CSE 454
Advanced Internet Systems

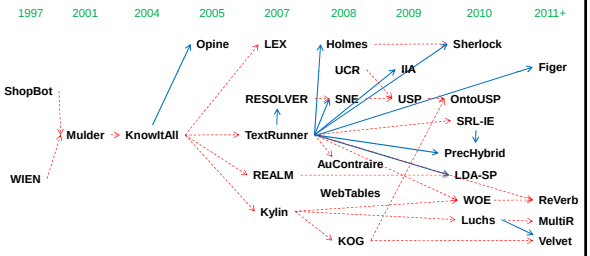
Machine Learning for Extraction

Dan Weld

Logistics

- Project Warm-Up
  - Due this Sunday
- Computing
  - $100 EC2 credit per student
- Team Selection
  - Topic survey later this week

An (Incomplete) Timeline of UW MR Systems



Perspective

- Crawling the Web
- Inverted indices
- Query processing
- Pagerank computation & ranking
- Information extraction
- Search UI
- Computational advertising
- Security & malware
- Social systems

Perspective

- Crawling the Web
- Inverted indices
- Query processing
- Pagerank computation & ranking
- **Information extraction**
- Search UI
- Computational advertising
- Security & malware
- Social systems

Today's Outline

- Supervised Learning – Compact Introduction
  - Learning as Function Approximation
  - Need for Bias
  - Overfitting
  - Bias / Variance Tradeoff
  - Loss Functions; Regularization; Learning as Optimization
  - Curse of Dimensionality
  - Logistic Regression
- IE as Supervised Learning
- Features for IE

Terminology

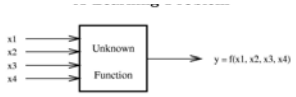
- Examples
  - Features
  - Labels
- Training Set
- Validation Set
- Test Set

Input: { …<X_1, …, X_k, Y>…}

Output: F: **X** → Y h: **X** → Y (hypothesis)

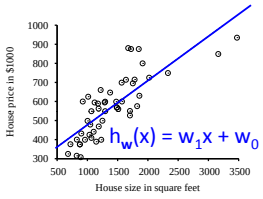
Objective: Minimize error of h on (unseen) test examples

Learning is Function Approximation

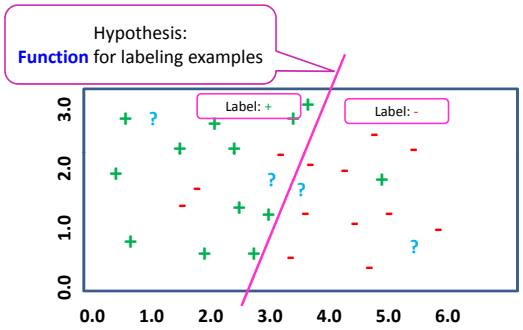


Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Linear Regression

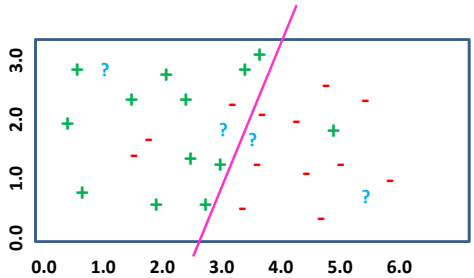


Classifier: Y (Range of F) is Discrete



Objective: Minimize error on ***test*** examples

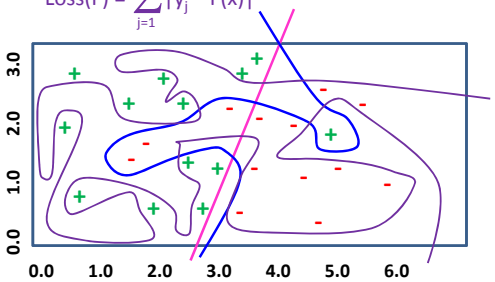
So… How good is this hypothesis?



