Variable Selection

6.783, Biomedical Decision Support

Lorenzo Rosasco
(lrosasco@mit.edu)
Department of Brain and Cognitive Science- MIT

November 2, 2009

About this class

- Why selecting variables
- Approaches to variable selection
- Sparsity-based regularization

Why Selecting Variables?

Often data have with tenths or hundreds thousands variables (computational biology, signal processing, combinatorial chemistry, ...)

- **interpretabilty of the model**
- **data driven representation**
- **compression**
- ....

A Useful Example

Biomarker Identification

Set up:

- ℓ patients belonging to 2 groups (say two different diseases)
- n measurements for *each* patient quantifying the expression of n genes

Goal:

- learn a classification rule to predict occurrence of the disease for future patients
- detect which are the genes responsible for the disease

$n \gg \ell$ paradigm

typically ℓ is in the order of tens and n of thousands....

Some Notation

Measurement matrix

Let X be the $\ell \times n$ measurements matrix.

$$X = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_\ell^1 & \dots & \dots & \dots & x_\ell^n \end{pmatrix}$$

- ℓ is the number of examples
- n is the number of variables

For each patient we have a response (output) $y \in R$ or $y = \pm 1$.
In particular we are given the responses for the training set

$$Y = (y_1, y_2, \dots, y_\ell)$$

Approaches to Variable Selection

Which variables are "*relevant*"? Different approaches are based on different way to specify what is relevant.

- Filters methods.
- Wrappers.
- Embedded methods.

(see "Introduction to variable and features selection" Guyon and Elisseeff ’03)

Embedded Methods

The selection procedure is **embedded** in the training phase.

An intuition

what happens to the generalization properties of empirical risk minimization as we subtract variables?

- if we keep all the variables we probably overfit
- if we take just a few variables we are likely to oversmooth

Selecting variables and approximating functions

Suppose the output is a linear combination of the variables

$$f(x) = \sum_{i=1}^{n} \beta_i x^i = \langle \beta, x \rangle$$

each coefficient β_i can be seen as a weight on the i-th variable.

- The intuition is that as we discard variable we make the model simple and avoid overfitting.
- Brute force approach **try ALL** possible subsets $\Rightarrow$ unfeasible!

Can we use regularization?

Tikhonov Regularization

$$\min_{\beta \in \mathbb{R}^n} \{ \frac{1}{\ell} \sum_{j=1}^{\ell} V(y_j, \langle \beta, x_j \rangle) + \lambda \sum_{i=1}^{n} \beta_i^2 \}$$

Sparsity and Ovefitting

Tikhonov regularization leads to solutions with good generalization properties.

$\Rightarrow$ How about variable selection?

In general all the β_i will be different from zero.

Selection property is not built in and can be done only adding a thresholding step.

Sparsity

Define the "zero"-norm (not a real norm) as

$$\|\beta\|_0 = \#\{i = 1, \ldots, n \mid \beta_i \neq 0\}$$

It is a measure of how "complex" is f and of how many variables are important.

Is it a good way to define sparsity?

ℓ_0 Regularization

If want to select variables we can look for

$$\min_{\beta \in \mathbb{R}^n} \{ \frac{1}{\ell} \sum_{j=1}^{\ell} V(y_j, \langle \beta, x_j \rangle) + \lambda \|\beta\|_0 \}$$

$\Rightarrow$ This is as difficult as trying all possible subsets of variables.

Can we find meaningful approximations?

Approximate solutions

Two main approaches

1. Convex relaxation (ℓ_1 regularization,Lasso, Basis Pursuit)
2. Greedy schemes (boosting algorithms, matching pursuit...)

We mostly discuss the first class of methods.

Convex Relaxation

A natural approximation to ℓ_0 regularization is given by:

$$\frac{1}{\ell}\sum_{j=1}^{\ell} V(y_j, \langle \beta, x_j \rangle) + \lambda \|\beta\|_1$$

where $\|\beta\|_1 = \sum_{i=1}^{n} |\beta^i|$.

