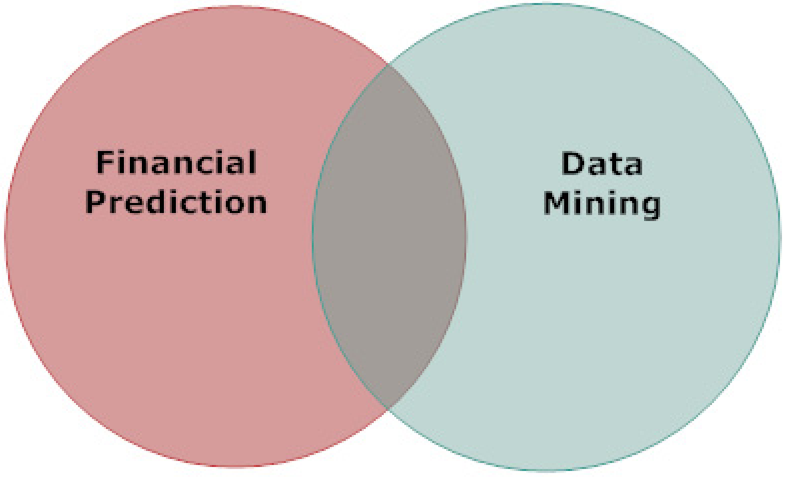
Financial data mining III
Price movement prediction in Hong Kong equity market

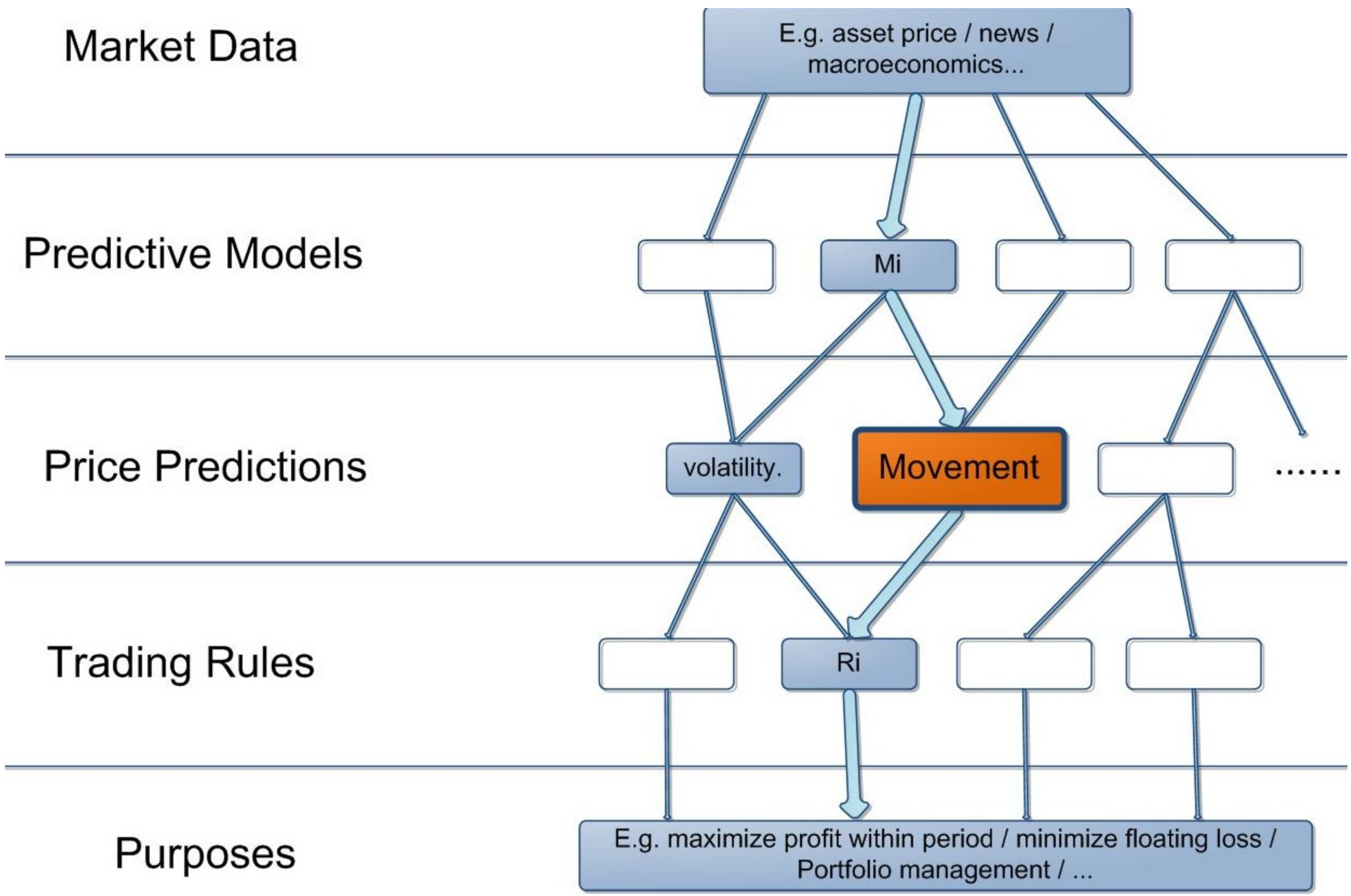
GROUP LCW1004
Ying Ting Chung
Au Chun Man