Objective: Minimize error on test examples

Only know ***training data***, so minimize error on that

$$\text{Loss}(F) = \sum_{j=1}^{n} |y_j - F(x)|$$



Generalization

- Hypotheses must ***generalize*** to correctly classify instances not in the training data.
- Simply memorizing training examples yields a [consistent] hypothesis ***that does not generalize***.

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when
PREJUDICE meets DATA!

Learning a "Frobnitz"

Frobnitz



Not a Frobnitz

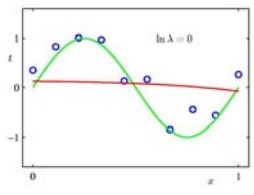


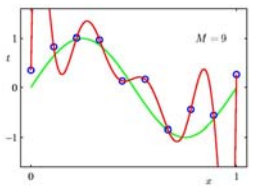
Bias

- The nice word for prejudice is "bias".
  - Different from "Bias" in statistics
- What kind of hypotheses will you ***consider***?
  - What is allowable ***range*** of approximation functions?
  - Eg conjunctions
    linear functions
- What kind of hypotheses do you ***prefer***?
  - Eg Simple hypotheses (Occam's Razor)
    few parameters,
    small parameters,

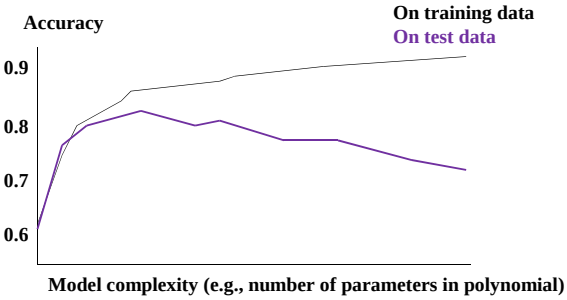
16

Fitting a Polynomial





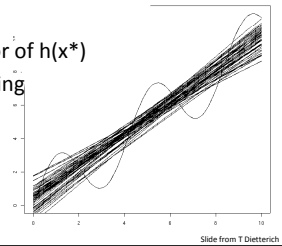
Overfitting



18

Bia / Variance Tradeoff

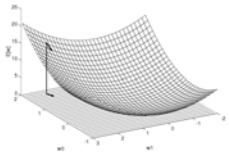
- Variance: **E[(h(x*) – h(x*))²]**
  How much h(x*) varies between training sets
  Reducing variance risks underfitting
- Bias: **[h(x*) – f(x*)]**
  Describes the ***average*** error of h(x*)
  Reducing bias risks overfitting



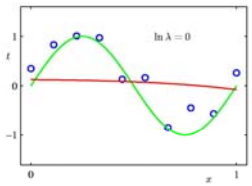
Slide from T Dietterich

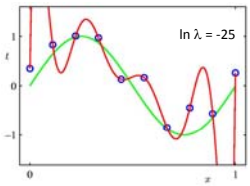
Learning as Optimization

- Loss Function
  – Loss(h,data) = error(h, data) + complexity(h)
  – Error + regularization
  – Minimize ***loss*** over training data
- Opt Methods
  – Closed form
  – Greedy search
  – Gradient ascent



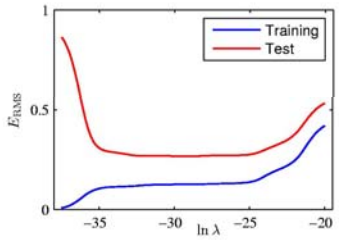
Effect of Regularization





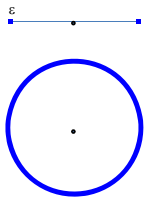
$$\text{Loss}(h_w) = \sum_{j=1}^{n}(y_j - (w_1 x_j + w_0))^2 \quad + \lambda \sum_{i=1}^{k} |w_i|$$

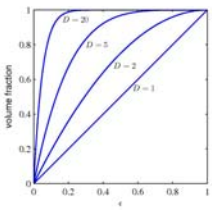
Regularization: E_{RMS} *vs.* $\ln \lambda$