If we choose the square loss

$$\frac{1}{\ell}\sum_{j=1}^{\ell}(y_j - \langle \beta, x_j \rangle)^2 = \|Y - X\beta\| 2$$

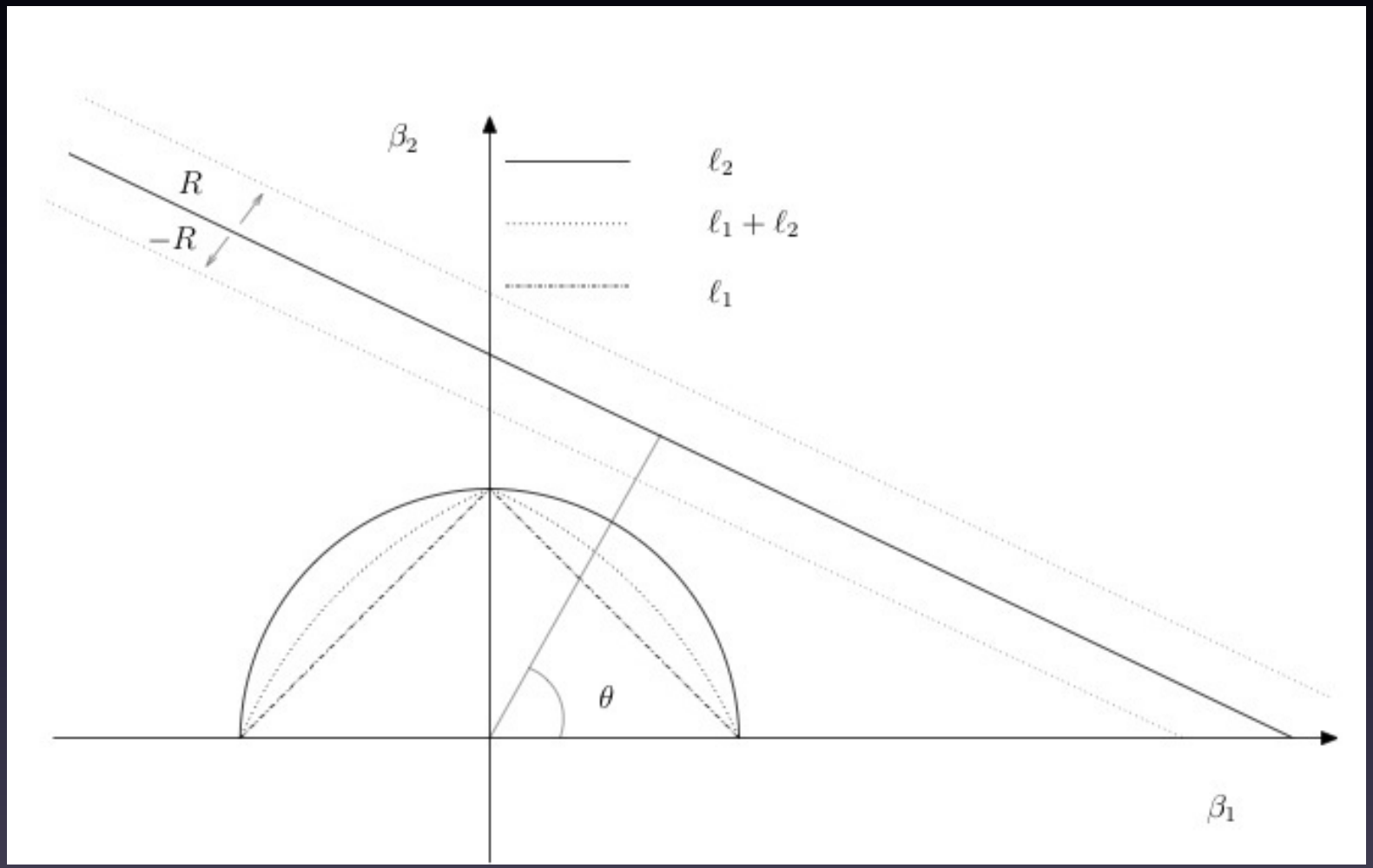
such a scheme is called **Basis Pursuit** or **Lasso**.

