Filter Methods

Part II : Information Theoretic Filters

Mutual Information

X is relevant to Y if they are dependent, i.e. $p(y|x) \neq p(y)$, or...

$$p(xy) \neq p(x)p(y)$$

So let's measure the KL-divergence between these distributions:

$$J(X_k) = I(X_k; Y) = \sum_{x \in X_k} \sum_{y \in Y} p(xy) \log \frac{p(xy)}{p(x)p(y)}$$

We rank features by their score J.

REMEMBER THIS?

Information Theory

Basic unit of information : entropy, denoted $H(X)$.

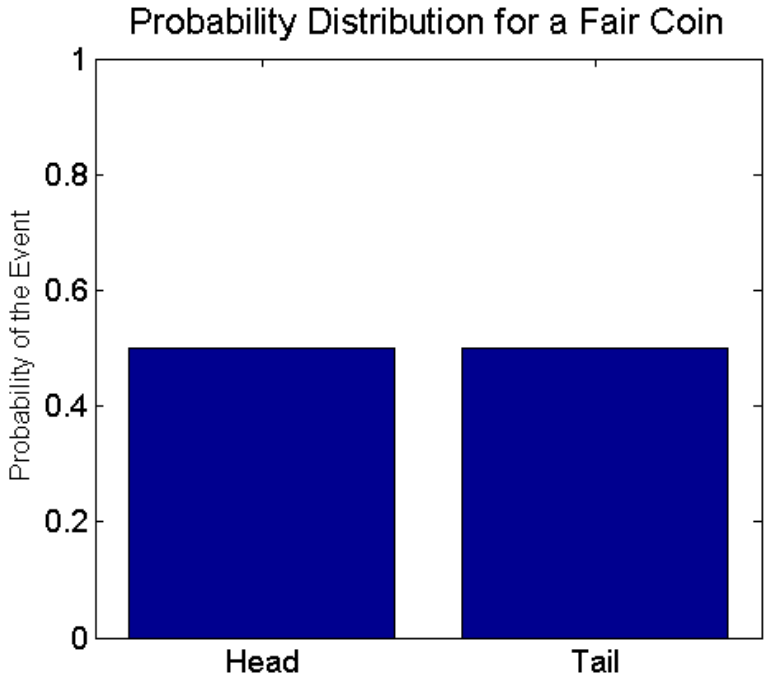
Measures amount of uncertainty in X.

Definition...

$$H(X) = \sum_i p(x_i) \log p(x_i)$$

Sum is over all possible values i that a variable X can adopt.
...$p(x_i)$ is the probability of that particular value occurring.

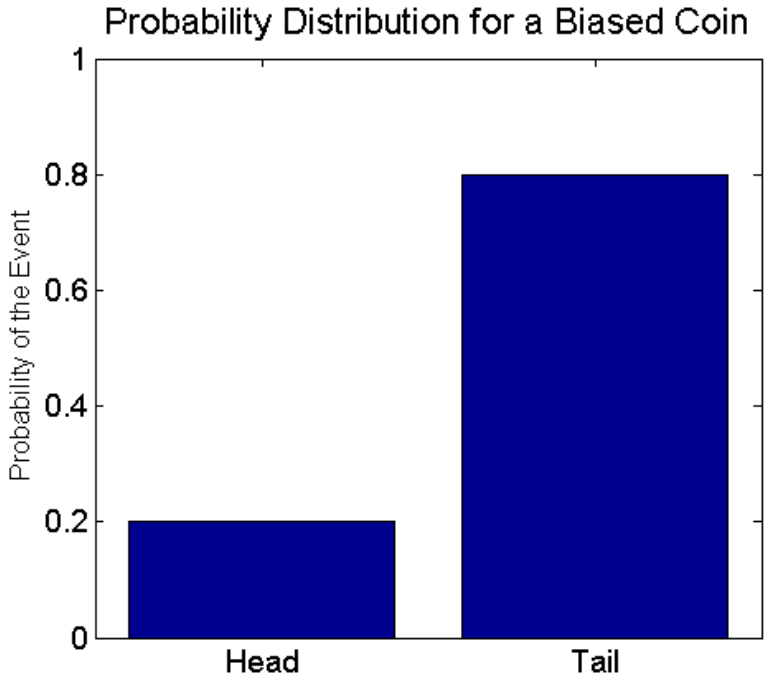
Heads or Tails?



High entropy.

entropy$\sim$uncertainty

Heads or Tails?