Curse of Dimensionality

- Intuitions fail
- Hard to distinguish hypotheses

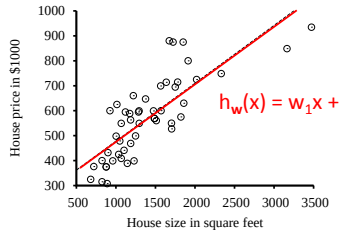




A Great Learning Algorithm

- Logistic Regression

Univariate Linear Regression



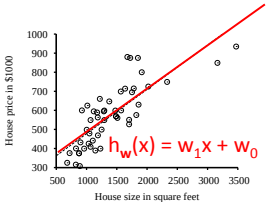
$h_w(x) = w_1 x + w_0$

$$\text{Loss}(h_w) = \sum_{j=1}^{n} L_2(y_j, h_w(x_j)) = \sum_{j=1}^{n} (y_j - h_w(x_j))^2 =$$

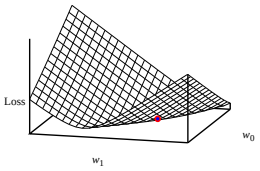
$$\sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2$$

25

Understanding Weight Space



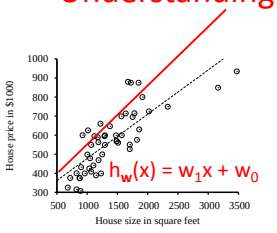
$h_w(x) = w_1 x + w_0$



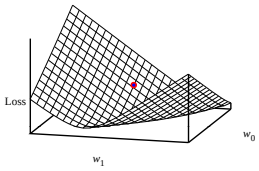
$$\text{Loss}(h_w) = \sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2$$

26

Understanding Weight Space



$h_w(x) = w_1 x + w_0$

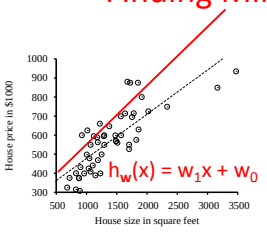


$$\text{Loss}(h_w) = \sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2$$

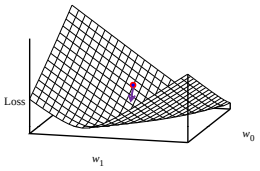
27

Finding Minimum Loss

$\text{Argmin}_w \text{Loss}(h_w)$



$h_w(x) = w_1 x + w_0$



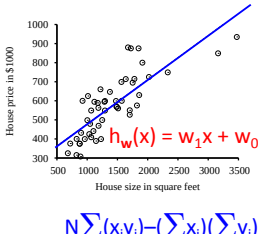
$$\text{Loss}(h_w) = \sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2$$

$$\frac{\partial}{\partial w_0} \text{Loss}(h_w) = 0 \qquad \frac{\partial}{\partial w_1} \text{Loss}(h_w) = 0$$

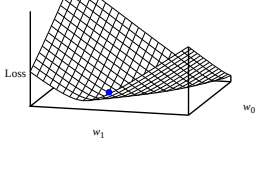
28

Unique Solution!

$\text{Argmin}_w \text{Loss}(h_w)$



$h_w(x) = w_1 x + w_0$

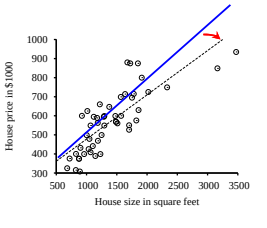


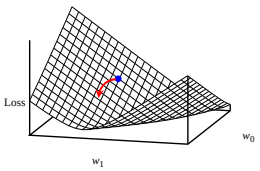
$$w_1 = \frac{N\sum(x_j y_j) - (\sum x_j)(\sum y_j)}{N\sum(x_j^2) - (\sum x_j)^2}$$

$$w_0 = (\sum(y_j) - w_1(\sum x_j)/N$$

Could also Solve Iteratively

$\text{Argmin}_w \text{Loss}(h_w)$





w = any point in weight space
Loop until convergence
For each w_i in w do

$$w_i := w_i - \alpha \frac{\partial}{\partial w_i} \text{Loss}(\mathbf{w})$$

Multivariate Linear Regression

$h_w(\mathbf{x}_j) = w_0 + \sum w_i x_{j,i} = \sum w_i x_{j,i} = w^T x_j$

$\text{Argmin}_w \text{Loss}(h_w)$

Unique Solution = $(x^T x)^{-1} x^T y$

Problem….

