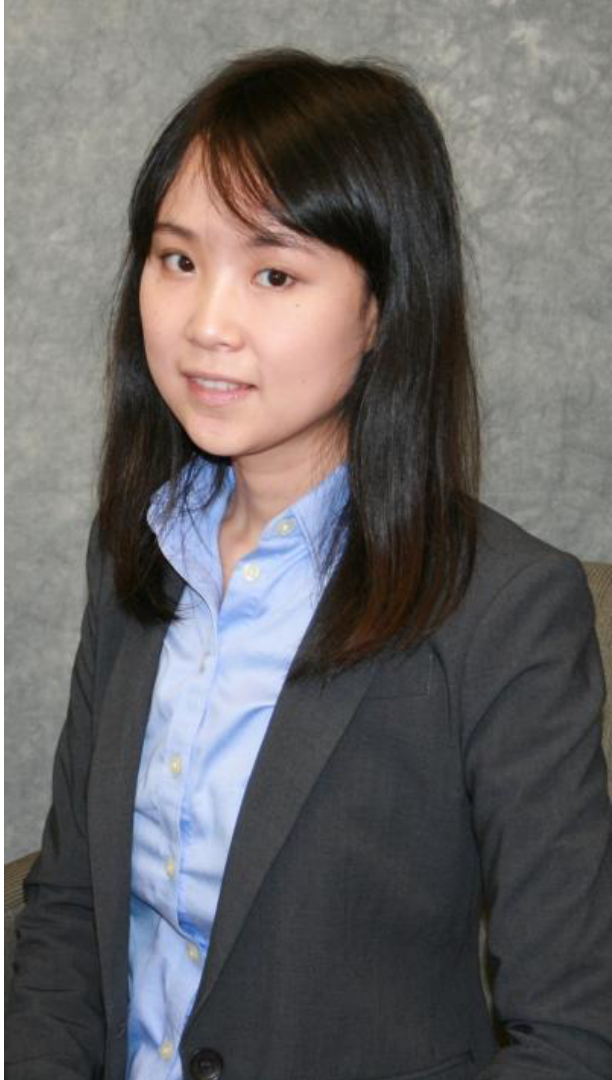
Computational Data Analysis

Machine Learning



Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Feature (variable) selection

Motivation: Apartment hunting

- Predict what is a reasonable rent? -- the response variable
- Which are the most important variables/features in predicting rent?

Rent ($)	Living area (ft^2)	Location	# bath	# bedroom
600	230	midtown	1	1
1000	506	buckhead	2	2
1100	433	midtown	1	2
500	109	downtown	1	1
	…			
?	150	midtown	2	1
?	270	downtown	1	1.5

Some features may contain redundant information

http://www.lockwoodresidential.com/2018/10/12/ten-apartment-hunting-tips-that-will-help-you-find-your-perfect-home/

Motivation: Document classification

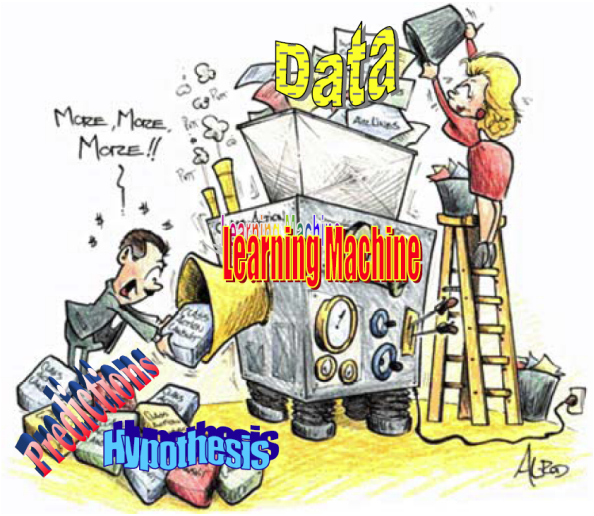
- Select the most important features before build classifier
- Suppose a rare term, say “arachnocentric”, has no information about a class, say “China”, but all instances of “arachnocentric” happen to occur in China documents in our training set. Then the learning method might produce a classifier that misassigns test documents containing arachnocentric to China.

Introduction to information retrieval, Manning, Raghavan, Schutze, 2008.

https://medium.com/@nishantv/ectd-document-classification-with-python-and-machine-learning-7124a1419f8

Feature selection

- Also known as variable selection
- Select a subset of relevant features (variables) for model
- Motivation
    - Simplify model
    - More data efficient (training a smaller model)
    - Better interpretation: which variables are more important for prediction, classification, or target machine learning task?
    - Often increases classification accuracy by eliminating noise features
    - Enhance generalization by reducing over fitting (bias-variance tradeoff)
- Applications: NLP, image analysis, genetic data



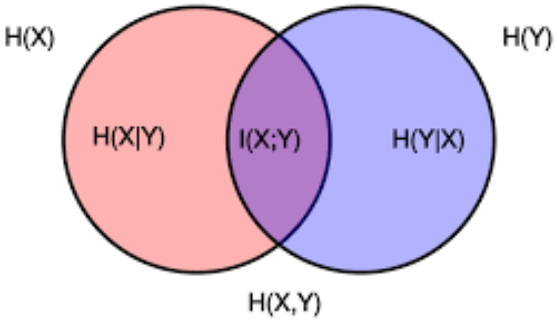
"Garbage in, garbage out."