Low entropy.

entropy$\sim$uncertainty

Entropy

Probability of fair coin landing heads is $p(x = 1) = 0.5$.

$$
\begin{aligned}
H(X) &= -\Big(p(x=1)\log_2 p(x=1) + p(x=0)\log_2 p(x=0)\Big) \\
&= -\Big(0.5\log_2 0.5 + 0.5\log_2 0.5\Big) \\
&= 1
\end{aligned}
$$

Note $\log_2$ is not necessary, can be any base logarithm.

If using natural log, above $H(X) \approx 0.6931$.

Matlab code for $H(X)$

```
function h = calcentropy(x)

N = length(x);
alphabet = unique(x);

h = 0;
for i = 1:length(alphabet)

  prob_i = sum(x==alphabet(i)) / N;

  h = h + prob_i * log2(prob_i);

end

h = -h;
```

Conditional Entropy

Entropy can be conditioned.
Uncertainty remaining when in X once we know Y.

$$H(X|Y) = \sum_{j \in Y} p(y_j) H(X|Y = y_j)$$

Average $H(X)$ over all possible values that Y can adopt.
...$p(y_j)$ is the probability of that particular value occurring.

Matlab code for $H(X|Y)$

```
% Calculate H(X|Y)
function ch = calcConditionalEntropy(x,y)

N = length(y);
alphabet_y = unique(y);

ch = 0;
for i = 1:length(alphabet_y)

  prob_i = sum(y==alphabet_y(i)) / N;

  %find all x rows where y is this particular value
  sub_x = x(y==alphabet_y(i));

  %weighted average over H(X|Y=y_i)
  ch = ch + prob_i * calcentropy( sub_x );

end
```

Back to Mutual Information

$$J(X_k) = I(X_k; Y) = \sum_{x \in X_k} \sum_{y \in Y} p(xy) \log \frac{p(xy)}{p(x)p(y)}$$

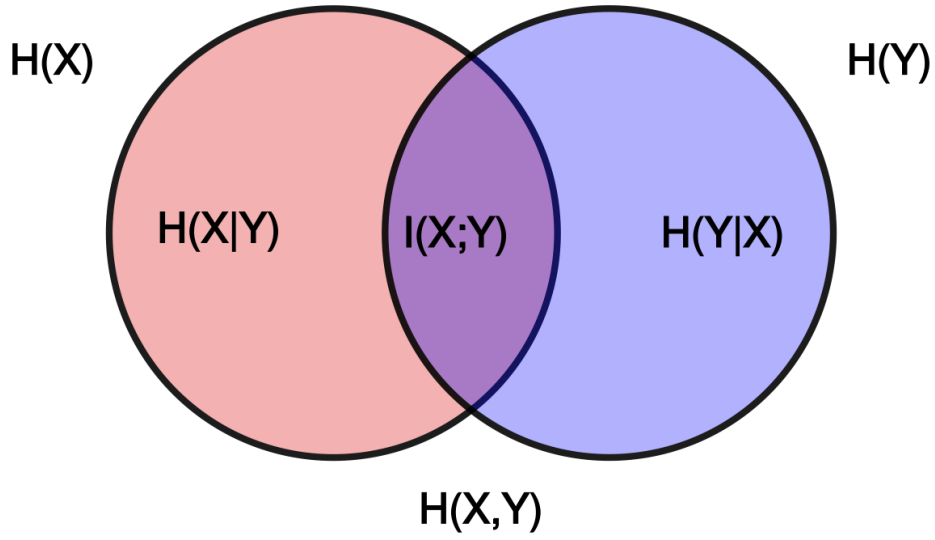
But it can also be written...

$$I(X_k; Y) = H(Y) - H(Y|X_k)$$

"Amount of uncertainty in the label once we known feature X_k".

In decision trees, it is known as "information gain".

Relations...



Mutual Info is always non-negative, so $I(X;Y) \geq 0$.

Mutual Info is symmetrical,

$$I(X;Y) = I(Y;X).$$
$$H(Y) - H(Y|X) = H(X|Y) - H(Y).$$

Properties of Information Theory

It is always non-negative,

$$I(X;Y) \geq 0.$$

It is symmetrical,

$$I(X;Y) = I(Y;X).$$
$$H(Y) - H(Y|X) = H(X|Y) - H(Y).$$

We can have joint mutual information:

$$I(X_1X_2X_3 \ ; \ Y).$$

And conditional mutual information:

$$I(X_1;Y|X_2X_3).$$

You don't need to know this now. Teaser ("anteprima"?) for the final lecture....

What if x, y are approximately Gaussian?

$X, Y \sim N(\mu, \sigma^2)$?

