Feature Selection

CS 760@UW-Madison

Goals for the lecture

you should understand the following concepts

- filtering-based feature selection
- information gain filtering
- Markov blanket filtering
- frequency pruning
- wrapper-based feature selection
- forward selection
- backward elimination
- L_1 and L_2 penalties
- lasso and ridge regression

Motivation for feature selection

1. We want models that we can interpret. We're specifically interested in which features are relevant for some task.

2. We're interested in getting models with better predictive accuracy, and feature selection may help.

3. We are concerned with efficiency. We want models that can be learned in a reasonable amount of time, and/or are compact and efficient to use.

Motivation for feature selection

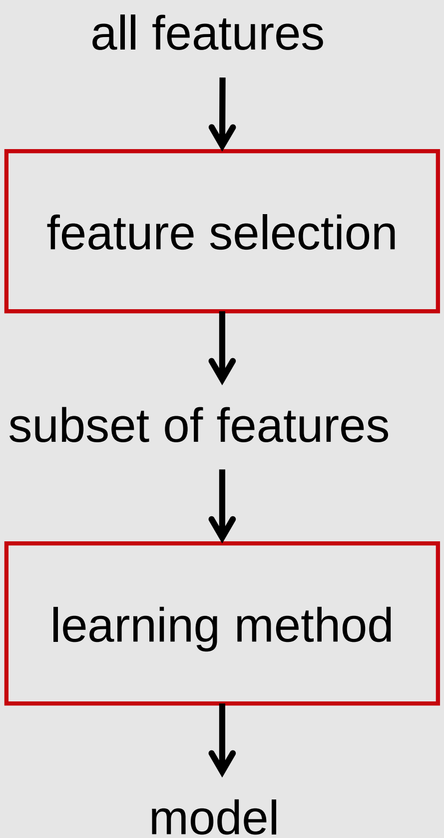
- some learning methods are sensitive to irrelevant or redundant features
  - k-NN
  - naïve Bayes
  - etc.

- other learning methods are ostensibly insensitive to irrelevant features (e.g. Weighted Majority) and/or redundant features (e.g. decision tree learners)

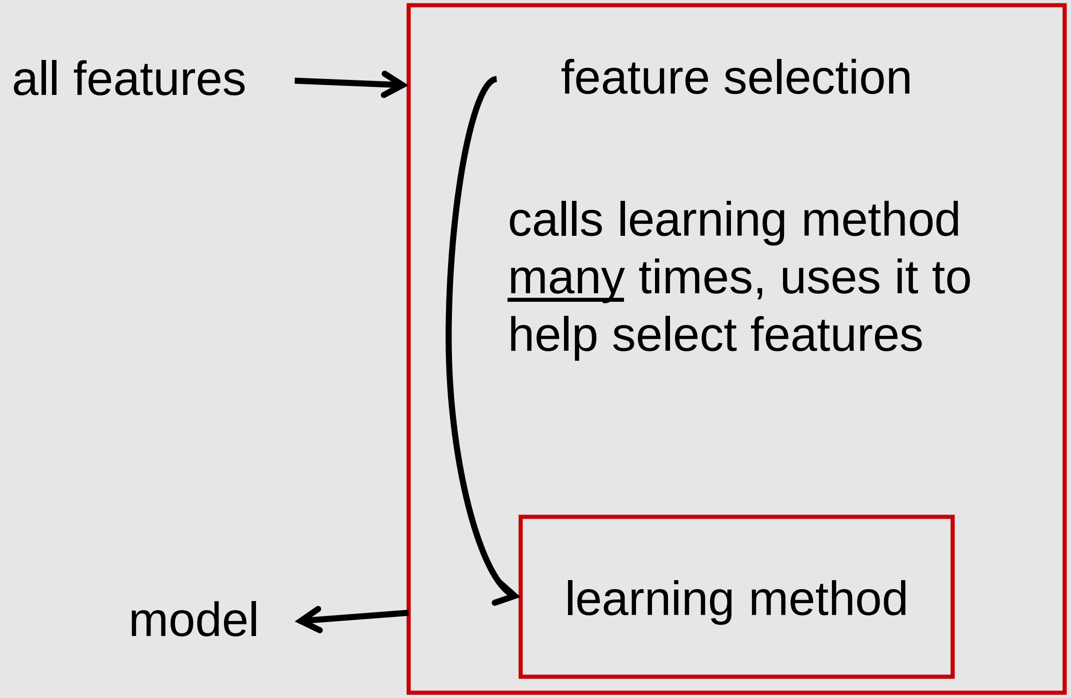
- empirically, feature selection is sometimes useful even with the latter class of methods [Kohavi & John, *Artificial Intelligence* 1997]