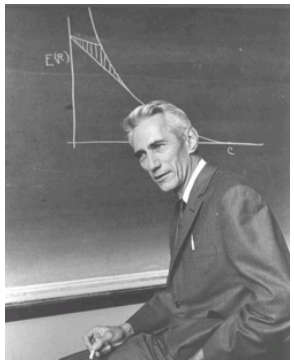
Major approaches for feature / variable selection

- Combination: evaluation metrics and search technique
- Evaluation metrics:
  - Heuristics: e.g., eliminate variables with small variance
  - Information theoretic metric
  - Bias-variance tradeoffs: AIC, BIC, Rp statistics
  - Error probability
- Search technique:
  - Subset selection: enumeration
  - l1-regularized algorithms: LASSO and extensions

Information theoretic metric

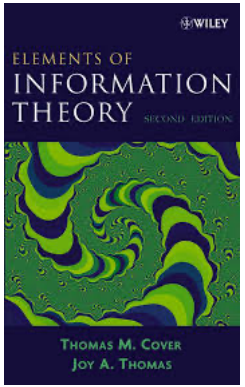
- Select the variables that contain the most important information for the response.
- How do we quantify information?
- Mutual information: can capture any kind of dependency between variables





Claude Shannon
(1916 – 2001)

The Bit Player (2018)

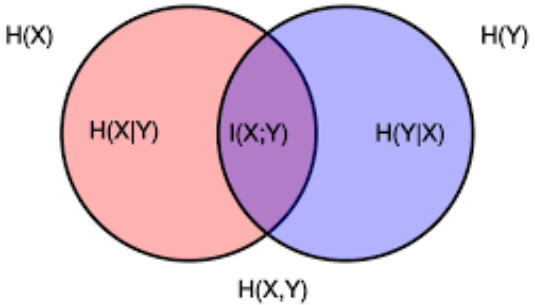


Information theoretical measures

- We are uncertain about the label Y before seeing any input
  - Quantify using **entropy** $H(Y)$

- Given a particular feature X_i, the uncertainty of Y reduces
  - Quantify using **conditional entropy** $H(Y|X_i)$

- Reduction in uncertainty is the informativeness of feature X_i
  - Quantify using **mutual information**
    $I(X_i, Y) = H(Y) - H(Y|X_i)$