Entropy of a K-dimensional Gaussian is $H(X) = \frac{1}{2} \log(2\pi e)^K |\Sigma|$

We use an alternative form for the mutual info:

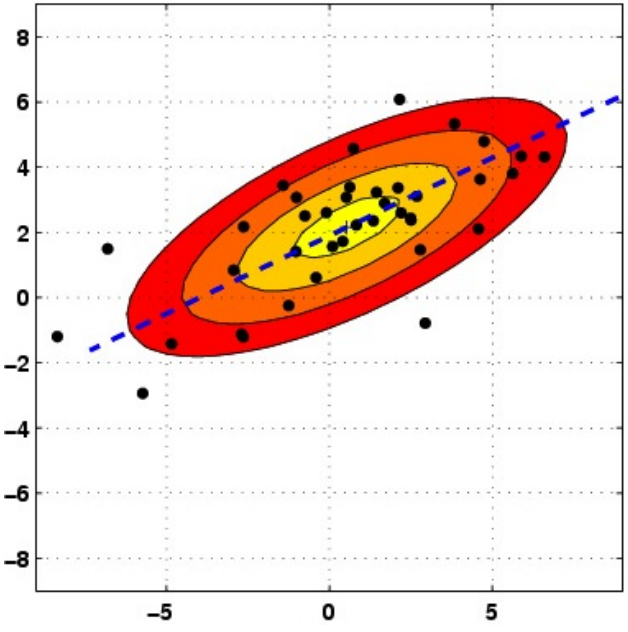
$$
\begin{aligned}
I(X;Y) &= H(X) + H(Y) - H(XY) \\
&= \tfrac{1}{2} \log(2\pi e \sigma_x^2) + \tfrac{1}{2} \log(2\pi e \sigma_y^2) - \tfrac{1}{2} \log(2\pi e)^2 |\Sigma| \\
&= -\tfrac{1}{2} \log(1 - r^2)
\end{aligned}
$$

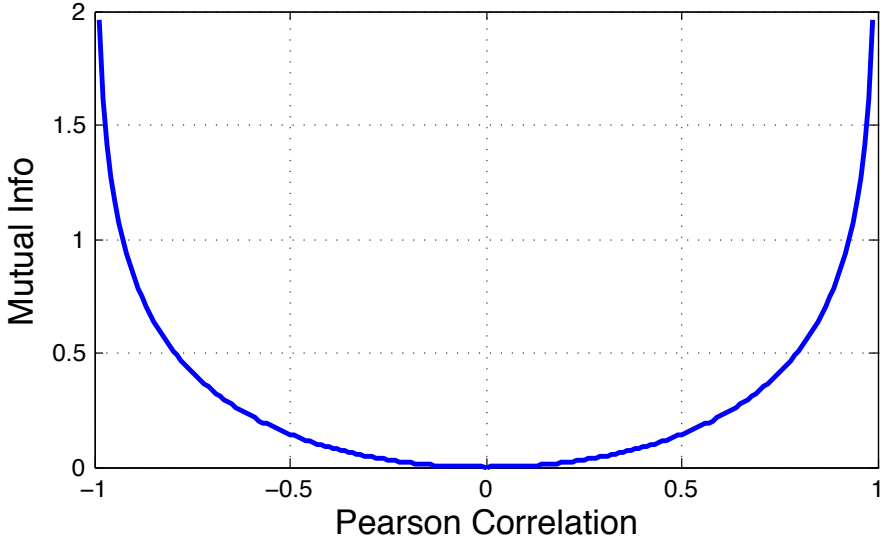
where r is Pearson's Correlation.

What if x, y are approximately Gaussian?

$$I(X;Y) = -\tfrac{1}{2}\log(1-r^2)$$

where r is Pearson's Correlation Coefficient between x and y.



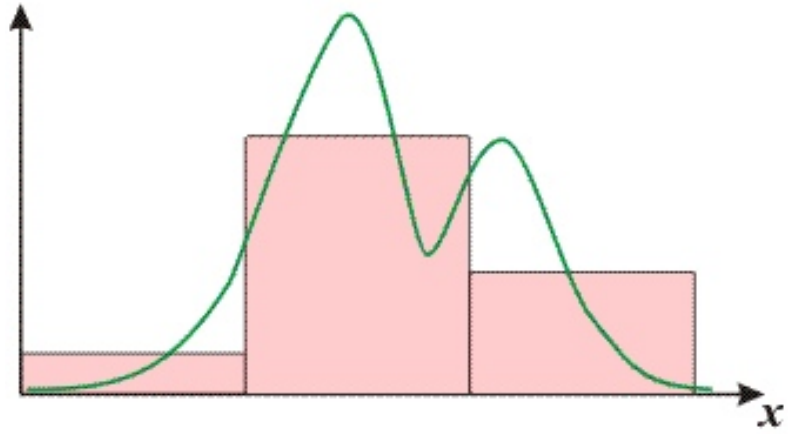


When x, y are Gaussian, MI is monotonic transform of Pearson.