31

Overfitting

Regularize!!

Penalize high weights

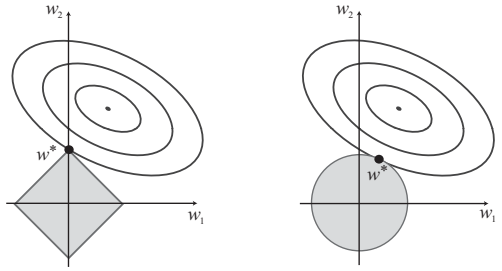
$$\text{Loss}(h_w) = \sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2 + \lambda \sum_{i=1}^{k} w_i^2$$

Alternatively….

$$\text{Loss}(h_w) = \sum_{j=1}^{n} (y_j - (w_1 x_j + w_0))^2 + \lambda \sum_{i=1}^{k} |w_i|$$

32

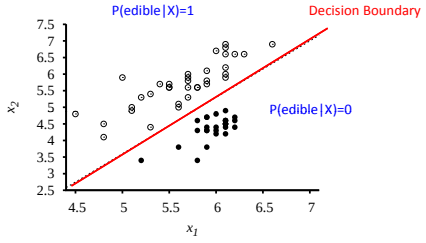
Regularization



L1 L2

33

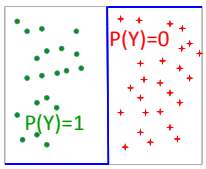
Back to Classification



34

Logistic Regression

- Learn P(Y|**X**) directly!
  - ☐ Assume a particular functional form
  - ☹ ***Not differentiable…***

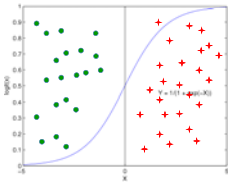


35

Logistic Regression

- Learn P(Y|**X**) directly!
  - ☐ Assume a particular functional form
  - ☐ Logistic Function
    - ☐ Aka Sigmoid

$$\frac{1}{1 + exp(-z)}$$

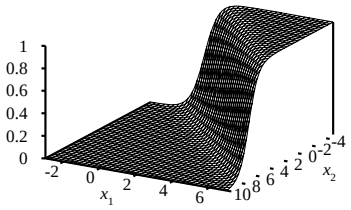


36

Logistic Function in n Dimensions

$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

Sigmoid applied to a linear function of the data:



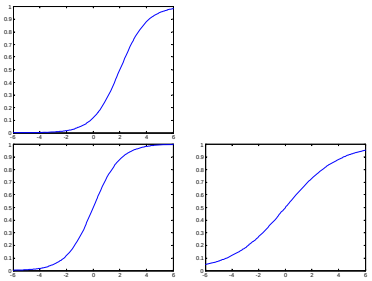
Features can be discrete or continuous!

37

Understanding Sigmoids

$$g(w_0+\sum_i w_i x_i) = \frac{1}{1+e^{w_0+\sum_i w_i x_i}}$$

w_0=-2, w_1=-1



w_0=0, w_1=-1

w_0=0, w_1= -0.5

Very convenient!

$$P(Y=1|X=<X_1,\ldots X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=0|X=<X_1,\ldots X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)} = exp(w_0+\sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln\frac{P(Y=0|X)}{P(Y=1|X)} = w_0+\sum_i w_i X_i$$

39

Loss Functions: Likelihood vs. Conditional Likelihood

Generative (Naïve Bayes) Loss function: **Data likelihood**

$$\ln P(\mathcal{D}\mid \mathbf{w}) = \sum_{j=1}^{N} \ln P(\mathbf{x}^j, y^j \mid \mathbf{w})$$
$$= \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) + \sum_{j=1}^{N} \ln P(\mathbf{x}^j \mid \mathbf{w})$$

Discriminative (Logistic Regr.) Loss funct: **Conditional Data Likelihood**

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_{\mathbf{X}}, \mathbf{w}) = \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$

Discriminative models *can't* compute P($\mathbf{x}^j$|**w**)!