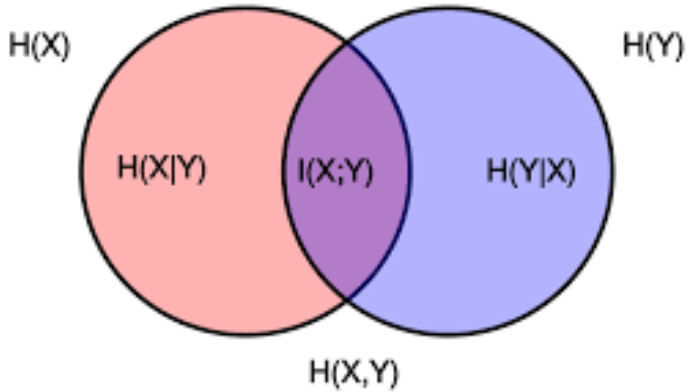


Entropy: Quantify uncertainty

- Entropy $H(Y)$ of a (discrete) random variable Y

$$H(Y) = -\sum_{k=1}^{K} P(y = k) \log_2 P(y = k)$$

The number of bits needed to describe Y



Example: Entropy of coin flip

S is a sample of coin flips

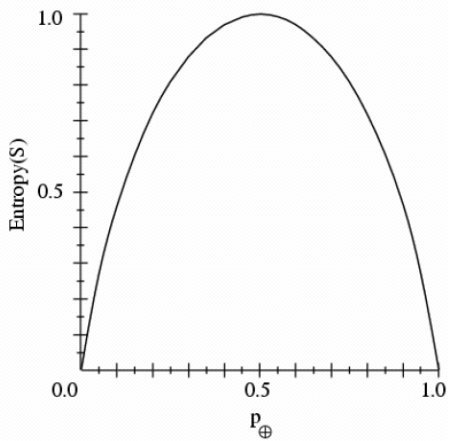
p is probability of getting a head in S

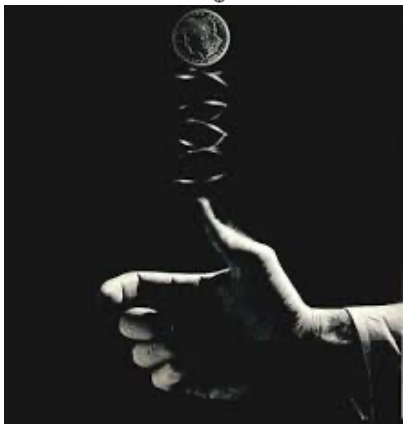
Entropy measure the uncertainty of S

$$H(S) = -p \log p - (1-p) \log(1-p)$$

Fair coin $p = \frac{1}{2}$

Biased coin $p \neq \frac{1}{2}$

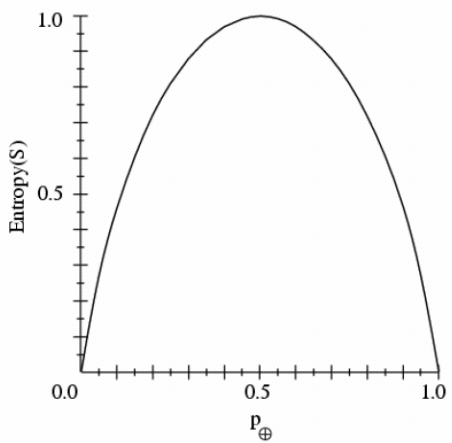




https://sites.google.com/site/ciese07030/Home/coin-flipping-probabilities

Examples

Fair coin vs. biased coin?



$$H(S) = -p \log p - (1-p) \log(1-p)$$

head	**0**
tail	**6**

P(h) = 0/6 = 0 P(t) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0 (bit)

head	**1**
tail	**5**

P(h) = 1/6 P(t) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65 (bit)

head	**3**
tail	**3**

P(h) = 1/2 P(t) = 1/2

Entropy = – (1/2) $\log_2$ (1/2) – (1/2) $\log_2$ (1/2) = 1 (bit)

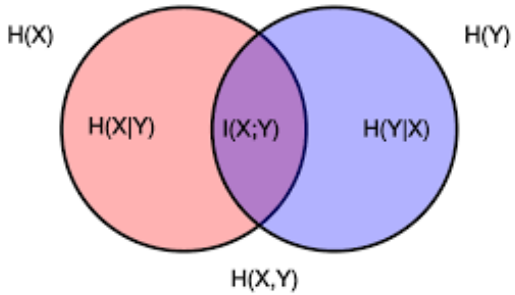
Conditional entropy

- Conditional entropy $H(Y|X)$ of a random variable Y given X_i

$$H(Y|X_i) = -\int H(Y|X_i = x_i)p(x_i)dx_i$$

$$= -\int \left(\sum_{k=1}^{K} P(y = k|X_i = x_i) \log_2 P(y = k, X_i = x_i) \right) p(x_i)dx_i$$

- Quantify the remaining uncertainty in Y after seeing feature X_i



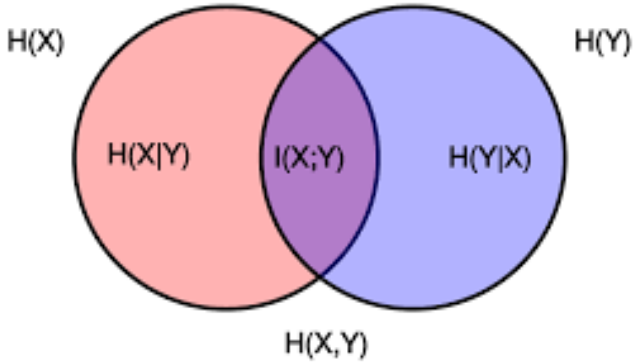
Mutual information

- Mutual information:

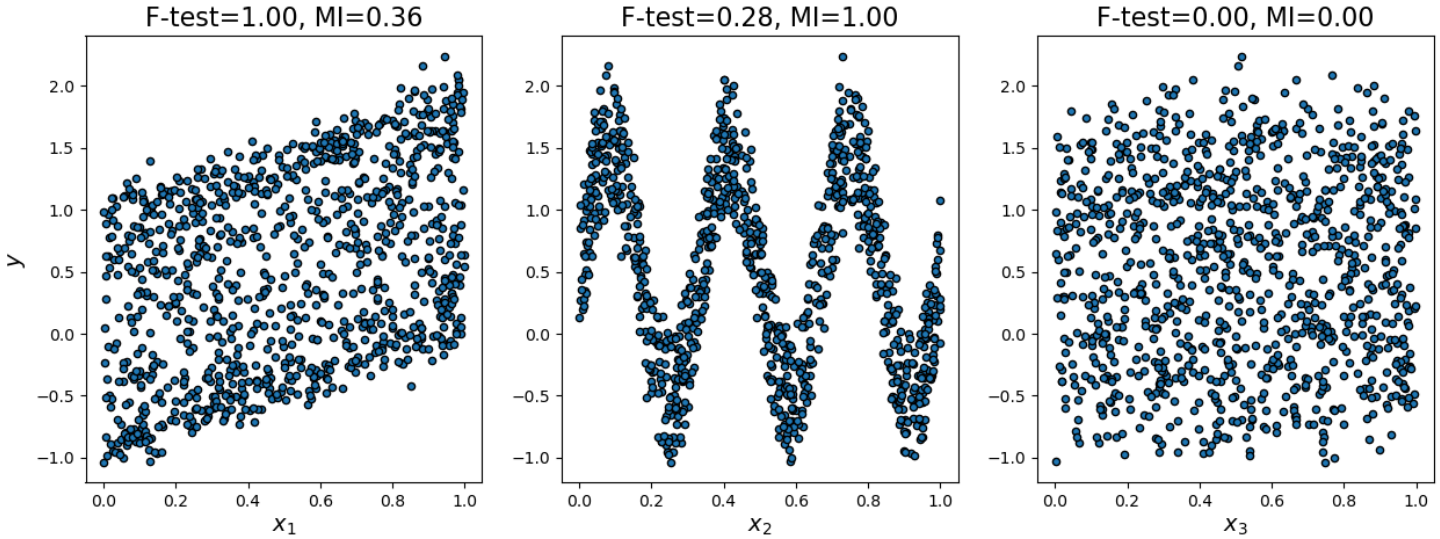
Quantify the reduction in uncertainty in Y after seeing feature X_i

$$I(X_i, Y) = H(Y) - H(Y|X_i)$$

- The more the reduction in entropy, the more informative a feature is.



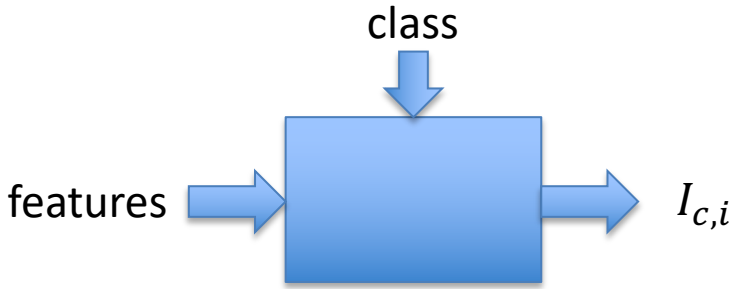
Demo: mutual information can capture non-linear dependence



https://scikit-learn.org/stable/auto_examples/feature_selection/plot_f_test_vs_mi.html#sphx-glr-auto-examples-feature-selection-plot-f-test-vs-mi-py

A feature selection algorithm

- Given a dataset $S = \{(x^1, y^1), \ldots, (x^m, y^m)\}$, $x \in R^n$, $y = \{1, \ldots, K\}$
- For each feature x_i
  - Estimate density $p(x_i)$
- For each class $y = c$
  - Estimate density $p(y = c)$
- For each class $y = c$, and each feature x_i
  - Estimate joint density $p(x_i, y = c)$
  - Score feature x_i using MI



$$I_{c,i} = \int \sum_{c=1}^{K} p(x_i, y = c) \log_2 \frac{p(x_i, y = c)}{p(x_i)p(y = c)} \, dx_i$$

Choose those feature x_i for class c with high score $I_{c,i}$

Example: Feature selection for document classification

- Mutual information (MI) measures how much information a term contains about the document class.
- Intuition:
  - If a term's distribution is the same in the class as it is in the collection as a whole, then it contains little information.
  - MI reaches its maximum value if the term is a perfect indicator for class membership, that is, if the term is present in a document if and only if the document is in the class.

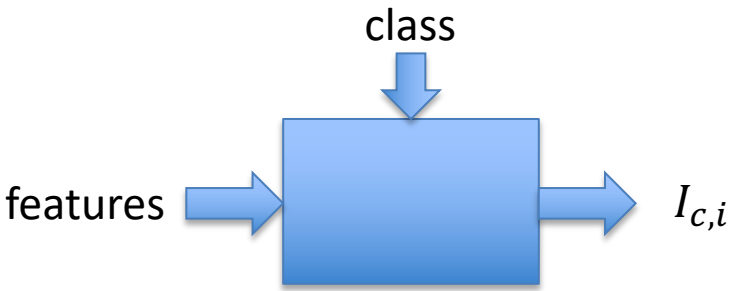
- Document class
- Terms

Introduction to information retrieval, Manning, Raghavan, Schutze, 2008.