What is the difference with Tikhonov regularization?

- We have seen that Tikhonov regularization is a good way to avoid overfitting.
- Lasso provides sparse solution Tikhonov regularization doesn't.

Why?

Geometry of the Problem



Solvers

Many different way to solve the corresponding optimization problem.

- Iterative Thresholding Algorithms
- Homotopy methods (LARS)
- Interior point methods

Coefficient Shrinkage

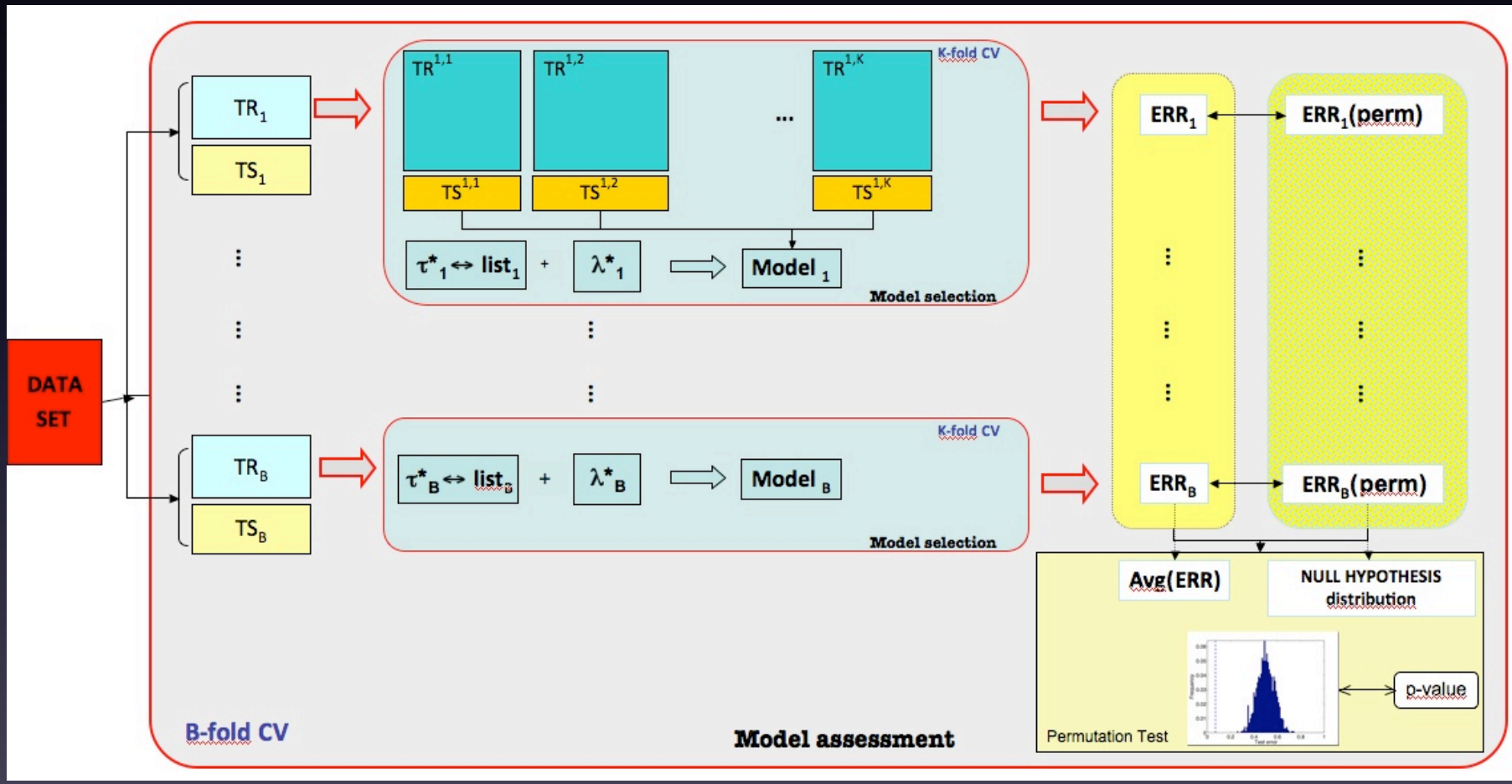
in practice one often observe excessive shrinkage of the coefficients: Least Squares on selected variables especially with really few examples cross validation is tricky...