Review

- To learn patterns from historical data
  - using data mining
  - for forecasting future price movement
- HSI (Hang Seng Index), daily prices

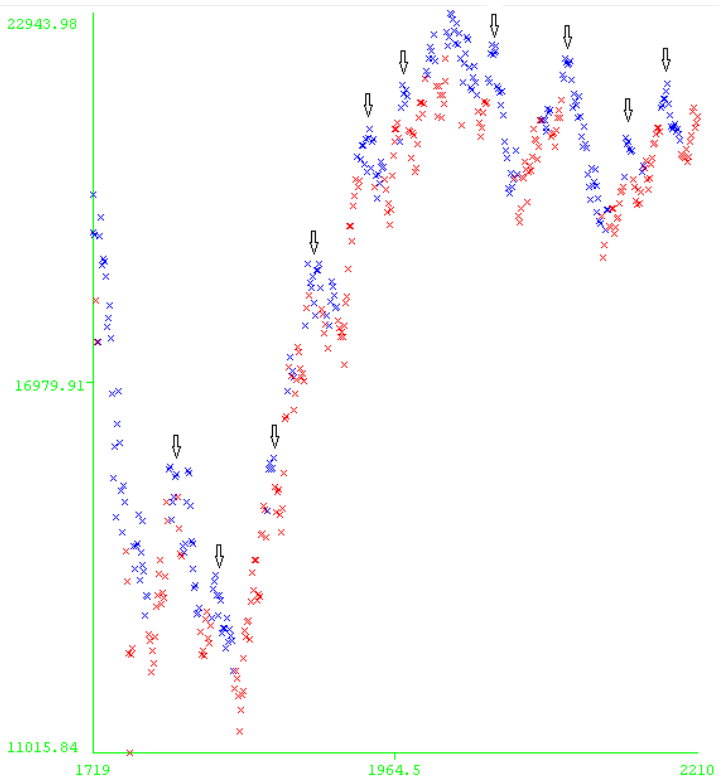


Prediction Problem



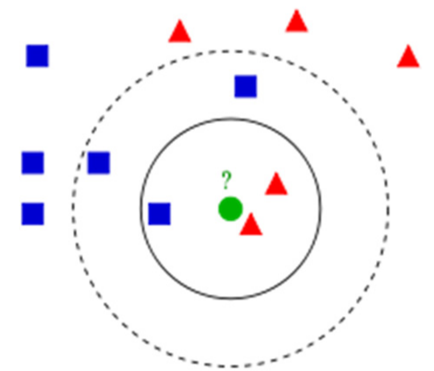
Semester 1

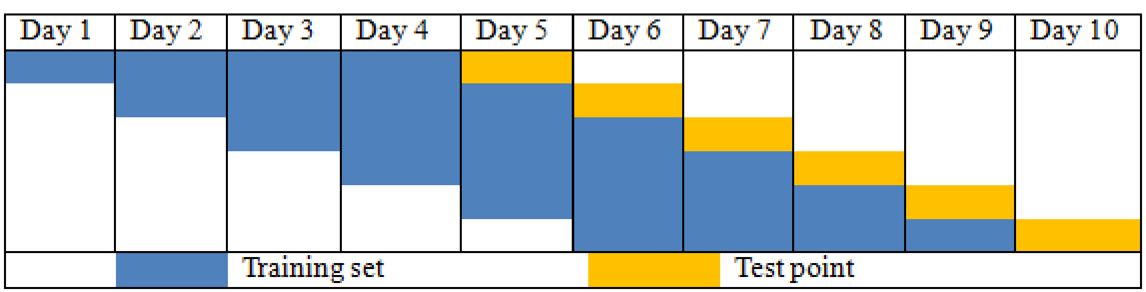
- Finished data collection and pre-processing
- Binary classification
  - Target label: “8-day bottom”



Semester 1

- K-Nearest Neighbour (k-NN)
  - Low bias
- Sliding window testing
  - Model changes through time





Data Visualization

- Label against various input attribute
- Label distribution through time
- Input Attributes:

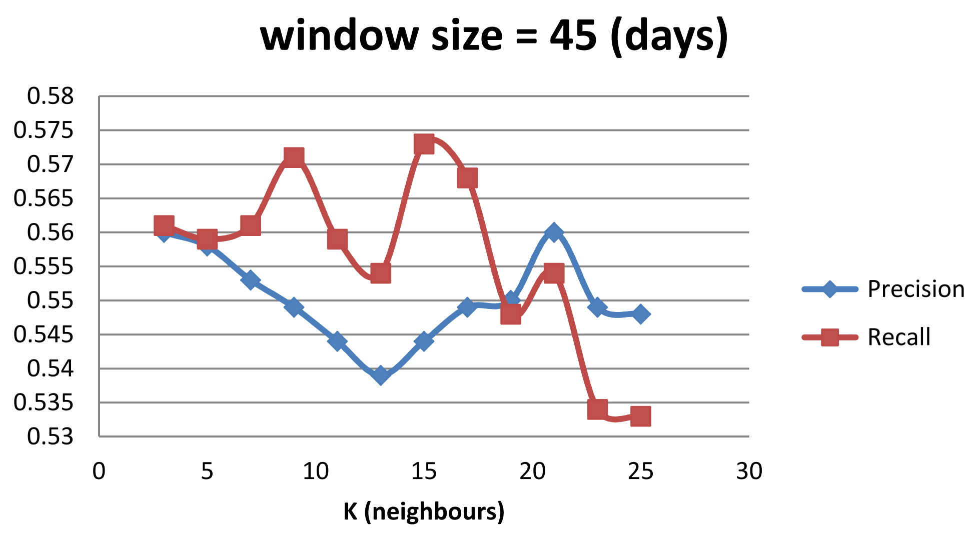
Slope of SMA(22) and EMA(8, 13, 18)
Price Deviation from SMA(9, 22) and EMA(8, 16)
Percentage aggregate return for last 5 and 6 days
Daily gap open

- Prediction Target: 5-day bottom

Data Visualization

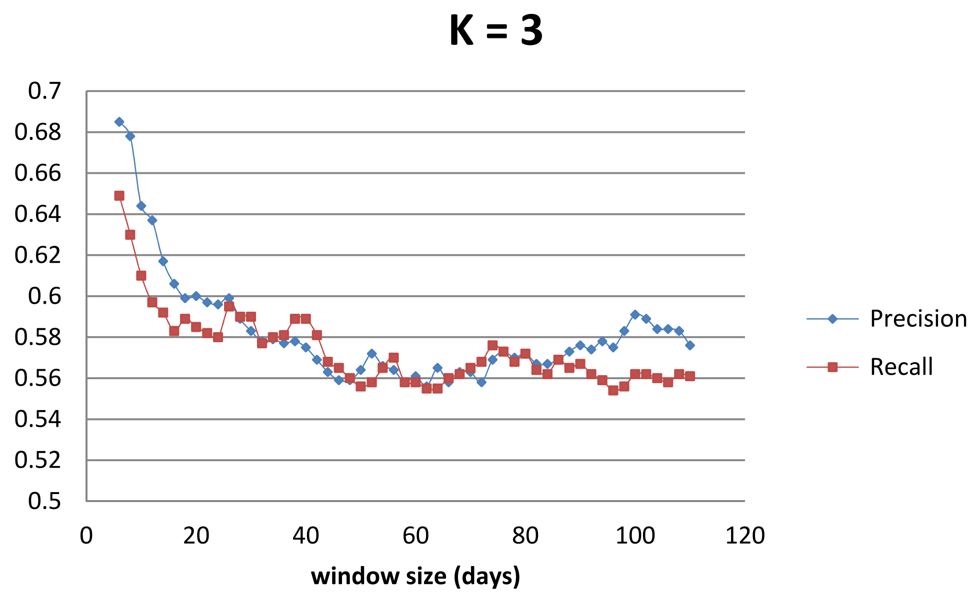
Parameter Estimation

- Selecting no. of voting neighbours



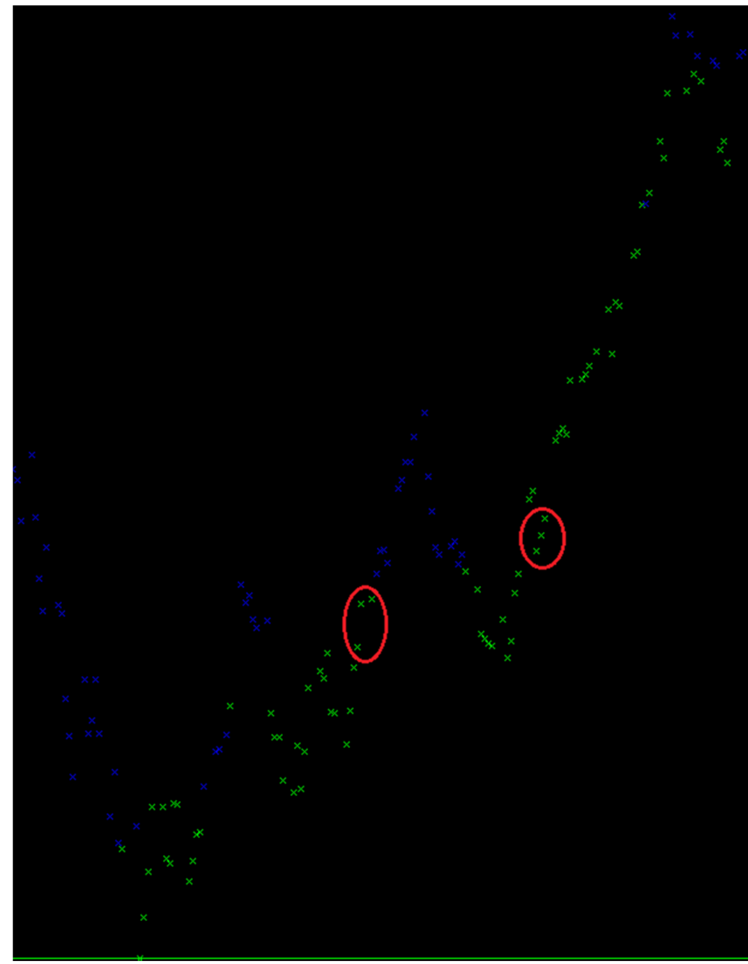
Parameter Estimation

- Selecting no. of days looking back
  - Window size reduced, performance improved?