Feature selection approaches

filtering-based
feature selection



wrapper-based
feature selection



Information gain filtering

- select only those features that have significant information gain (mutual information with the class variable)

$$\text{InfoGain}(Y, X_i) = H(Y) - H(Y \mid X_i)$$

entropy of class variable (in training set)

entropy of class variable given feature X_i

- unlikely to select features that are highly predictive only when combined with other features
- may select many redundant features

Markov blanket filtering

[Koller & Sahami, *ICML* 1996]

- a Markov blanket $\boldsymbol{M}_i$ for a variable X_i is a set of variables such that all other variables are conditionally independent of X_i given $\boldsymbol{M}_i$
- we can try to find and remove features that minimize the criterion:

$\boldsymbol{x}$ projected onto features in $\boldsymbol{M}_i$

$$\mathrm{D}(X_i, \boldsymbol{M}_i) = \sum_{\boldsymbol{x}_{\boldsymbol{M}_i}, x_i} \left[P(\boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i}, X_i = x_i) \times D_{KL}\left(P(Y \mid \boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i}, X_i = x_i) \;\|\; P(Y \mid \boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i}) \right) \right]$$

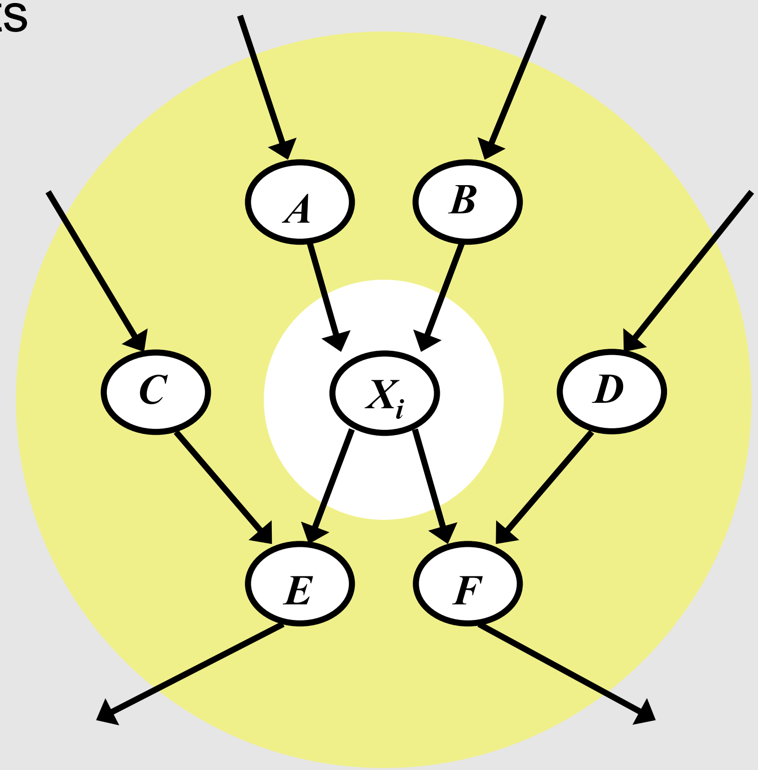
Kullback-Leibler divergence
(distance between 2 distributions)

- if Y is conditionally independent of feature X_i given a subset of other features, we should be able to omit X_i

Bayes net view of a Markov blanket

$$P(X_i \mid \boldsymbol{M}_i, Z) = P(X_i \mid \boldsymbol{M}_i)$$

- the Markov blanket $\boldsymbol{M}_i$ for variable X_i consists of its parents, its children, and its children's parents



- but we know that finding the best Bayes net structure is NP-hard; can we find approximate Markov blankets efficiently?