- Fix λ. Solve ℓ_1 regularization.
- Compute Least Squares (LS) on selected variables.

Note:
Cross validation should be done for this *two-steps* algorithm.
If we have many selected variables. LS can be replaced by Tikhonov (need to choose two parameters then).

Model Selection

Especially with really few examples cross validation is tricky.



Regularization Path

With Iterative thresholding: use a continuation strategy when computing regularization path:

- Fix decreasing sequence of values for λ.
- Start from big values of λ.
- Use found solution to initialize search for the solution for the next smaller λ.

Remarks on Small Samples

With very few samples we should check statistical significance of the obrained results:

- Permuation Test
- Bootstrap
- ...

Beyond ℓ_1

- **About Uniqueness**: the solution of ℓ_1 regularization is not unique. Note that the various solution have the **same prediction properties** but **different selection properties.**
- **Correlated Variables**: If we have a group of correlated variables the algorithm is going to select just one of them. This can be bad for interpretability but maybe good for compression.

Elastic Net Regularization

One possible way to cope with the previous problems is to consider

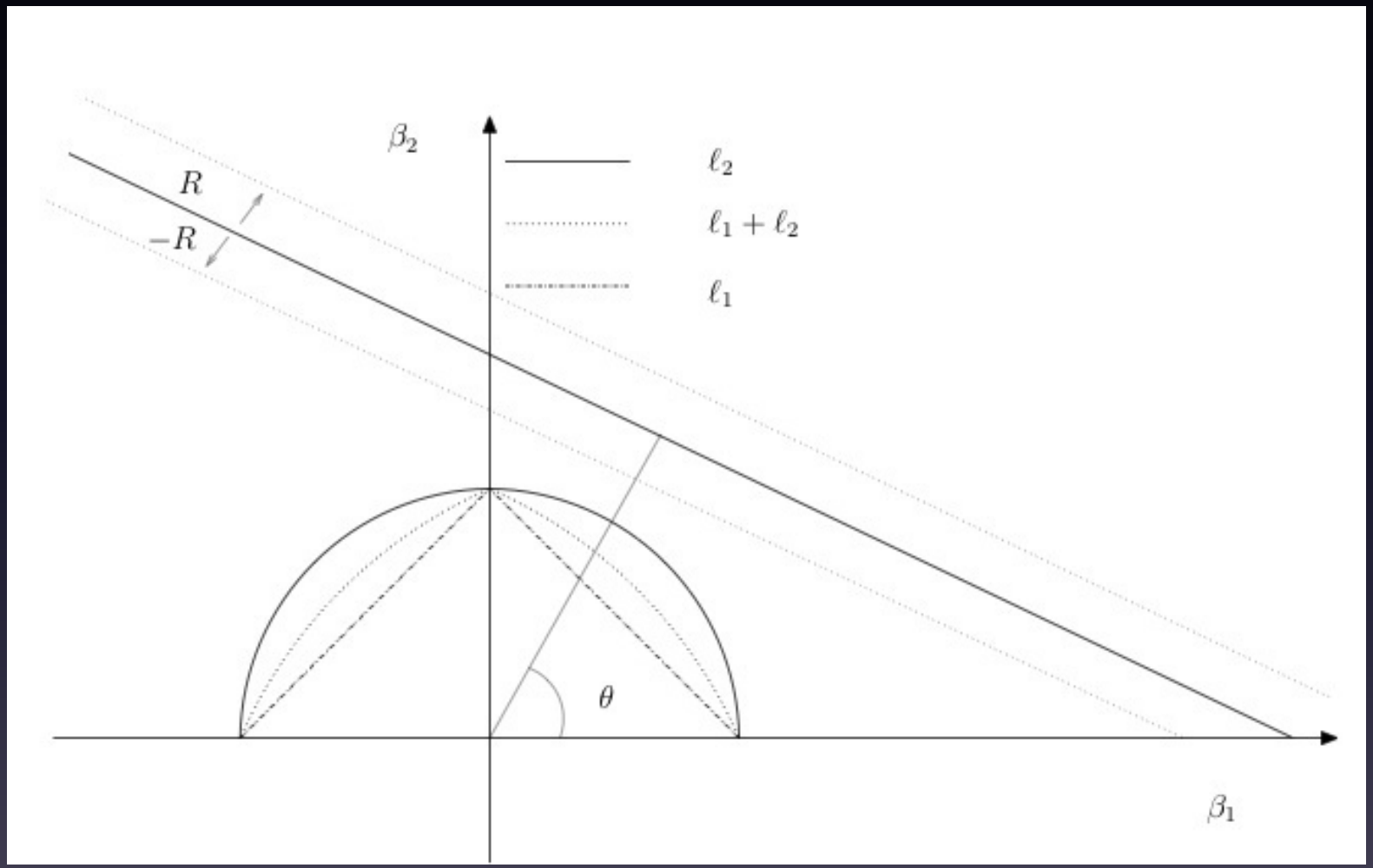
$$\min_{\beta \in \mathbb{R}^p} \|Y - \beta X\|^2 + \lambda(\alpha \|\beta\|_1 + (1-\alpha) \|\beta\|_2^2).$$