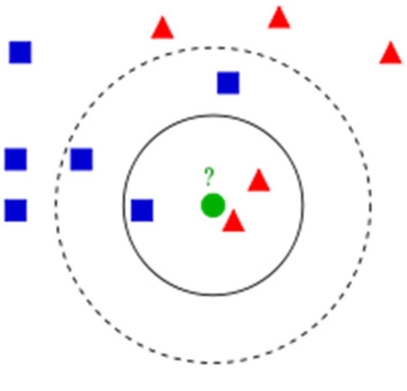
Problem with fixed sliding window

- Given 3-NN majority voting
- Property of target label
  - Continually positive
- Fail to learn
  - From history
  - From input variable

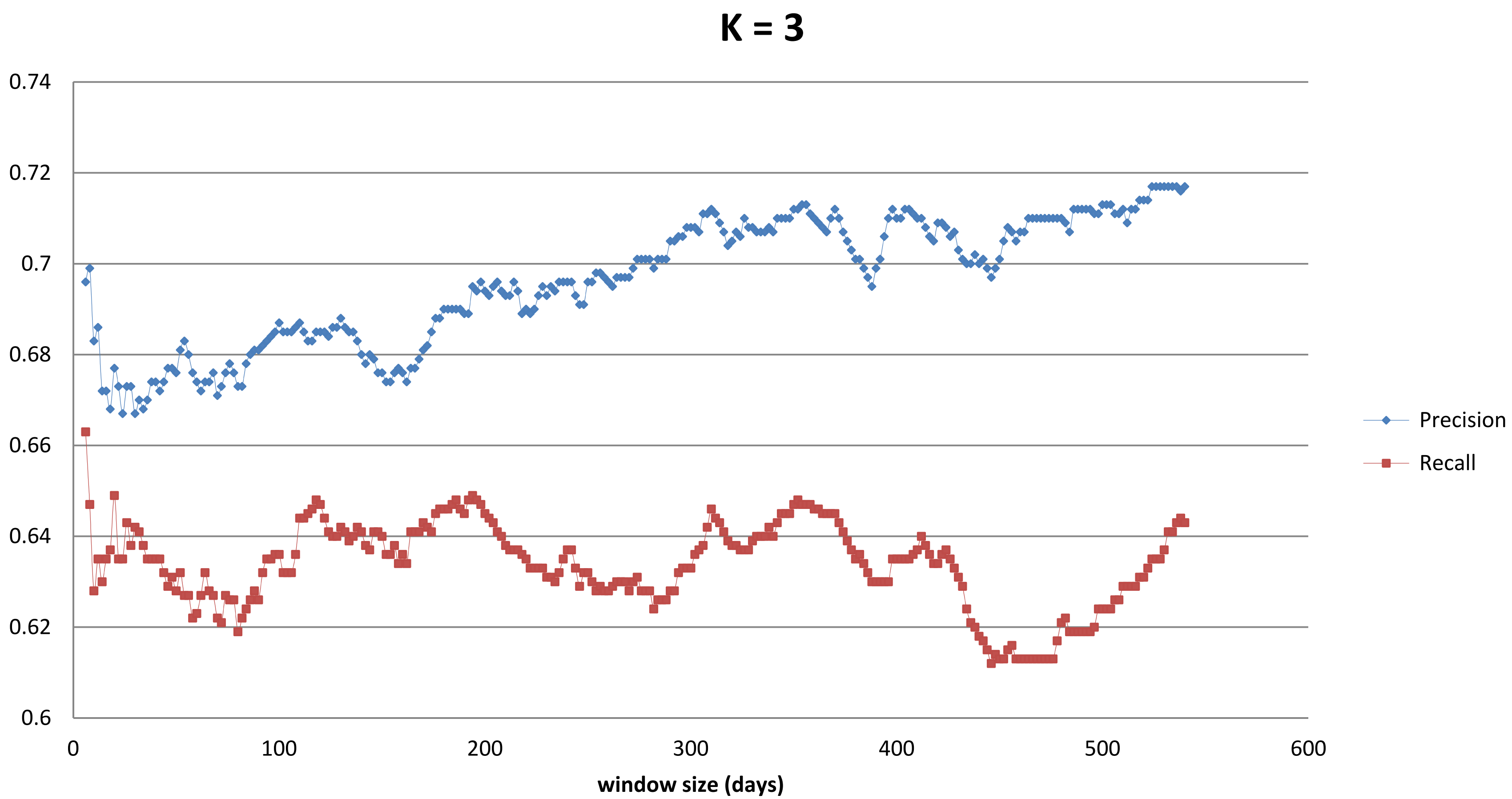


Function of time as attribute

- Recent labels are more important in trend-dominant market
- New dimension in feature space:
  Exponential function of time index
- Recent data vectors pulled closer
  to current query point
- i.e. Memory decay
- controversial