Estimating Mutual Info

Biggest problem of $I(X;Y)$? Estimating it...

Discretize data for feature selection.
Use original data for classifying.



Filter Ranking using Mutual Information

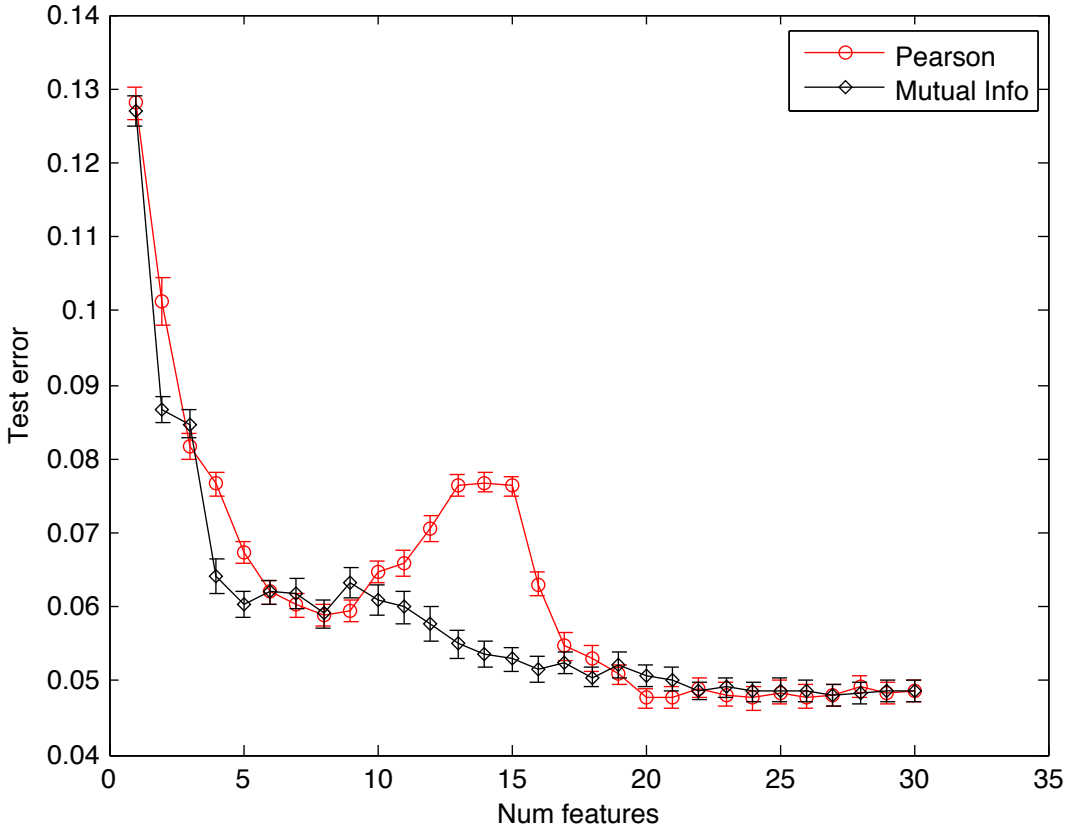
Rank features $X_k, \forall k$ by their values of $J_{mim} = I(X_k; Y)$.
Retain the highest ranked features, discard the lowest ranked.

i	$J(X_k)$
35	0.846
42	0.811
10	0.810
654	0.611
22	0.443
59	0.388
...	...
212	0.09
39	0.05

Cut-off point decided by user, e.g. $|S| = 5$, so $S = \{35, 42, 10, 654, 22\}$.