- λ is the regularization parameter.
- α controls the amount of correlation in the variables.

Elastic Net Regularization (cont.)

- The ℓ_1 term promotes sparsity and the ℓ_2 term correlation.
- The solution is unique.
- A whole group of correlated variables is selected rather than just one variable in the group.

Geometry of the Problem



Identifying Modules of Correlated Genes

- Fix α big. Tune λ and solve.
- Keep λ fixed and solve for decreasing values of α.

$\lambda = 0.06$ $\gamma = 5 \cdot 10^{-6}$ $\lambda\varepsilon$	test error (test set size: 51)	# of selected genes	intersection w. genes selected for bigger ε
0	5	19	95%
0.001	5	20	100%
0.0025	5	25	100%
0.005	4	31	97%
0.01	5	40	98%
0.1	6	85	94%
1	5	121	–

Identifying Modules of Correlated Genes (cont.)

- Use genes in the first list as initial centroids for clustering.
- Agglomerative clustering of the second list using the centroids.
- Compute new centroids.
- Agglomerative clustering of the third list using the centroids.

ϵ	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
0	1	1	1	1	1	1	1	1	1	1
$4 \cdot 10^3$	2	2	1	1	2	1	1	1	1	1
$3 \cdot 10^2$	2	4	2	1	2	2	1	3	3	1
$2 \cdot 10^1$	7	9	5	2	4	2	1	8	5	3
1.7	12	28	21	3	8	4	6	17	5	7
13	27	40	48	3	10	7	6	21	8	9
100	31	49	60	3	11	8	6	22	10	11

Summing up

- Variable selection can be embedded in the training phase using regularization
- Model selection is crucial
- Correlation: Elastic Net

Approximate solutions

Two main approaches

1. Convex relaxation (ℓ_1 regularization,Lasso, Basis Pursuit)
2. Greedy schemes (boosting algorithms, matching pursuit...)

We mostly discuss the first class of methods.

Greedy Methods

Similar techniques, different names:

- statistics - forward stagewise regression,
- approximation theory - greedy algorithms,
- learning - boosting methods,
- signal processing - projection pursuit methods.

Greedy Methods: Approach

Based on iterating two steps:
After some initialization:

1. selection a feature
2. find/update the solution.

These schemes proceed incrementally and are not based on a global optimization procedure.

Orthogonal Matching Pursuit

Consider the following iteration.