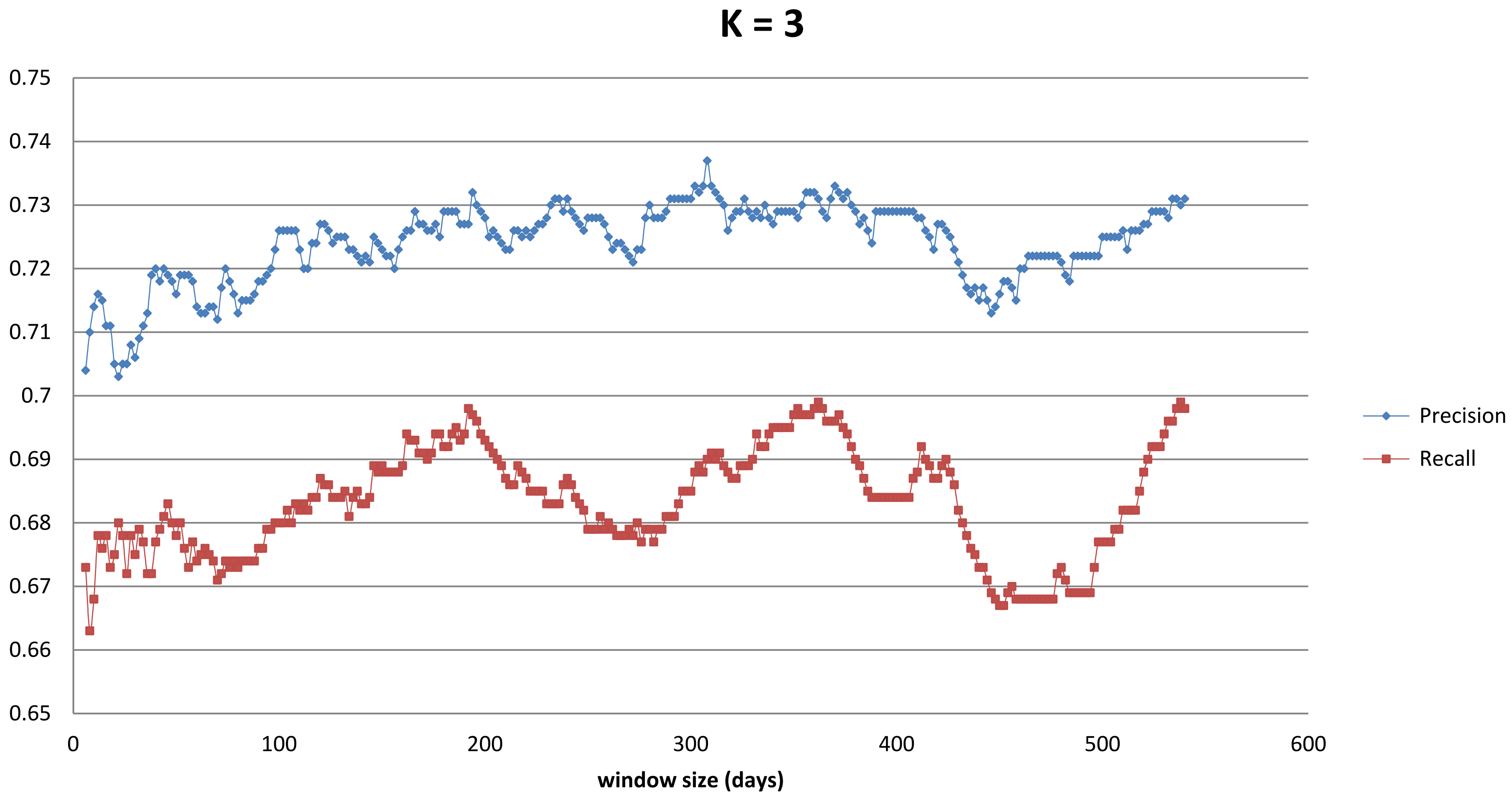
Function of time as attribute



Function of time as attribute

- New dimension in feature space:
  Sine function of time index
- Data vectors at regular interval pulled closer to each other
- Choice of period?

Function of time as attribute



Result

- Looking back: 540 days
- K = 3 neighbours

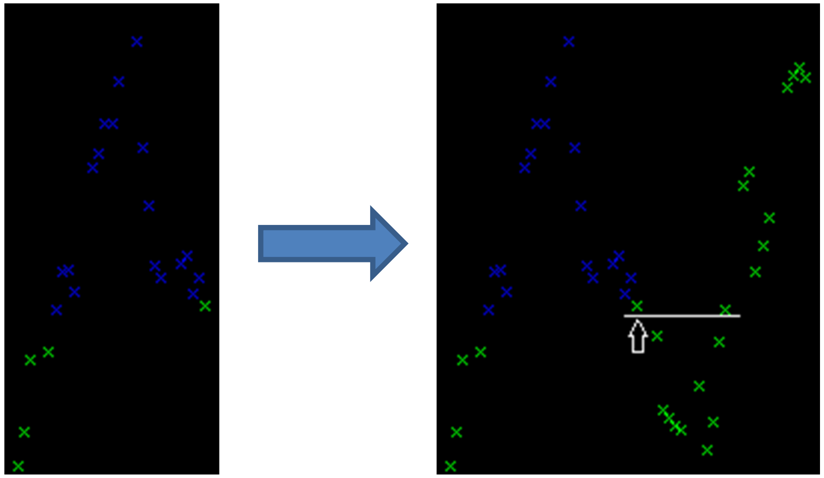
```
=== Confusion Matrix ===

   a   b   c     actual class
 324   0  77 |   a = -1
   1   2   2 |   b = 0
  93   0 215 |   c = 1
```