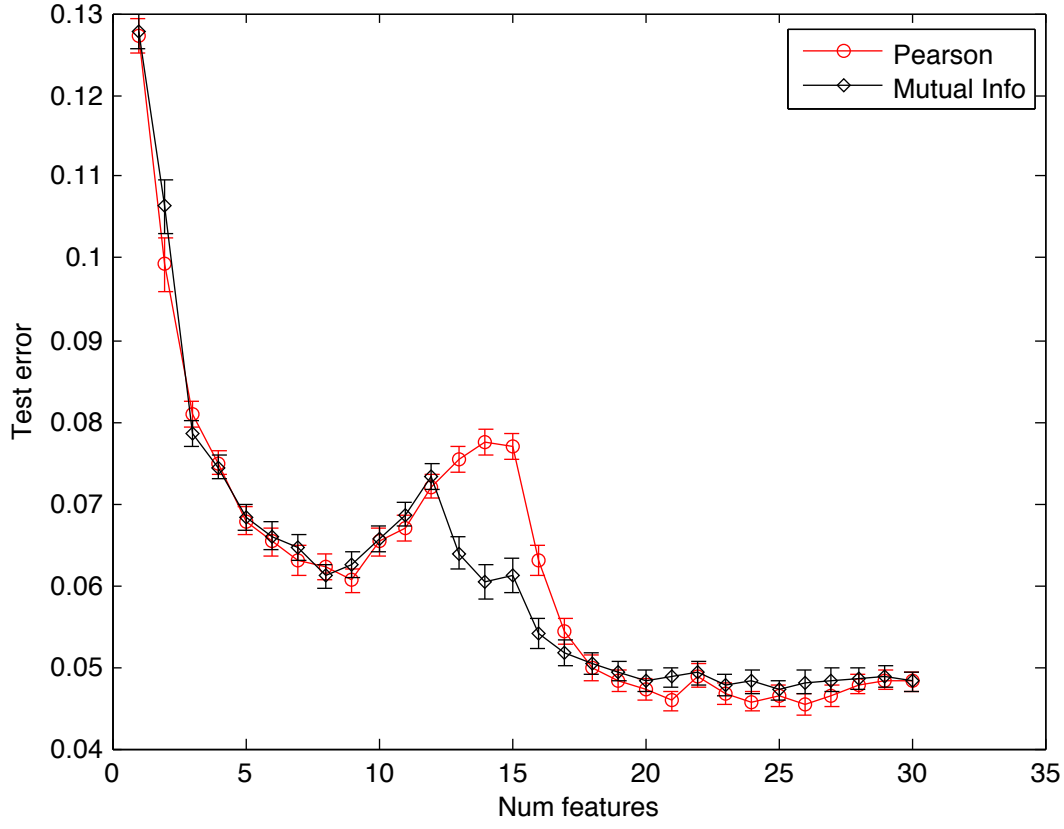
Pearson versus Mutual Info

Breast cancer data, discretized to 2 bins using training data.
Used 1-nn classifier and selected features to measure OOB error



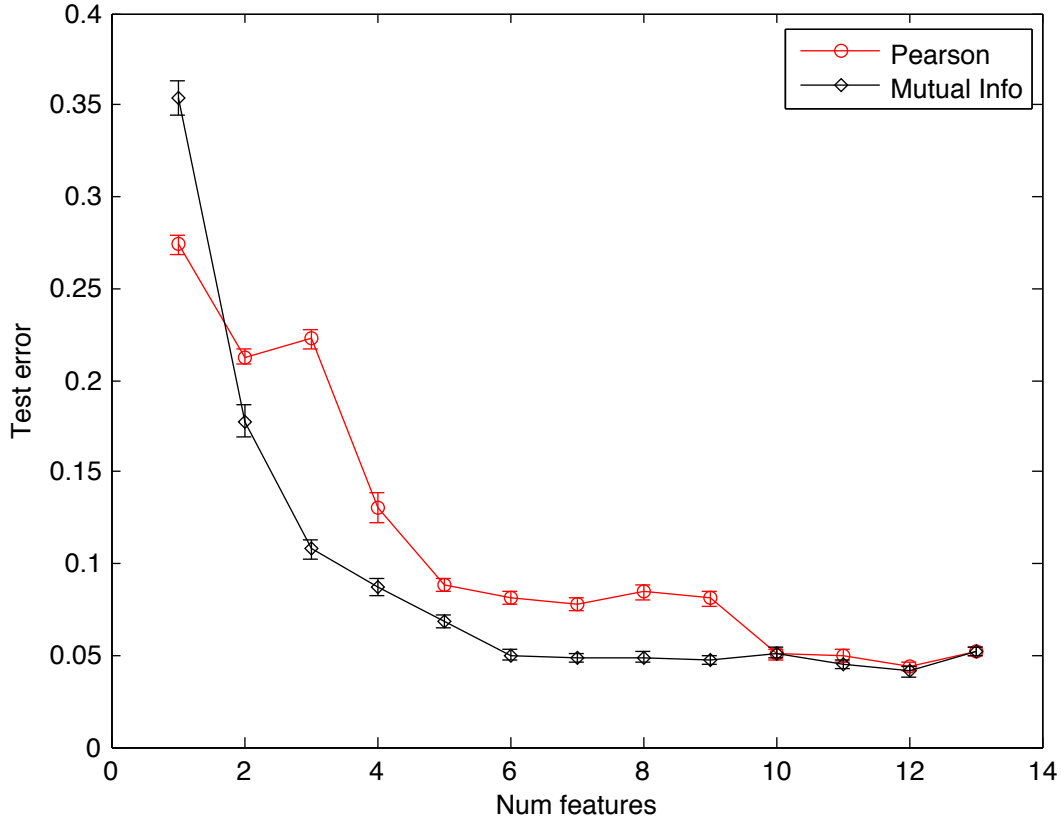
Pearson versus Mutual Info

Discretizing into 4 bins....



