster or slower than the mean time taken by all students on this problem step, across problems.
- The time taken in the last 3, or 5, actions, expressed as the sum of the numbers of standard deviations each action's time was faster or slower than the mean time taken by all students on that problem step, across problems. (two variables)
- How many seconds the student spent on each opportunity to practice the primary skill involved in this action, averaged across problems.

## Note

- The time taken in the last 3, or 5, actions
- 3 and 5 are magic numbers
  - I've found them more useful in my analyses than 2,4,6, but why?
  - Not really sure

## Details of Previous Interaction

- The total number of times the student has gotten this specific problem step wrong, across all problems. (includes multiple attempts within one problem)
- What percentage of past problems the student made errors on this problem step in
- The number of times the student asked for help or made errors at this skill, including previous problems.
- How many of the last 5 actions involved this problem step.
- How many times the student asked for help in the last 8 actions.
- How many errors the student made in the last 5 actions.

## Willing to share

- I have rather... un-commented... java code that distills these features automatically
- I'd be happy to share it with anyone here...

## Are these features the be-all-and-end-all?

- Definitely not
- Other people have other feature sets (see Walonoski & Heffernan, 2006; Amershi & Conati, 2007; Arroyo & Woolf, 2006; D'Mello et al, 2009 for some other nice examples, for instance)
- And it always is beneficial to introspect about your specific problem, and try new possibilities

## One mistake to avoid

- Predicting the past from the future
- One sees this a surprising amount
- Can be perfectly valid for creating training set labels, but should not be used as predictive features
  - Can't be used in real life!