Example: Feature selection for document classification

- Reuter's dataset
- Document class C
- Features: terms U

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)},$$



https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection

https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

Example: Feature selection for document classification

- From MLE of probabilities

class

term

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{export} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{export} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

$$\begin{aligned} I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)} \\ &\quad + \frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)} \\ &\quad + \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\ &\quad + \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\ &\approx 0.0001105 \end{aligned}$$

https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection

https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

Result: Reuter's dataset

UK		*China*		*poultry*	
london	0.1925	china	0.0997	poultry	0.0013
uk	0.0755	chinese	0.0523	meat	0.0008
british	0.0596	beijing	0.0444	chicken	0.0006
stg	0.0555	yuan	0.0344	agriculture	0.0005
britain	0.0469	shanghai	0.0292	avian	0.0004
plc	0.0357	hong	0.0198	broiler	0.0003
england	0.0238	kong	0.0195	veterinary	0.0003
pence	0.0212	xinhua	0.0155	birds	0.0003
pounds	0.0149	province	0.0117	inspection	0.0003
english	0.0126	taiwan	0.0108	pathogenic	0.0003

coffee		*elections*		*sports*	
coffee	0.0111	election	0.0519	soccer	0.0681
bags	0.0042	elections	0.0342	cup	0.0515
growers	0.0025	polls	0.0339	match	0.0441
kg	0.0019	voters	0.0315	matches	0.0408
colombia	0.0018	party	0.0303	played	0.0388
brazil	0.0016	vote	0.0299	league	0.0386
export	0.0014	poll	0.0225	beat	0.0301
exporters	0.0013	candidate	0.0202	game	0.0299
exports	0.0013	campaign	0.0202	games	0.0284
crop	0.0012	democratic	0.0198	team	0.0264

Figure 13.7: Features with high mutual information scores for six Reuters-RCV1 classes.

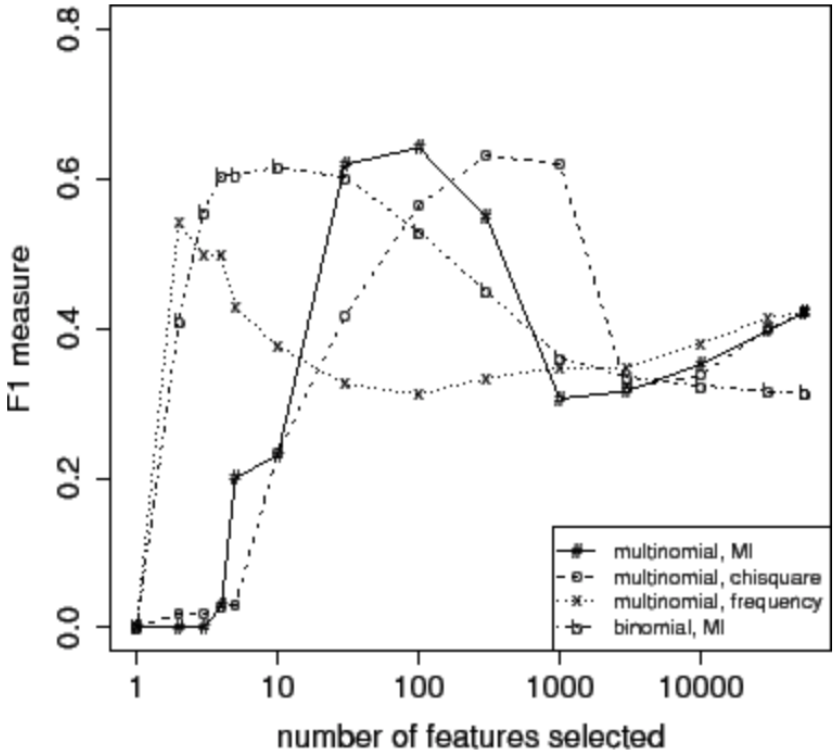
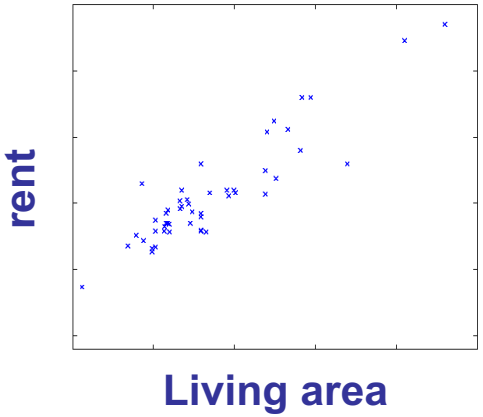


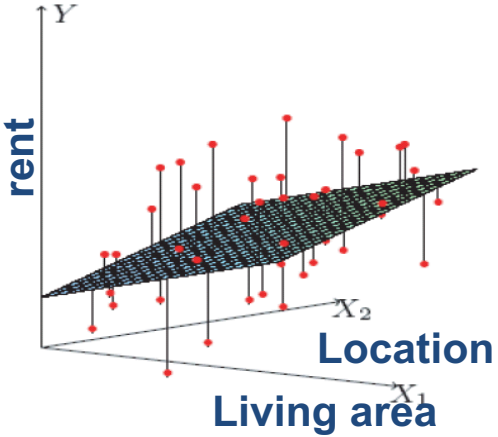
Figure 13.8: Effect of feature set size on accuracy for multinomial and Bernoulli models.