Or, … "They don't *waste effort* learning P(**X**)"

Focus only on P(Y|**X**) - all that matters for classification

41

Expressing Conditional Log Likelihood

$$l(\mathbf{w}) = \sum_j \ln P(y^j|\mathbf{x}^j,\mathbf{w})$$

$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$ln(P(Y=0|\mathbf{X},\mathbf{w}) = -ln(1+exp(w_0+\Sigma_i w_i X_i)$$

$$P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$ln(P(Y=0|\mathbf{X},\mathbf{w}) = w_0+\Sigma_i w_i X_i - ln(1+exp(w_0+\Sigma_i w_i X_i)$$

$$l(\mathbf{w}) = \sum_j y^j \ln P(y^j=1|\mathbf{x}^j,\mathbf{w}) + (1-y^j)\ln P(y^j=0|\mathbf{x}^j,\mathbf{w})$$

1 when correct answer is 1

Probability of predicting 1

1 when correct answer is 0

Probability of predicting 0

42

Expressing Conditional Log Likelihood

$$l(\mathbf{w}) = \sum_j \ln P(y^j|\mathbf{x}^j,\mathbf{w})$$

$$ln(P(Y=0|\mathbf{X},\mathbf{w}) = -ln(1+exp(w_0+\Sigma_i w_i X_i)$$

$$ln(P(Y=0|\mathbf{X},\mathbf{w}) = w_0+\Sigma_i w_i X_i - ln(1+exp(w_0+\Sigma_i w_i X_i)$$

$$l(\mathbf{w}) = \sum_j y^j \ln P(y^j=1|\mathbf{x}^j,\mathbf{w}) + (1-y^j)\ln P(y^j=0|\mathbf{x}^j,\mathbf{w})$$

$$= \sum_j y^j(w_0+\sum_i^n w_i x_i^j) - \ln(1+exp(w_0+\sum_i^n w_i x_i^j))$$

Maximizing Conditional Log Likelihood

$$P(Y=0|X,W)=\frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$
$$P(Y=1|X,W)=\frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$l(\mathbf{w}) = \ln \prod_j P(y^j|\mathbf{x}^j,\mathbf{w})$$
$$= \sum_j y^j(w_0+\sum_i^n w_i x_i^j) - \ln(1+exp(w_0+\sum_i^n w_i x_i^j))$$

Bad news: no closed-form solution to maximize *l*(**w**)

Good news: *l*(**w**) is concave function of **w**!

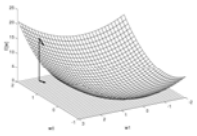
No local minima

Concave functions easy to optimize

Optimizing Concave Functions Gradient Ascent

Conditional likelihood for Logistic Regression is concave !

Find optimum with gradient ascent



Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0},\ldots,\frac{\partial l(\mathbf{w})}{\partial w_n}]'$

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

Learning rate, η>0

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

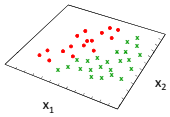
Gradient ascent is simplest of optimization approaches

e.g., Conjugate gradient ascent much better (see reading)

Earthquake or Nuclear Test?

$$P(Y=1|X=<X_1,\ldots X_n>)=\frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies



$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = w_0 + \sum_i w_i X_i$$

linear classification rule!