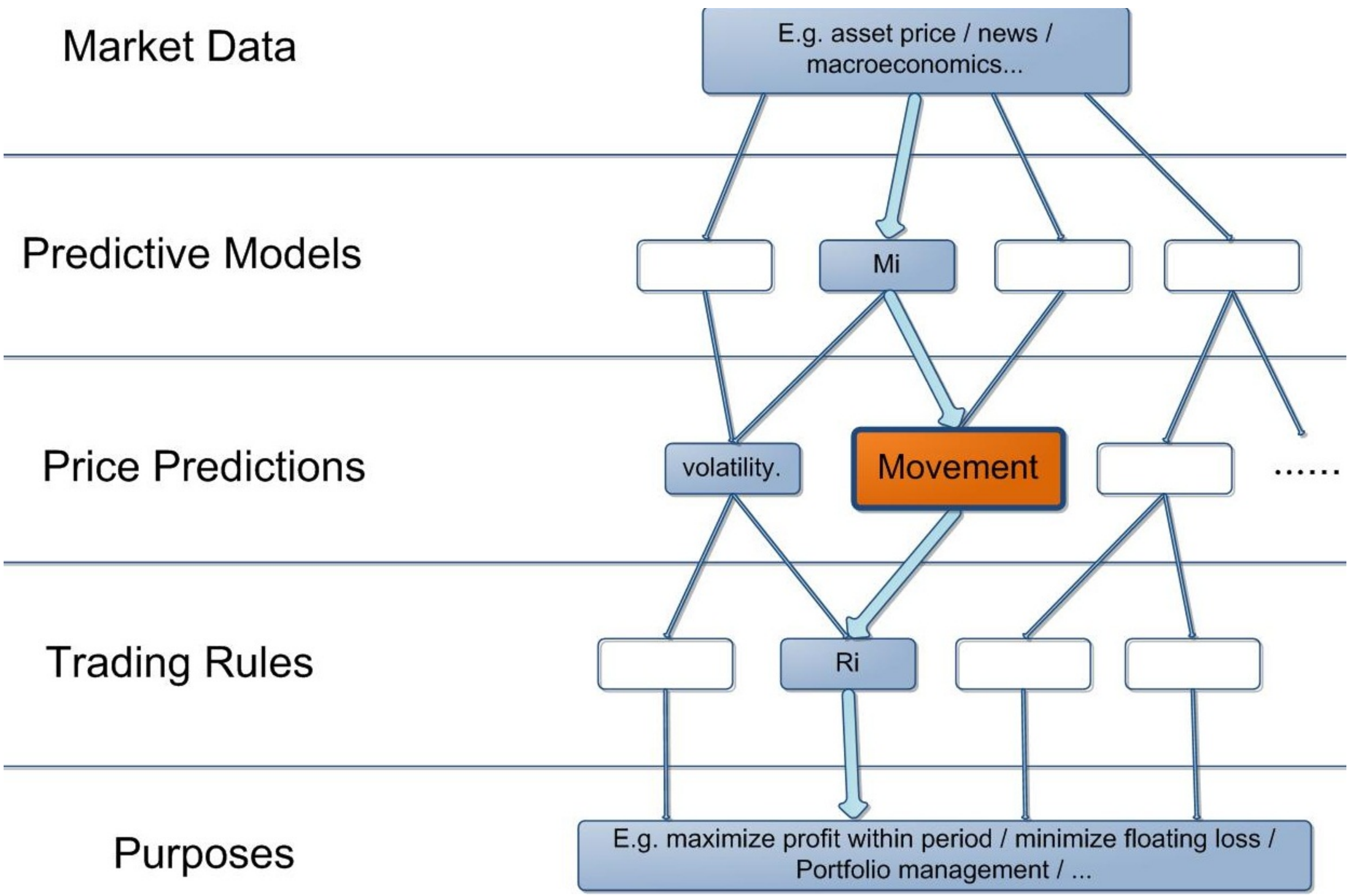
- Error rate: 24%
- Precision: 0.731; Recall: 0.698

Limitation of label definition

- Consider a use case:
  - A trader buys on first occurrence of positive prediction?
  - He needs a smart trading rule



Prediction Problem



Re-consider label definition

- N-bottom + r% gain after N days
  - *For each time point t, if its closing price is lower than the low price by m% after N time points, then time point t is a hold-N-bottom.*

- Trading signal

Re-consider label definition

- N-bottom + m% stop-loss within N days
  - *For each time point t, if its closing price is lower than the minimum low price within N time points in the future by m%, then time point t is a stop-N-bottom.*