Major approaches for feature / variable selection

- Combination: evaluation metrics and search technique
- Evaluation metrics:
  - Heuristics: e.g., eliminate variables with small variance
  - Information theoretic metric
  - Bias-variance tradeoffs: AIC, BIC, Rp statistics
  - Error probability
- Search technique:
  - Subset selection: enumeration
  - l1-regularized algorithms: LASSO and extensions

Apartment hunting





Features:

- Living area, distance to campus, # bedroom …
- Denote as $x = (x_1, x_2, \ldots, x_n)^\top$

Target:

- Rent, denoted as y

Training set:

- $X = [x^1 \ldots x^m] \in R^{n \times m}$
- $y = (y^1, y^2, \ldots, y^m)^\top \in R^m$

Linear regression model

- Assume y is a linear function of x (features) plus noise ϵ

$$y = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n + \epsilon$$

where ϵ is an error term: unmodeled effects or random noise

- Let $\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T$, and augment data by one dimension

$$x \leftarrow (1, x^T)^T$$

- Then $y = \theta^T x + \epsilon$
- Variable section:

select a subset of variables that are most “important”

$\leftrightarrow$ find θ and their locations of ”zero-entries”

Least-square (LS) method

Given m data points, find θ that minimizes the mean square error

$$\hat{\theta} = argmin_{\theta}\, L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2$$

Our usual trick: set gradient to 0 and find parameter

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)x^i = 0$$

$$\Leftrightarrow -\frac{2}{m}\sum_{i=1}^{m} y^i x^i + \frac{2}{m}\sum_{i=1}^{m} x^i x^{i\top}\theta = 0$$

$$\Leftrightarrow \hat{\theta} = (XX^\top)^{-1}Xy$$

Ridge regression

- What if we cannot invert XX^T (e.g., when variables are highly correlated)
- Given m data points, find θ that minimizes the regularized mean square error

$$\theta^r = argmin_\theta \, L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_2^2$$

- gradient becomes

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m}Xy + \frac{2}{m}XX^\top\theta + 2\lambda\theta = 0$$

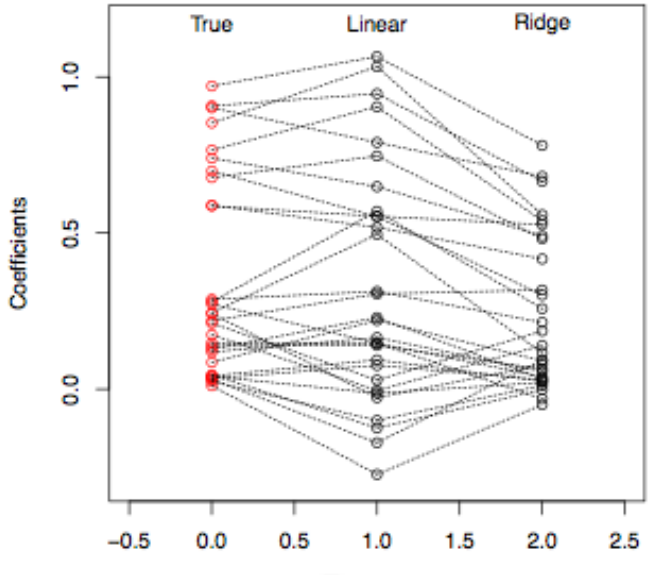
$$\Rightarrow \theta^r = \left(\frac{1}{m}XX^T + \lambda I\right)^{-1}\left(\frac{1}{m}Xy\right)$$

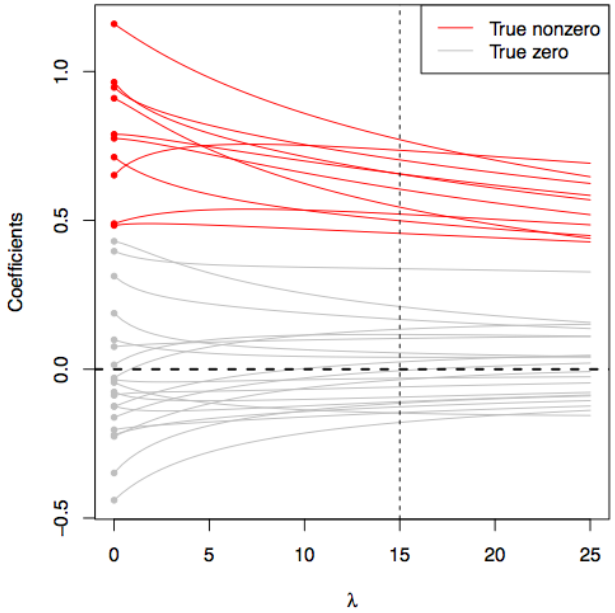
Regularizer Parameter

- If we choose a different λ, the solution will be different.