Pearson versus Mutual Info

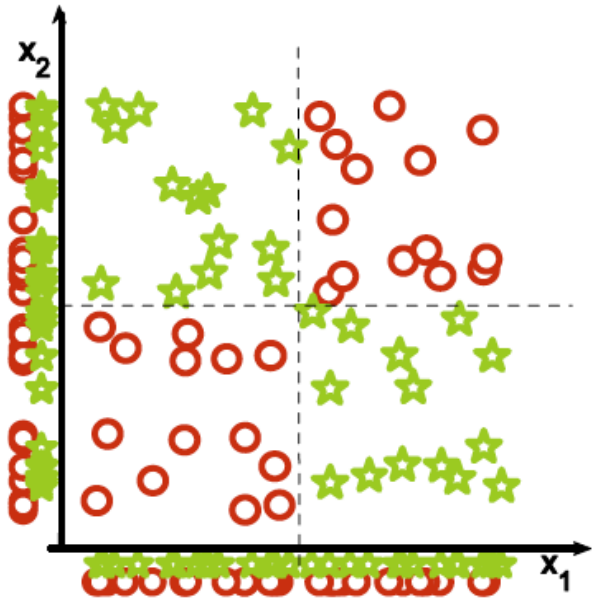
Wine data, 2 bins.



Performance is very much data-dependent.

Things to Remember

Features can be individually **irrelevant**,
and only useful when combined with others



Filter Ranking using Mutual Information

Rank features $X_k, \forall k$ by their values of $J = I(X_k; Y)$.
Retain the highest ranked features, discard the lowest ranked.

i	$J(X_k)$
35	0.846
42	0.811
10	0.810
654	0.611
22	0.443
59	0.388
...	...
212	0.09
39	0.05

Cut-off point decided by user, e.g. $|S| = 5$, so $S = \{35, 42, 10, 654, 22\}$.

but... what if I tell you features 42 and 10 are almost identical?!

Multi-variate Filters with Mutual Information

Problem: Highly correlated features are no use!

We need **relevant** features, but not **redundant** features.

One solution: Penalize correlations.

$$J_{mifs}(X_k) \quad = \quad I(X_k;Y) - \sum_{X_j \in S} I(X_k;X_j)$$

(Batitti, IEEE TNN 1994)

Forward/Backward with Filter Criteria

Full feature set $\Omega = \{X_1, .., X_M\}$.

Forward Selection:

10. $S = \emptyset$
20. $X_k = \arg\max_{X_k \in \Omega} J(X_k)$
30. $S \leftarrow S \cup X_k$.
40. $\Omega \leftarrow \Omega \backslash X_k$.
50. `Goto 20.`

Backward Elimination:

10. $S = \Omega$
20. $X_k = \arg\min_{X_k \in S} J(X_k)$
30. $S \leftarrow S \backslash X_k$.
40. $\Omega \leftarrow \Omega \cup X_k$.
50. `Goto 20.`

Lots of Filters using Mutual Information!

Max Relevance Min Redundancy (MRMR)
Peng et al, IEEE PAMI 2005.

$$J_{mrmr}(X_k) \quad = \quad I(X_k; Y) - \frac{1}{|S|} \sum_{X_i \in S} I(X_k; X_i)$$

Penalizes intra-feature correlations, reducing redundancy.

Averages over S, bringing relevancy/redundancy terms
to same order of magnitude.

Joint Mutual Information

Another solution: 'Joint Mutual Information'
(Yang & Moody, NIPS 1999)

$$J_{jmi}(X_k) \quad = \quad \sum_{X_i \in S} I(X_k X_i; Y)$$

"How useful is X_k when paired with each of the existing features?"

Re-examining the problem

These solutions seem sensible.

The conditional mutual information is $I(X;Y|S)$.
Measures the amount of shared info X, Y when we know S.