Heuristic to find an approximate Markov blanket

$$\mathrm{D}(X_i, \boldsymbol{M}_i) = \sum_{\boldsymbol{x}_{\boldsymbol{M}_i}, x_i} \begin{bmatrix} P(\boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i}, X_i = x_i) \times \\ D_{KL}\left(P(Y \mid \boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i}, X_i = x_i) \parallel P(Y \mid \boldsymbol{M}_i = \boldsymbol{x}_{\boldsymbol{M}_i})\right) \end{bmatrix}$$

// initialize feature set to include all features

$\boldsymbol{F} = \boldsymbol{X}$

iterate

for each feature X_i in $\boldsymbol{F}$

let $\boldsymbol{M}_i$ be set of k features most correlated with X_i

compute $\Delta(X_i, \boldsymbol{M}_i)$

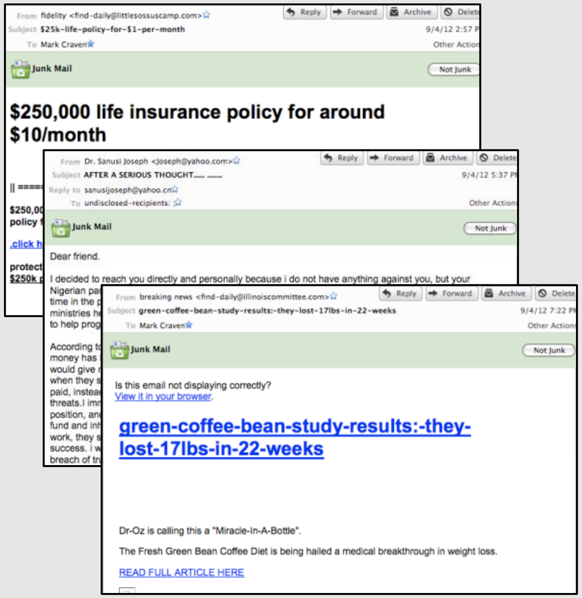
choose the X_r that minimizes $\Delta(X_r, \boldsymbol{M}_r)$

$\boldsymbol{F} = \boldsymbol{F} - \{ X_r \}$

return $\boldsymbol{F}$

Another filtering-based method: *frequency pruning*

- remove features whose value distributions are highly skewed
- common to remove very high-frequency and low-frequency words in text-classification tasks such as spam filtering



some words occur so <u>frequently</u> that they are not informative about a document's class

the
be
to
of
…

some words occur so <u>infrequently</u> that they are not useful for classification

accubation
cacodaemonomania
echopraxia
ichneutic
zoosemiotics
…

Example: feature selection for cancer classification

- classification task is to distinguish two types of leukemia: AML, ALL
- 7130 features represent expression levels of genes in tumor samples
- 72 instances (patients)
- three-stage filtering approach which includes information gain and Markov blanket [Xing et al., *ICML* 2001]

