

**INPUT SELECTION  
AND LEARNING IN  
INPUT SPACE**

Amir Atiya

Dept Computer Engineering  
Cairo University  
Giza, Egypt  
[amir@alumni.caltech.edu](mailto:amir@alumni.caltech.edu)

## The Input Selection (or Feature Selection) Problem:

- With more and more applications becoming more complex, one is confronted with more and more potential inputs, hence the difficult task of selecting the best inputs.
- Curse of dimensionality.

## Two Types of Inputs we Have to Remove:

- Irrelevant inputs: Inputs that do not contribute to the prediction of the target values.
- Correlated inputs: extra unneeded degrees of freedom.

### Irrelevant Inputs:

- Consider the linear regression case.
- How to determine an irrelevant input?

Let  $N$  and  $M$  denote the number of inputs and training data points respectively.

Let  $x_n(m)$  denote the input  $n$  of the  $m^{\text{th}}$  training pattern, and let  $y(m)$  be the target output for training pattern  $m$ .

Let  $x_n = (x_i(1), \dots, x_i(M))$ ,  
 $y = (y(1), \dots, y(M))$ .

**Theorem:** Assume the  $N$  input vectors  $x_n$  are distributed according to a spherical distribution, i.e.  $p(x_n) = \text{fn}(\|x_n\|)$ . Also, assume  $y$  is distributed according to any spherical distribution, and let  $y, x_1, \dots, x_N$  be independent. Then, the distribution of  $\mathcal{E} \equiv E/\|y\|^2$  (normalized sum of square error) is given by

$$p(\mathcal{E}) = \text{Beta}\left(\frac{M-N}{2}, \frac{N}{2}\right)$$

$$\equiv \frac{\Gamma\left(\frac{M}{2}\right)}{\Gamma\left(\frac{M-N}{2}\right)\Gamma\left(\frac{N}{2}\right)} \mathcal{E}^{\frac{M-N}{2}-1} (1-\mathcal{E})^{\frac{N}{2}-1}$$

and the expectation is given by

$$\bar{\mathcal{E}} = 1 - \frac{N}{M}$$

- For problems with hundreds of inputs the combinatoric nature of the input selection problem necessitates quick screening methods to narrow down the choice. ■
- Experiment: How is the individual input performance indicative of how an input will fare in the group (of chosen inputs).
- We took  $M = 20$ . Only inputs with a specific fixed individual performance are considered ( $= E/\|y\|^2$ ). We performed a Monte Carlo experiment.
- The results indicate the value of looking at each input individually.

## Correlated Inputs:

- Check correlation matrix:

$$R = \begin{pmatrix} 1 & 0.8 & 0.6 & -0.1 \\ 0.8 & 1 & 0.3 & 0.3 \\ 0.6 & 0.3 & 1 & 0.5 \\ -0.1 & 0.3 & 0.5 & 1 \end{pmatrix}$$

## We Propose a Method for Visualizing Correlations:

Let  $R_{ij} \equiv$  correlation coefficient between inputs  $i$  and  $j$ : “a measure of similarity”. Then,

$1 - |R_{ij}| \equiv$  a measure of distance between inputs  $i$  and  $j$ .

- We model the inputs as points in 2 or 3-dimensional space.
- The distance between these points are designed to be as close as possible to  $1 - |R_{ij}|$
- This way, we “visualize” all inputs and their similarity relationships in a two-dimensional graph.

Method: Define the objective function:

$$J = \sum_{i,j} \left[ 1 - |R_{ij}| - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right]^2$$

Minimize using gradient descent.

## Another Method:

- Shift all inputs so that they are zero mean. Then

$R_{ij} = E(x_i x_j) / \sqrt{E(x_i^2) E(x_j^2)} \equiv$  dot product of two normalized vectors

$\cos(\theta) = y_1^T y_2$  where  $|y_1| = |y_2| = 1 \longrightarrow$   
 $\cos(\theta)$  represents  $R_{ij}$ .

- Every input is represented as a vector in some space (2D or 3D).
- Find vectors  $y_i \in \mathcal{R}^K$  ( $K = 2$  or  $3$ ) such that the following function is minimized:

$$J = \sum_i \sum_j [R_{ij} - y_i^T y_j]^2, \quad \text{s.t. } \|y_i\|^2 = 1$$

- We have developed an algorithm to minimize  $J$ . The problem is equivalent to finding the best low rank approximation for  $R$ .

## Typical Input Selection Algorithms:

- 1) Forward selection algorithm.
- 2) Backward selection model.
- 3) Forward/backward selection algorithm. ■
- 4) Branch and Bound method.

## **Typical Learning Paradigm:**

Choose very few inputs, then apply sophisticated nonlinear model to learn input/output relationship. ■

## **Sparse Basis Selection Methods:**

- Developed for the signal compression problem.
- Uses a very large dictionary of basis signals.
- Forward or backward selection to select the most effective few of basis selection.
- Combines basis signals linearly.

- We extend the basis selection model to nonlinear regression.
- We use simple linear model on an expanded nonlinearly transformed input space, e.g.

$$u_1, u_2, u_3, \dots, e^{\alpha u_1}, e^{\alpha u_2}, \dots, \\ \log|u_1|^2, \dots, u_1^\beta, u_2^\beta, u_1 u_2, \dots$$

The basis, or selected nonlinear terms, are chosen by forward or backward selection model. ■

Thus, learning is through two means: updating weights (of the linear model) and dynamically selecting the nonlinearities that enter into the model.

We developed a new adaptive algorithm that ■ selects inputs in an adaptive manner and updates the regression weights recursively.

It uses forward/backward selection, and improves the speed considerably.

## Advantages:

- 1) Learning is almost guaranteed:
  - Need to solve straightforward linear regression problem:  $w = (X^T X)^{-1} X^T y$ .
  - Forward or backward selection are very reliable methods.
- 2) This is particularly helpful in applications that need adaptive or online learning without human supervision. ■
- 3) Examples of such applications are prediction problems in communications networks, e.g. traffic prediction, delay prediction etc. ■

## Application to Video Traffic Flow Forecasting: ■

Predicting MPEG-coded source traffic flow can help in:

- Improving QoS and streaming characteristics of real-time video.
- Dynamic bandwidth allocation.

Input dictionary: around 100 inputs such as moving averages, standard deviations, several nonlinear transformations of these.