Seems sensible too...?

$$J_{cmi}(X_k) = I(X_k;Y|S)$$

As S grows, this is harder and harder to estimate.

$$I(X;Y|S) = \sum_{s\in S} p(s) \sum_{x\in X_k} \sum_{y\in Y} p(xy|s) \log \frac{p(xy|s)}{p(x|s)p(y|s)}$$

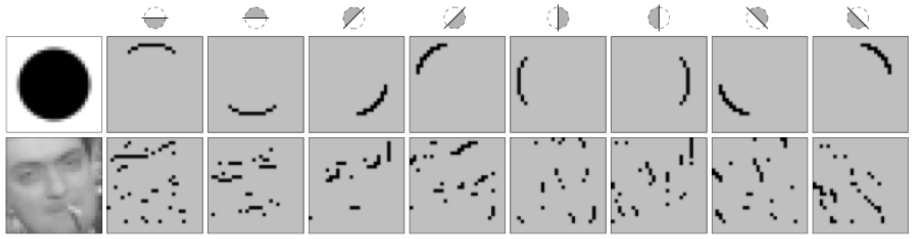
More challenging with higher arity features.
So we need low-dimensional approximations!

Conditional Mutual Information

If you want serious speed....

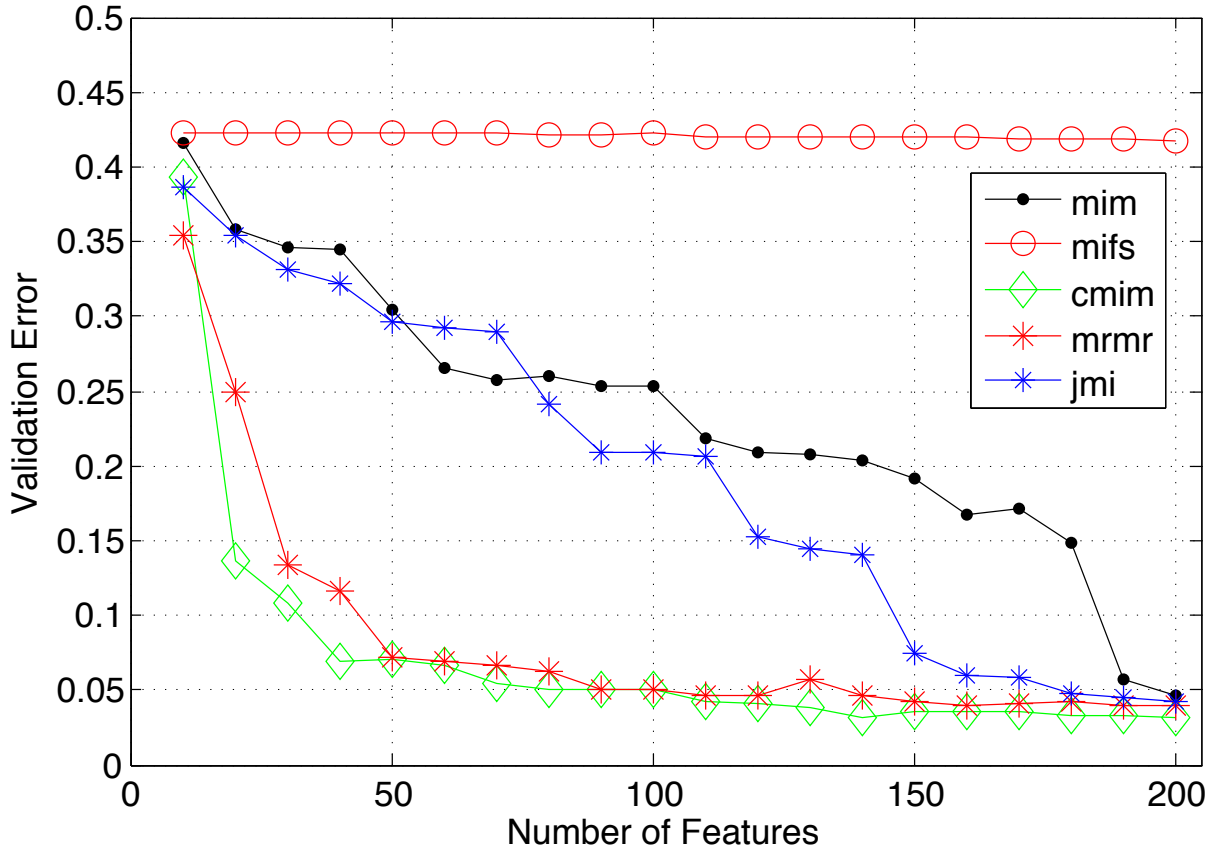
$$J_{cmim}(X_k) = \min_{j \in S} \left\{ I(X_k; Y | X_j) \right\}$$

F.Fleuret, "Fast Binary Feature Selection using Conditional Mutual Information", Journal of Machine Learning Research, vol 5, 2004.