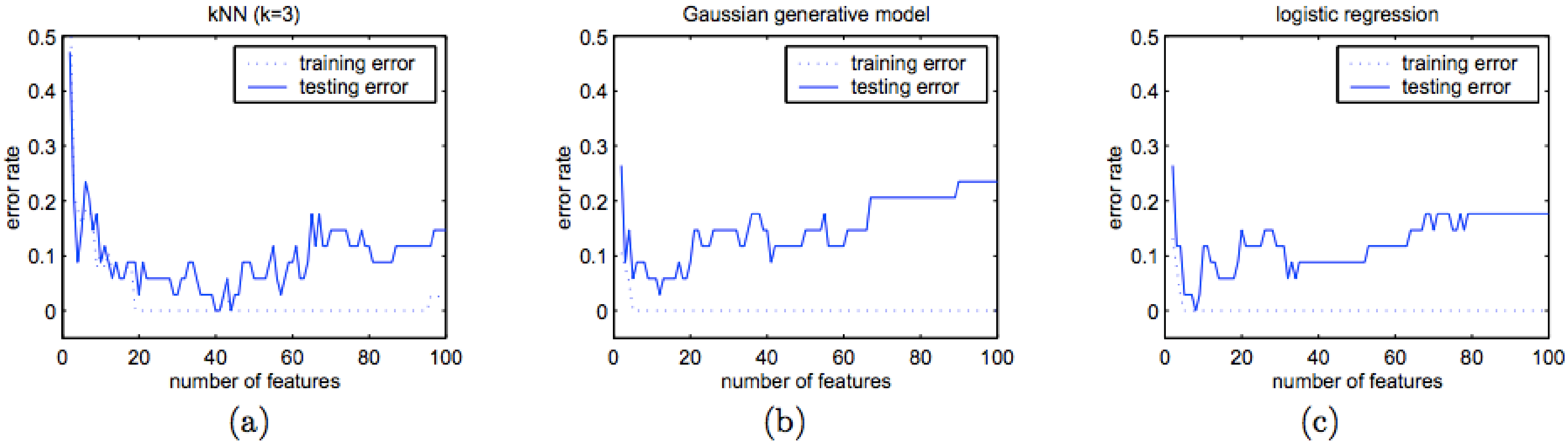


Figure from Xing et al., *ICML* 2001

Wrapper-based feature selection

- frame the feature-selection task as a search problem
- evaluate each feature set by using the learning method to score it (how accurate of a model can be learned with it?)

Feature selection as a search problem

state = set of features

start state = *empty* (forward selection)
or *full* (backward elimination)

operators

add/subtract a feature

scoring function

training or tuning-set or CV accuracy using learning method on a given state's feature set