Ridge regression shrinks coefficients

$m = 50, n = 20, \lambda = 25$





But it does not select variable…

Variable selection for regression

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots + \theta_n x_n + \epsilon$$

- select a subset of variables that are most “important”

  $\leftrightarrow$ find θ and their locations of ”zero-entries”

- Direct enumeration is expensive: when there are n variables, we have to consider $O(2^n)$ possibilities
- Related to solving

$$\frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_0$$

- LASSO address this issue by *convex relaxation*

LASSO

(Least absolute shrinkage and selection operator)

- Variable selection in linear regression

$$\theta^r = argmin_\theta\, L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_1$$

- Regularizer λ controls model complexity: larger λ restricts less parameters will be selected
- This is a convex optimization problem and can be solved efficiently
- L1 penalty can be used for other type of algorithms to encourage **sparsity** in solution

(Tibshrani, 1996)

Demo: Diabetes dataset

Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

https://scikit-learn.org/stable/auto_examples/feature_selection/plot_select_from_model_diabetes.html#select-from-the-model-features-with-the-higest-score

https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

LASSO versus ridge

- We are interested in the level of prostate-specific antigen (PSA) elevated in men who has prostate cancer
- Measure PSA on 97 patients, 8 clinical variables
- Ridge does not perform variable selection; LASSO does
- From LASSO results,

if we want 3 leading factors,

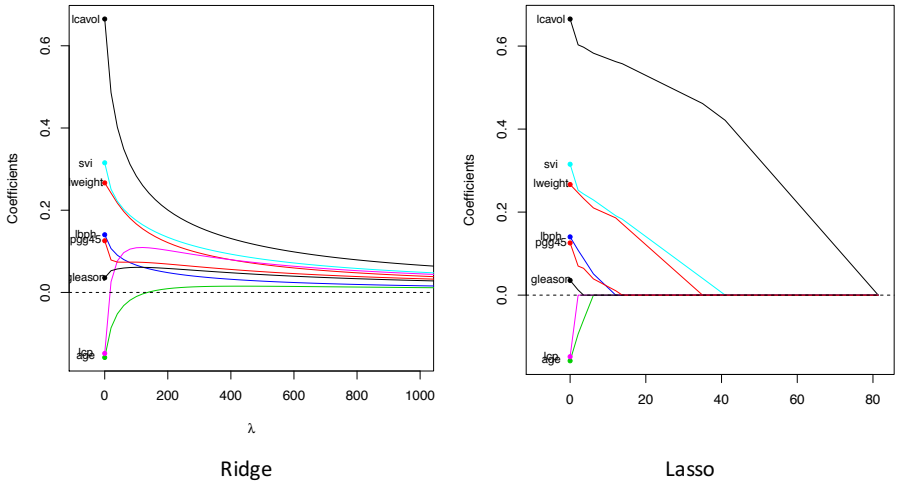
we report “cancer volume”,

“seminal sesicle invation”,

“prostate weight”

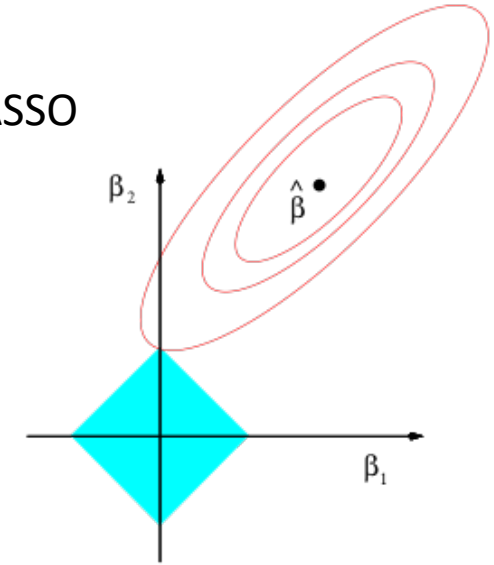
- Select λ:

using cross-validation

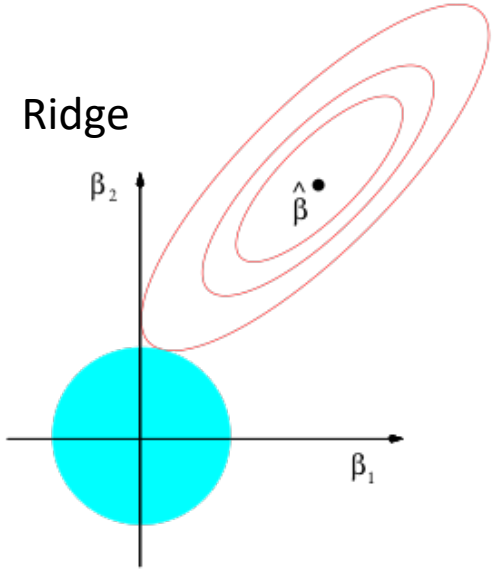


Why LASSO can select variable but not ridge?