Set $f_0 = 0$
for $t = 1 : t_{stop}$
$r_t = Y - f_{t-1}$;
$t = argmax_{j=1,\ldots,n} \langle X^j, r_t \rangle$
$\beta_t = \langle r_{t-1}, X^t \rangle$
$f_t = f_{t-1} + X^t \beta_t$
end

- The number of iterations is the regularization parameter (early stopping)
- Each iteration selects one variable

Some comments on greedy algorithms

- Under suitable assumptions the algorithm can be shown to approximate the ℓ_0 norm solution
- The number of iterations corresponds the number of features selected
- The computations cost increases as we have to solve larger least squares problem
- We do not have to work with the whole data matrix

Approaches to Variable Selection

Which variables are "*relevant*"? Different approaches are based on different way to specify what is relevant.

- Filters methods.
- Wrappers.
- Embedded methods.

(see "Introduction to variable and features selection" Guyon and Elisseeff ’03)

Filter methods

- Select variables by ranking them
- Ranking is performed according to some criterium

Ranking criteria (1)

The idea is to rank the variables X^i, $i = 1, \ldots, n$ according to some criteria $r(i)$

- **Correlation criteria:**
- Rank variables according to a correlation score

$$r(i) = \frac{cov(X^i, Y)}{\sqrt{Var(X^i)Var(Y)}}$$

- Similar criteria are Fisher correlation, t-test correlation, ...
- Such methods detect linear dependencies between variable and target.

Ranking criteria (2)

Single variable classifiers:

- Select variables according to their *individual* predictive power
- The *predictive power* can be expressed in terms of error rate, or related criteria:
  - false positive rate (fpr) - false negative rate (fnr)
  - ROC curves (false pos. rate vs hit rate) $\Rightarrow$ equal error rate or area under the curve
- In the case of similarly performing variables other ranking criteria are needed

Ranking criteria (3)

information based criteria

- based on empirical estimates of mutual the which measure of dependency between random variables:

$$r(i) = \int_{X^i} \int_{Y} P(X^i, Y) \log \frac{P(X^i, Y)}{P(X^i)P(Y)}$$

- $P(X^i)$ $P(Y)$ $P(X^i, Y)$ are unknown

Filter methods: pros and cons

Cons

- Filter methods usually consider individual variables
- They are prone to provide redundant sets of features
- No explicit selection is embedded in the method: some thresholding is needed.

Pros

- Straightforward and simple
- Scalable
- Good empirical performance

Wrapper methods

An algorithm of choice is used (as a *black box*) to evaluate the importance of a set of variables for the given task.

One needs to define

- how to search the space of all possible variable subsets
- how to assess prediction performance of the learning machine
  $\Rightarrow$ usually a validation set or cross-validation is adopted
- which predictor to use
  $\Rightarrow$ many machines have been adopted in the literature (decision trees, näive Bayes, LS, SVM, ...)

Wrapper methods

$\Rightarrow$ a strategy is needed to explore the variable space.

A brute force approach is infeasible even for relatively few features (NP-hard problem).

two main approaches

- **forward selection**: start with no variables and incrementally add them...
- **backward elimination**: start with all the variables and proceed iteratively to discard them...

...until some stopping criteria is satisfied. For example a fixed number of variables, or a certain validation error, is reached.

Both methods yield nested subsets

A wrapper method: RFE (1)

Recursive Feature Elimination (RFE) was proposed in 2002 as a way to rank genes. The original formulation applies linear SVMs as black box predictors — SVM-RFE

- **init:** input measurements matrix

$$X_0 = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_\ell^1 & \dots & \dots & \dots & x_\ell^n \end{pmatrix}$$

list of survived features $s = [1, \dots, n]$

feature ranked list $r = []$

A wrapper method: RFE (2)

Repeat until $s = []$

- restrict the training samples to survived variables

$$X = X_0(:, s)$$

- train the classifier on X obtaining the solution α
  Performance evaluation of current s over a validation set
- compute the weight vector and the ranking criteria
  in the original work (linear SVM-RFE)

$$\mathbf{w} = \sum_k \alpha_k y_k \mathbf{x}_k$$

$$c_i = w_i^2$$

- find the feature f with smallest ranking
- add f on top of the ranked list r
- remove f from s

Wrapper methods: pros and cons

Pros

- Using the learning machine as a black box they are universal and simple

Cons

- They require various re-training and parameter tuning
- They require the implementation of efficient search mechanisms
- Selection is not embedded and thresholding is needed