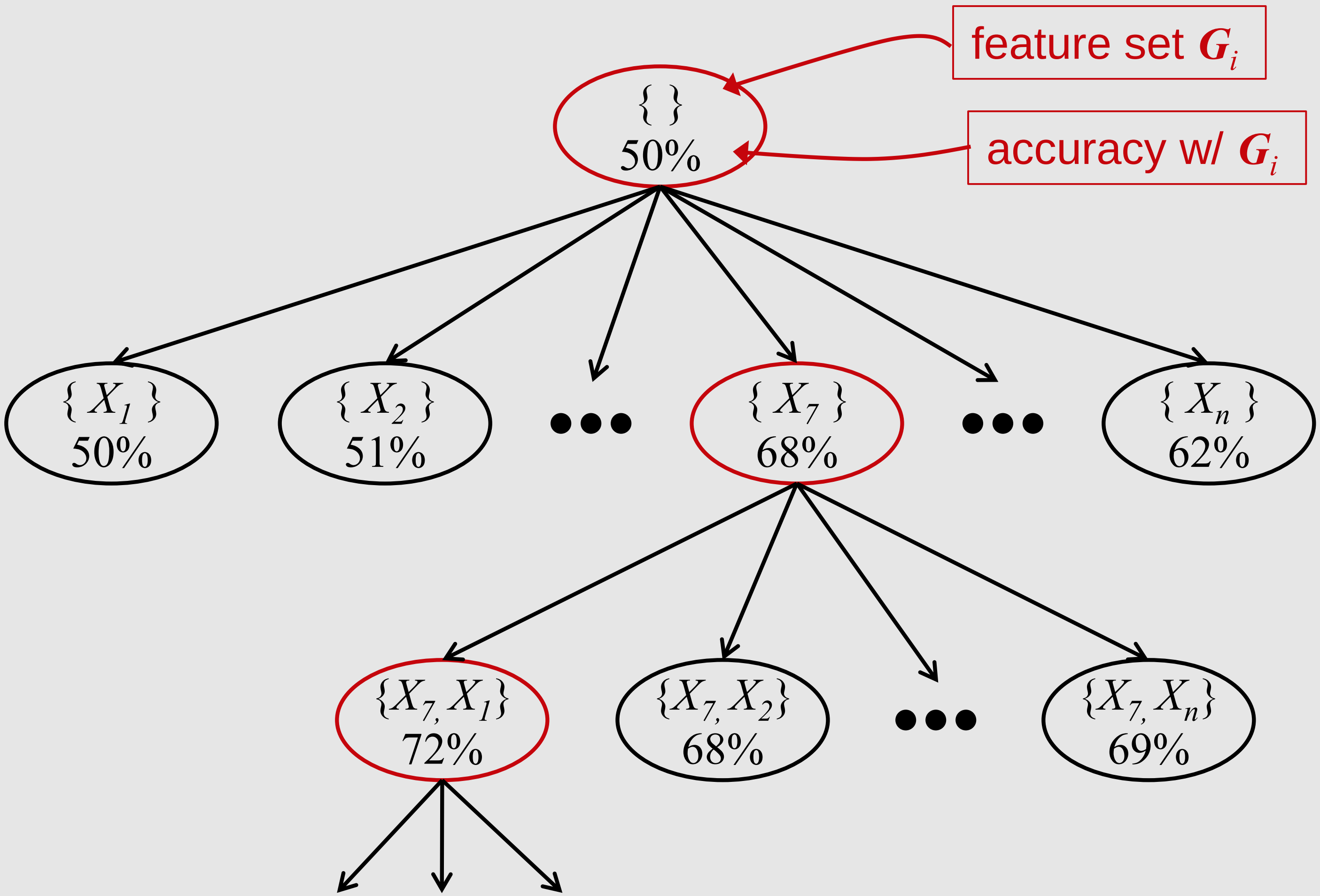
Forward selection

Given: feature set $\{X_i, \ldots, X_n\}$, training set D, learning method L

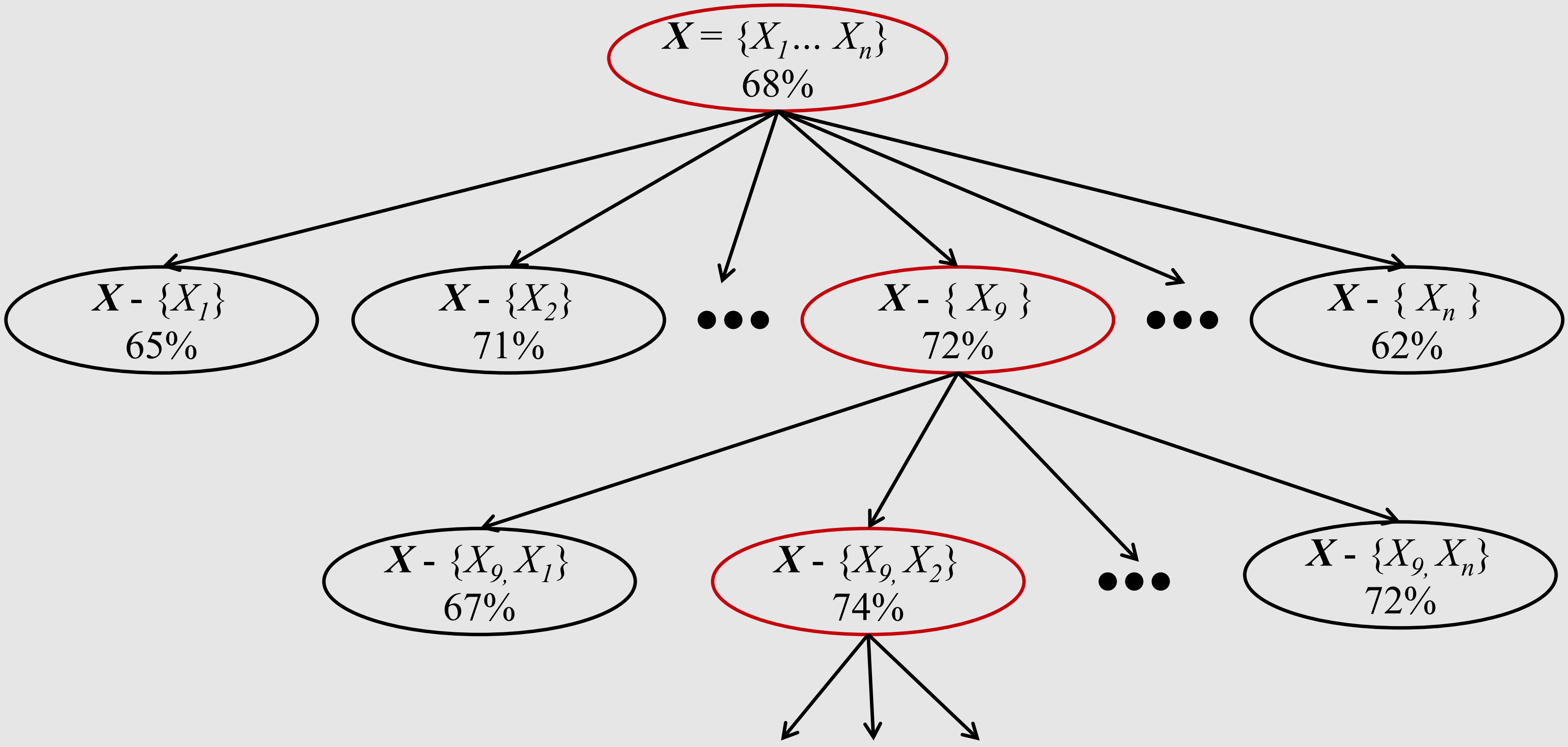
$\boldsymbol{F} \leftarrow \{\ \}$
while score of $\boldsymbol{F}$ is improving
  for $i \leftarrow 1$ to n do
    if $X_i \notin \boldsymbol{F}$
      $\boldsymbol{G}_i \leftarrow \boldsymbol{F} \cup \{X_i\}$
      $Score_i =$ Evaluate($\boldsymbol{G}_i$, L, D)
  $\boldsymbol{F} \leftarrow \boldsymbol{G}_b$ with best $Score_b$
return feature set $\boldsymbol{F}$