- More specific, more useful
- Fewer positive labels -> imbalanced dataset

New Result

- Looking back: 120 days
- K = 3 neighbours

```
=== Confusion Matrix ===

   a   b   c     actual class
 791   0  79 |   a = -1
   2  12   0 |   b = 0
 131   0 119 |   c = 1
```

- Error rate: 19.6%
- Precision: 0.601; Recall: 0.476

Artificial Neural Networks

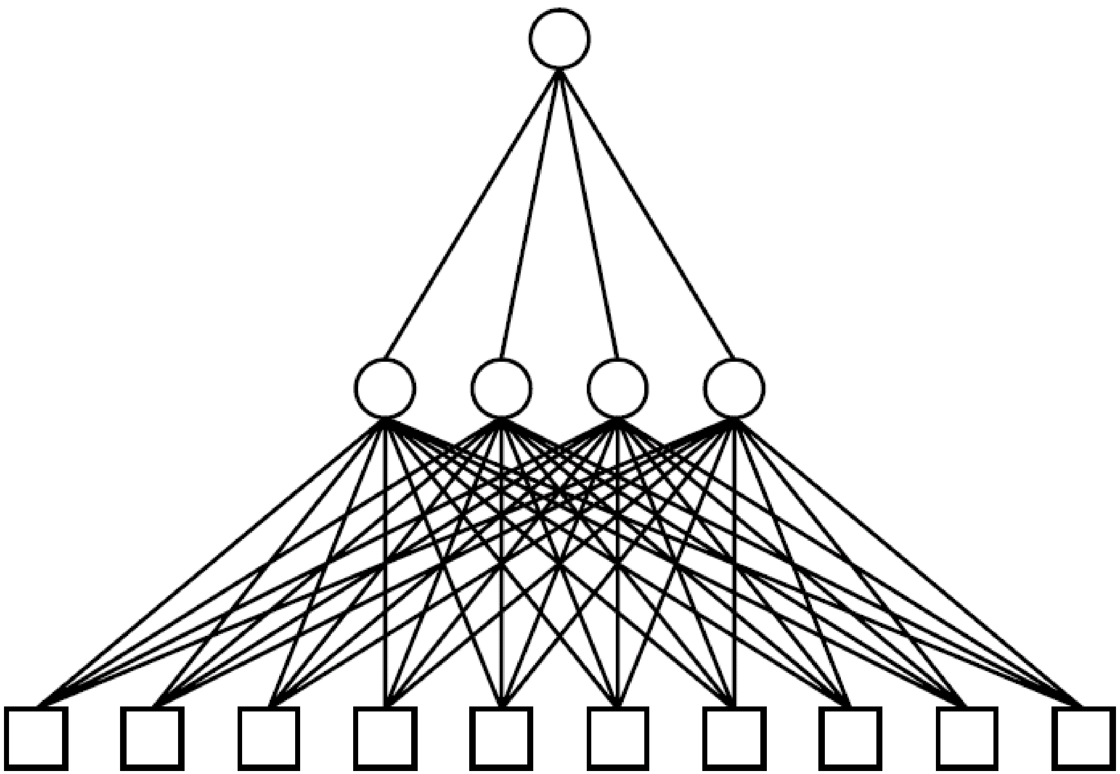
- Theory
- Input and Output Selection
- Experiment

Artificial Neural Networks - Theory

Output units y_p

Hidden units h_m

Input units x_n



Back Propagation Algorithm

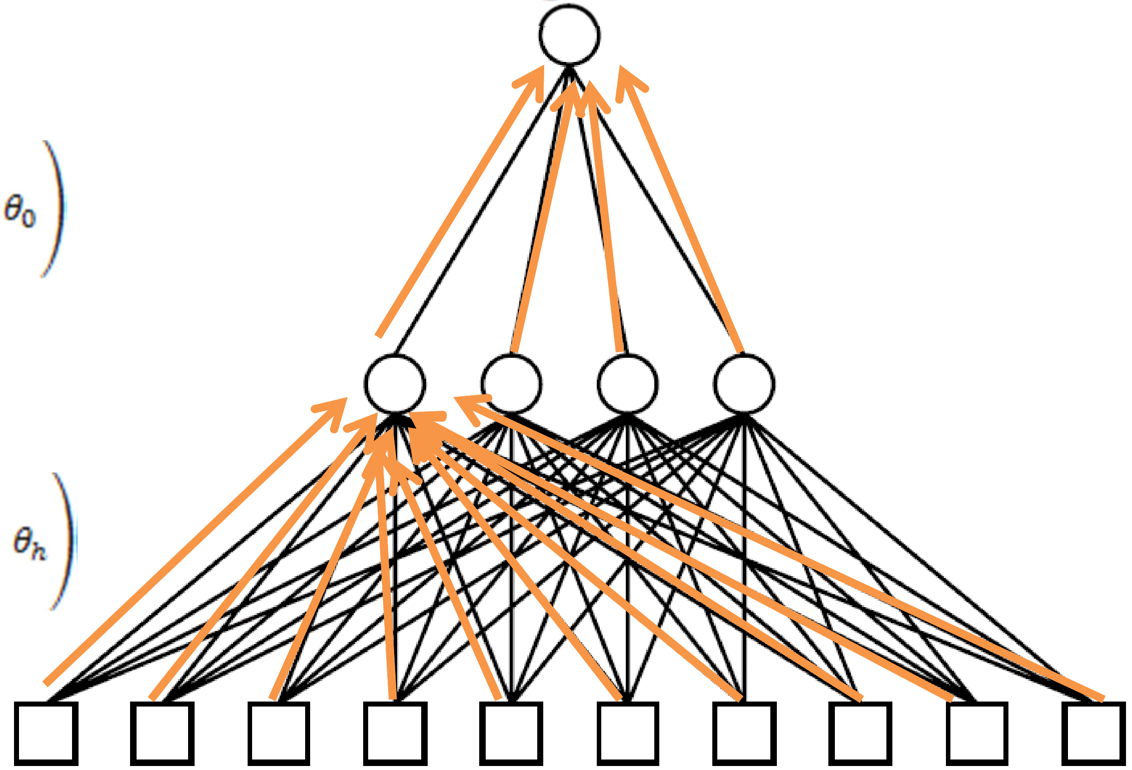
- Step 1 : forward pass
- Step 2 : backward pass
- Step 3 : weight updates