"The implementation we propose selects 50 features among 40,000, based on a training set of 500 examples in $\frac{1}{10}$ of a second on a 1Ghz PC."

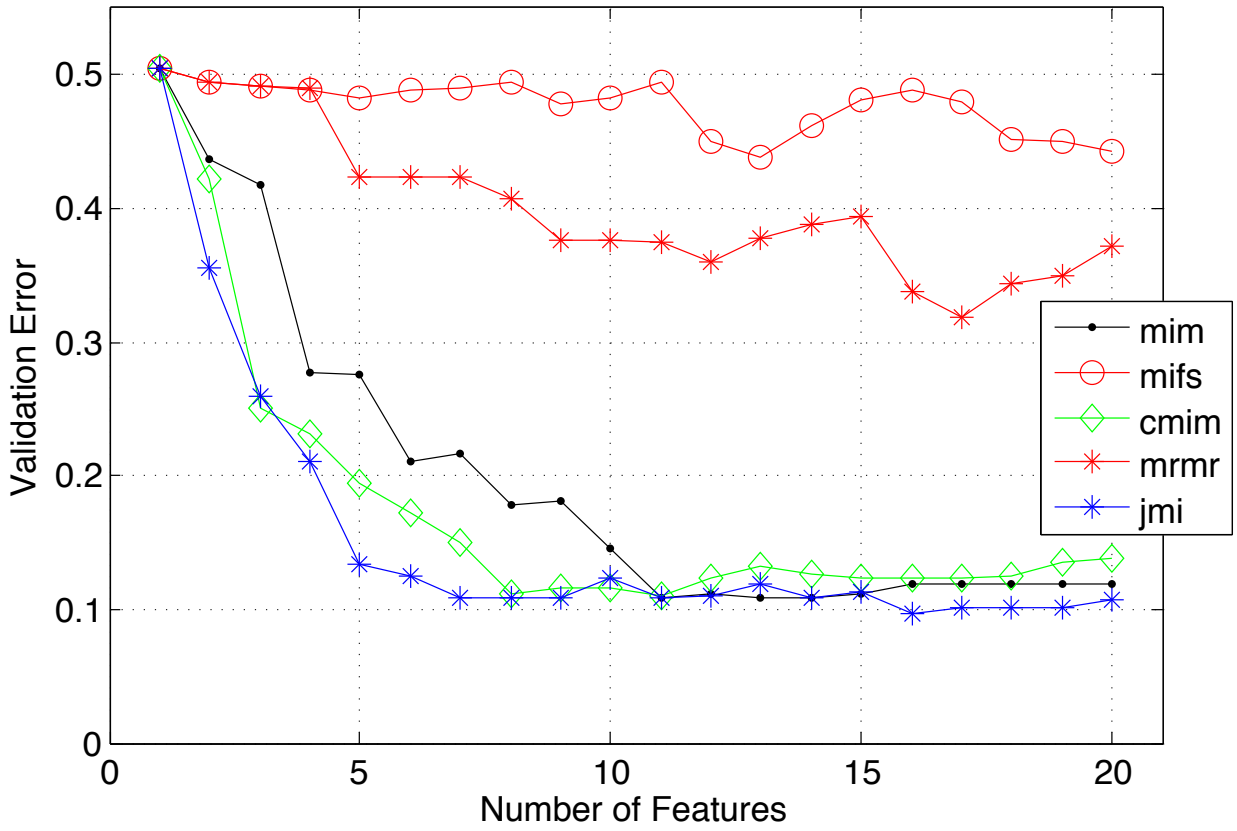
GISETTE data (handwriting recognition)



CMIM and MRMR do well.

MIFS fails, while JMI and MIM do reasonably well, eventually.

MADELON data (artificial data)



CMIM, JMI do well - and surprisingly, so does MIM.
MIFS fails, as does MRMR - why might that be...?

Re-examining the problem, again...

Use the 'chain' rule of mutual information and you get:

$$I(X_k; Y|S) = I(X_k; Y) - I(X_k; S) + I(X_k; S|Y)$$

Now terms have meaning.

First term $I(X_k; Y)$ is relevancy of X_k.
Second term $I(X_k; S)$ is redundancy of X_k against S.
Third term $I(X_k; S|Y)$ is conditional redundancy.

Ensemble methods... Redundancy $\approx$ Diversity?

G.Brown, "Some Thoughts at the Interface of Feature Selection and Ensemble Methods", Intl Workshop Multiple Classifier Systems 2009

Conclusions

There are lots of filter criteria using mutual information

Not really clear which is best, or when to use them.

- All assume various degrees of dependence.
- Some are difficult to estimate.
- Some are fast!

More on this in the final session of this course....