scores feature set G by learning model(s) with L and assessing its (their) accuracy

Forward selection



Backward elimination



Forward selection vs. backward elimination

- both use a hill-climbing search

forward selection

- efficient for choosing a small subset of the features
- misses features whose usefulness requires other features (feature synergy)

backward elimination

- efficient for discarding a small subset of the features
- preserves features whose usefulness requires other features

Feature selection via shrinkage (regularization)

- instead of explicitly selecting features, in some approaches we can bias the learning process towards using a small number of features

- key idea: objective function has two parts
  - term representing error minimalization
  - term that "shrinks" parameters toward 0

Linear regression

- consider the case of linear regression

$$f(\boldsymbol{x}) = w_0 + \sum_{i=1}^{n} x_i w_i$$

- the standard approach minimizes sum squared error

$$E(\boldsymbol{w}) = \sum_{d \in D} \left(y^{(d)} - f\left(\boldsymbol{x}^{(d)}\right) \right)^2$$

$$= \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i \right)^2$$

Ridge regression and the Lasso

- Ridge regression adds a penalty term, the L_2 norm of the weights

$$E(\boldsymbol{w}) = \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i \right)^2 + \lambda \sum_{i=1}^{n} w_i^2$$

- the Lasso method adds a penalty term, the L_1 norm of the weights

$$E(\boldsymbol{w}) = \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i \right)^2 + \lambda \sum_{i=1}^{n} |w_i|$$

Lasso optimization

$$\arg\min_{\boldsymbol{w}} \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i \right)^2 + \lambda \sum_{i=1}^{n} |w_i|$$

- this is equivalent to the following constrained optimization problem (we get the formulation above by applying the method of Lagrange multipliers to the formulation below)

$$\arg\min_{\boldsymbol{w}} \sum_{d \in D} \left(y^{(d)} - w_0 - \sum_{i=1}^{n} x_i^{(d)} w_i \right)^2 \text{ subject to } \sum_{i=1}^{n} |w_i| \leq t$$