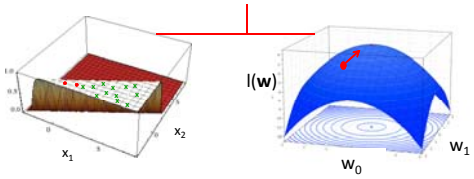
If $w_0 + \sum_i w_i X_i > 1$, then predict Y=0

Logistic w/ Initial Weights

w_0=20 w_1= -5 w_2 =10

Loss(H_w) = Error(H_w, data)

Minimize Error → Maximize l(**w**) = ln P(D_Y | D_x, H_w)



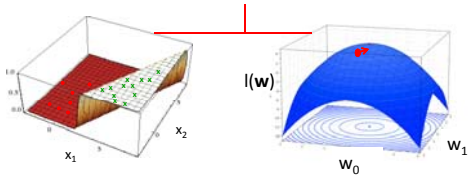
Update rule:

$$\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$$
$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

Gradient Ascent

w_0=40 w_1= -10 w_2 =5

Maximize l(**w**) = ln P(D_Y | D_x, H_w)



Update rule:

$$\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$$
$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

IE as Classification

\+ Citigroup has taken over EMI, the British …

\+ Citigroup's acquisition of EMI comes just ahead of …

\- Google's Adwords system has long included ways to connect to Youtube.

Preprocessed Data Files

Each line corresponds to a sentence. "John likes eating sausage."

tokens	after tokenization	John likes eating sausage .

Preprocessed Data Files

Each line corresponds to a sentence. "John likes eating sausage."

tokens	after tokenization	John likes eating sausage .
pos	Part-of-Speech tags	John/NNP likes/VBZ eating/VBG sausage/NN ./.

Grade School: "9 parts of speech in English"

- Noun
- Verb
- Article
- Adjective
- Preposition
- Pronoun
- Adverb
- Conjunction
- Interjection

But: plurals, possessive, case, tense, aspect, ….

Preprocessed Data Files

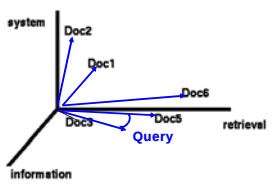
Each line corresponds to a sentence. "John likes eating sausage."

tokens	after tokenization	John likes eating sausage .
pos	Part-of-Speech tags	John/NNP likes/VBZ eating/VBG sausage/NN ./.
ner	Named Entities	Pers John likes eating sausage.

Text as Vectors

- Each document, *j*, can be viewed as a vector of *frequency* values,
  - one component for each word (or phrase)
- So we have a vector space
  - Words (or phrases) are axes
  - documents live in this space
  - even with stemming, may have 20,000+ dimensions

Vector Space Representation



Documents that are close to query (measured using vector-space metric) => returned first.

slide from Raghavan, Schütze, Larson

Lemmatization

- Reduce inflectional/variant forms to base form
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*

the boy's cars are different colors

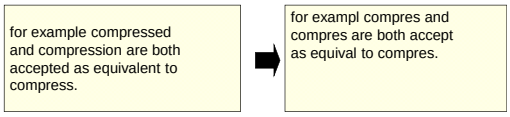
→

the boy car be different color

slide from Raghavan, Schütze, Larson

Stemming

- Reduce terms to their "roots" before indexing
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.



slide from Raghavan, Schütze, Larson

Porter's algorithm

- Common algorithm for stemming English
- Conventions + 5 phases of reductions
  - phases applied sequentially
  - each phase consists of a set of commands
  - sample convention: *Of the rules in a compound command, select the one that applies to the longest suffix.*
- Porter's stemmer available:
  http//www.sims.berkeley.edu/~hearst/irbook/porter.html

slide from Raghavan, Schütze, Larson

Typical rules in Porter

- *sses* → *ss*
- *ies* → *i*
- *ational* → *ate*
- *tional* → *tion*

slide from Raghavan, Schütze, Larson

Challenges

- Sandy
- Sanded
- Sander

➔ Sand ???

slide from Raghavan, Schütze, Larson