LASSO



Ridge



$$\min \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_1$$

$$\min \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_2^2$$

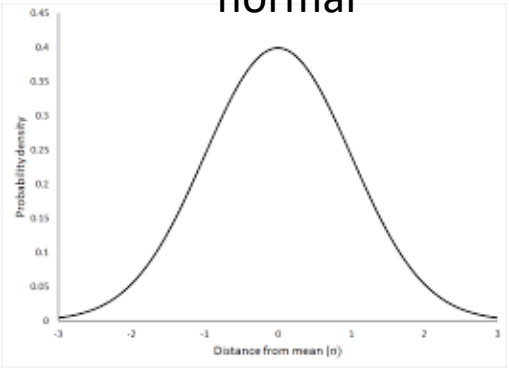
Fig. credit: Hastie, Tibshrani, Wainwright.

Bayesian interpretation

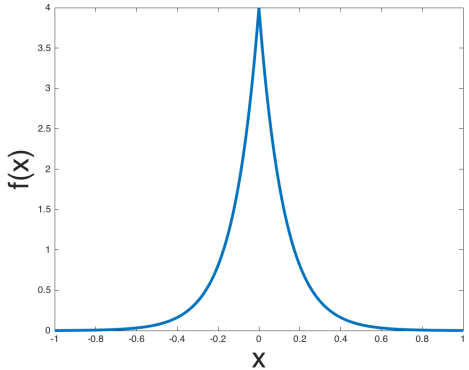
If we assume entries of θ is a random variable and follows

- Zero-mean normal distribution: solving the maximum-a-posterior corresponds to Ridge regression
- Laplacian distribution (double exponential): solving the maximum-a-posterior corresponds to LASSO

normal



Laplacian



Laplace distribution has fatter tails than the normal distribution.

Elastic net

- Ridge regression: helps when variables are correlated but cannot perform variable selection
- LASSO: helps with variable selection, not stable when variables are correlated

- Elastic net: combines two approaches $\alpha \in [0,1]$

$$\min \frac{1}{m} \sum_{i=1}^{m} \left(y^i - \theta^\top x^i\right)^2 + \lambda(\alpha \|\theta\|_2^2 + (1-\alpha)\|\theta\|_1)$$

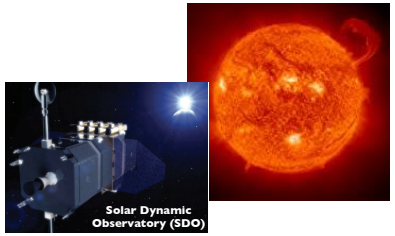
(Zhou, Hastie 2010)

Mutual information driven measurement design

- streaming data generated at high speed
- requires real-time sensing and recovery

Monitoring wind farms
2000 sensors and each
330MBytes/sec

In-situ real-time seismic
imaging, hundreds of sensors,
sampling ~200 Hz



Each frame ~ 70k pixels
130Mbps
11 terabytes/day

Algorithm 1 Info-Greedy Sensing

Require: distributions of signal x and noise w, error tolerance ε or maximum number of iterations M

1: $i \leftarrow 1$

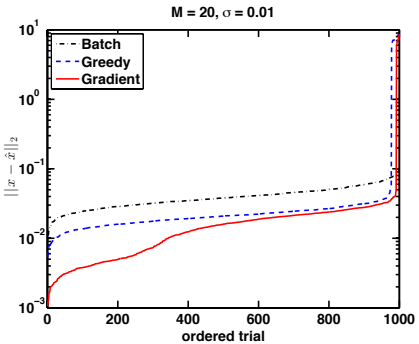
2: **repeat**

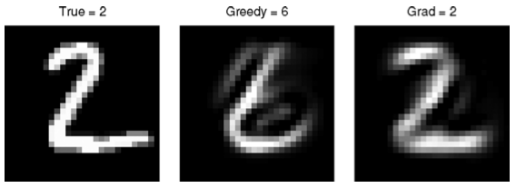
3: $a_i \leftarrow \text{argmax}_{a_i} \mathbb{I}\left[x; a_i^\intercal x + w_i \middle| y_j, a_j, j < i\right] / \beta_i$

4: $y_i = a_i^\intercal x + w_i$ {measurement}

5: $i \leftarrow i + 1$

6: **until** $\mathbb{I}\left[x; a_i^\intercal x + w_i \middle| y_j, a_j, j \le i\right] \le \delta(\varepsilon)$ or $i > M$.





Info-greedy sequential adaptive compressed sensing, IEEE Journal of Selected Topics in Signal processing, Braun, Pokutta, Xie, Vol. 9. No. 4, June 2015.