Ridge regression and the Lasso

β's are the weights in this figure

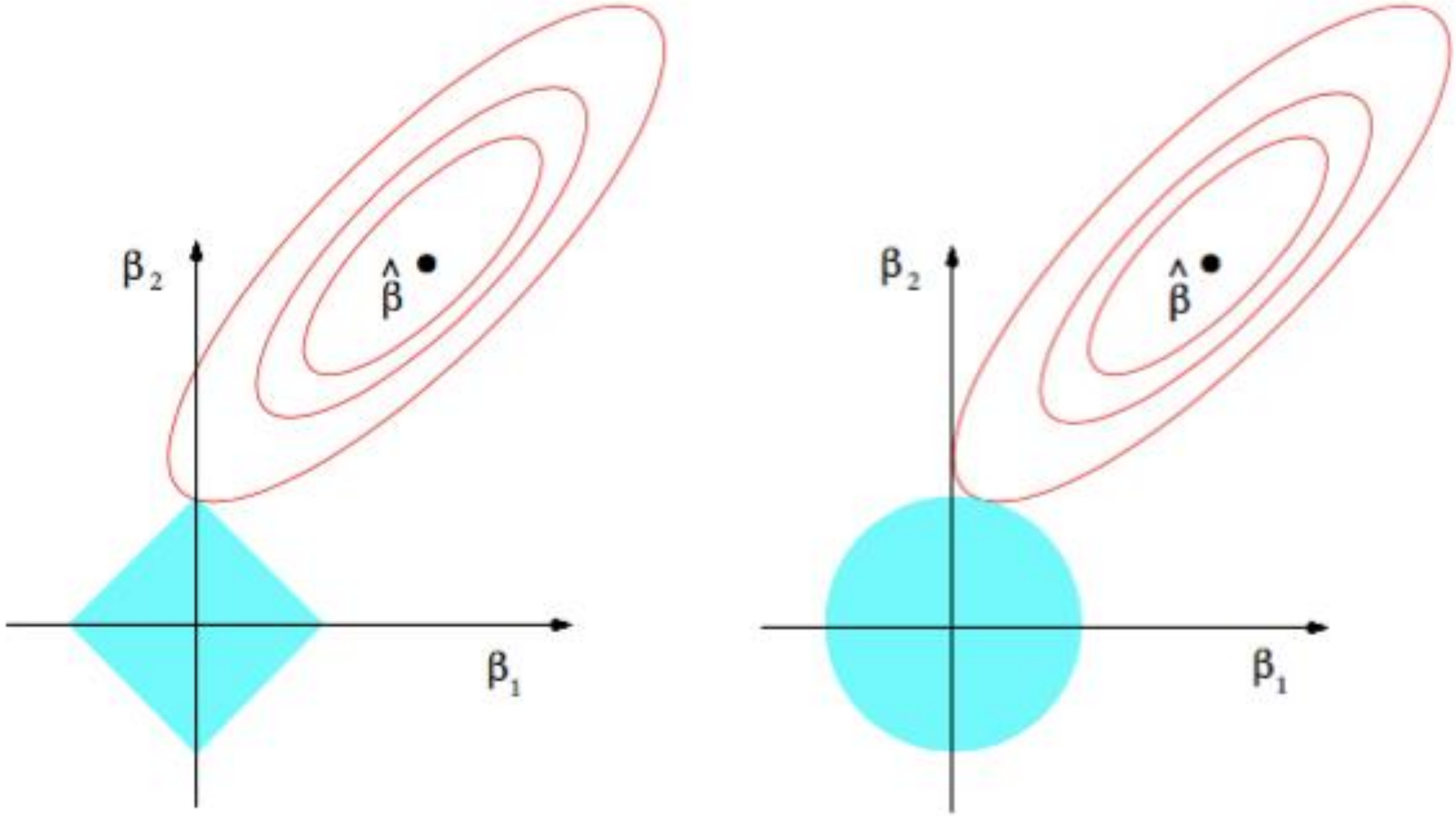


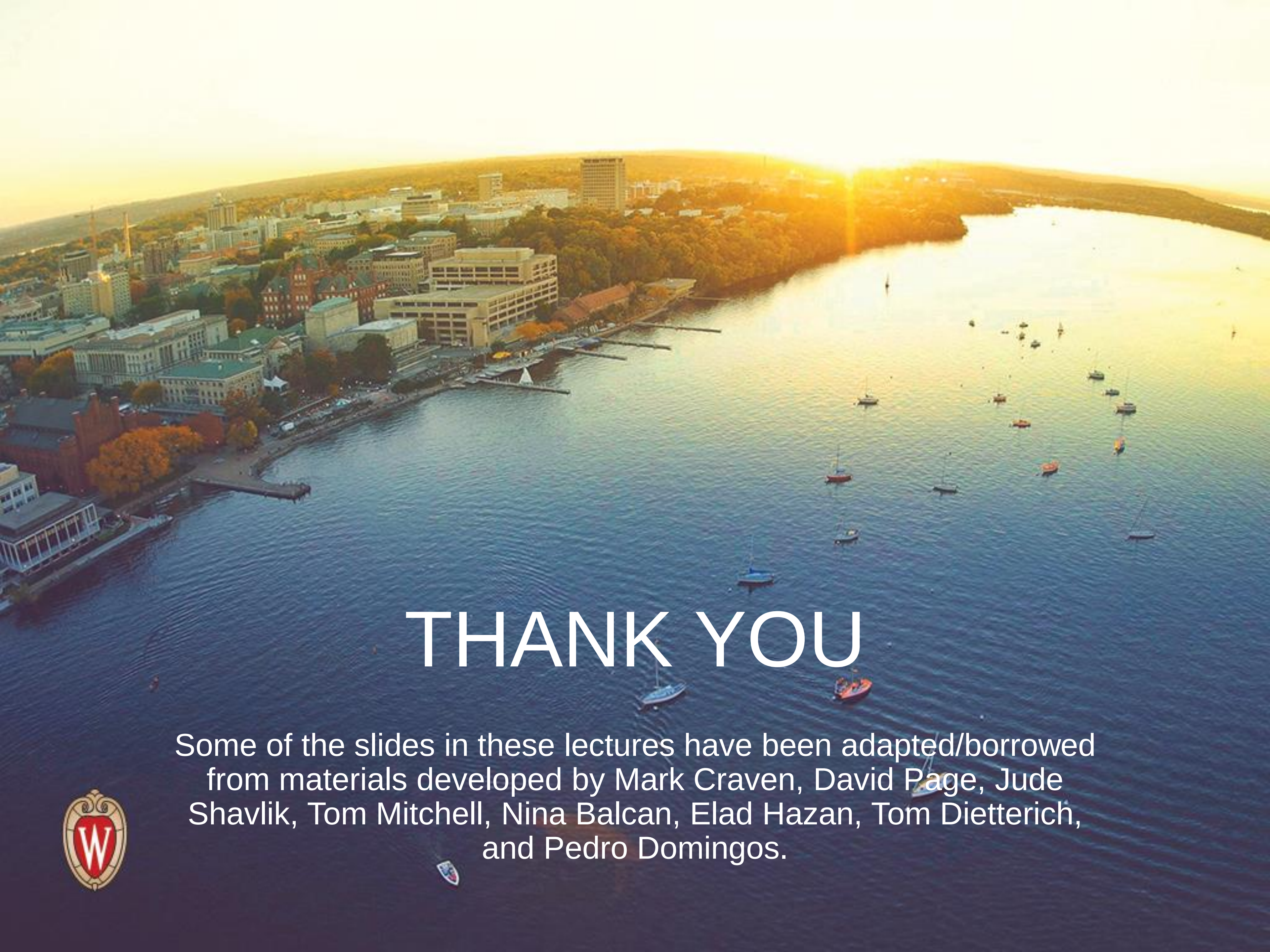
FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \le t$ *and* $\beta_1^2 + \beta_2^2 \le t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Feature selection via shrinkage

- Lasso (L_1) tends to make many weights 0, inherently performing feature selection
- Ridge regression (L_2) shrinks weights but isn't as biased towards selecting features
- L_1 and L_2 penalties can be used with other learning methods (logistic regression, neural nets, SVMs, etc.)
- both can help avoid overfitting by reducing variance
- there are many variants with somewhat different biases
  - elastic net: includes L_1 and L_2 penalties
  - group lasso: bias towards selecting defined groups of features
  - fused lasso: bias towards selecting "adjacent" features in a defined chain
  - etc.

Comments on feature selection

- filtering-based methods are generally more efficient
- wrapper-based methods use the inductive bias of the learning method to select features
- forward selection and backward elimination are most common search methods in the wrapper appraoach, but others can be used [Kohavi & John, *Artificial Intelligence* 1997]
- feature-selection methods may sometimes be beneficial to get
  - more comprehensible models
  - more accurate models
- for some types of models, we can incorporate feature selection into the learning process (e.g. L_1 regularization)
- dimensionality reduction methods may sometimes lead to more accurate models, but often lower comprehensibility



THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.