Back Propagation Algorithm – Step 1 :forward pass

- set of input vectors: $(x_1, x_2, \dots, x_p)$
- set of output vectors: $(y_1, y_2, \dots, y_p)$

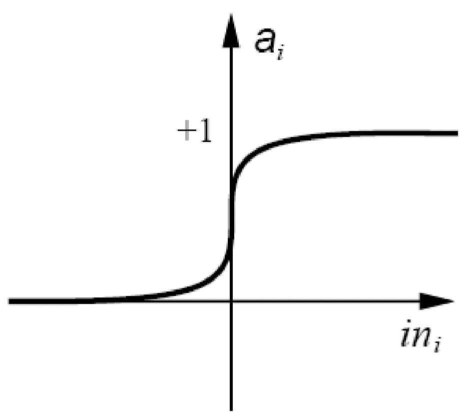
$$y_p = f\left(\sum_p w_{kj} h_p + \theta_0\right)$$

$$h_p = f\left(\sum_p w_{ji} x_p + \theta_h\right)$$



Back Propagation Algorithm – Step 1 :forward pass

- *f(.)* is a sigmoid activation function



$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

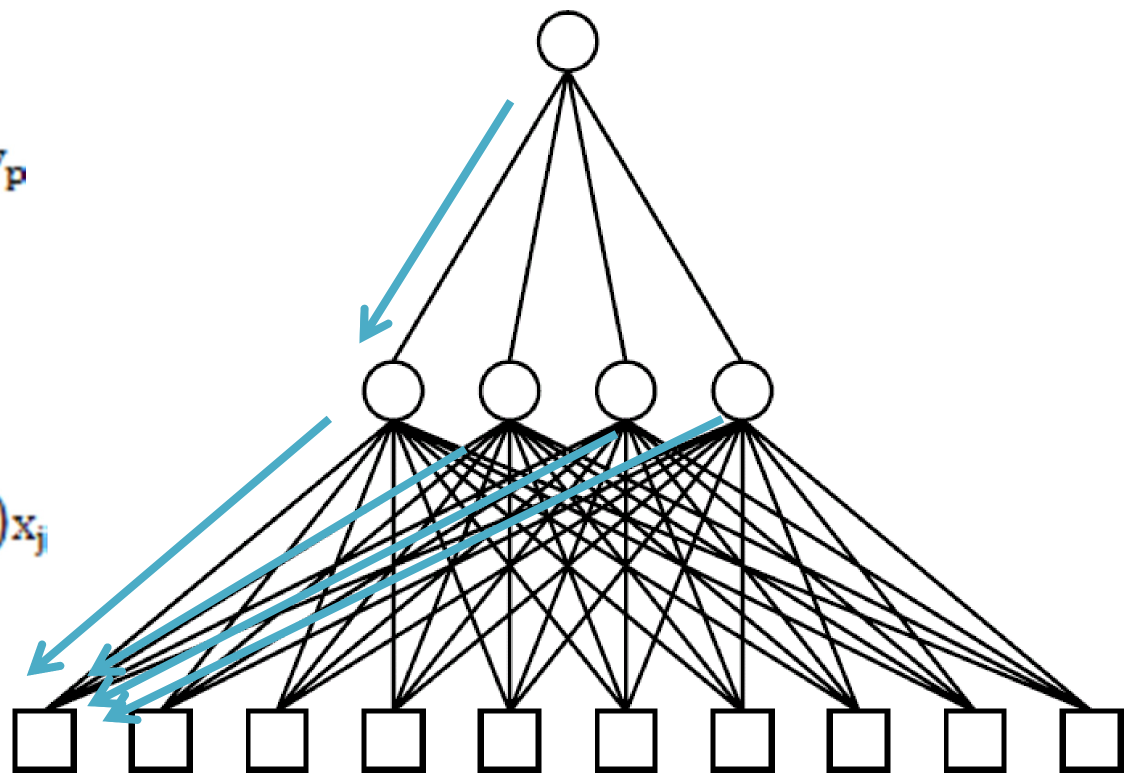
(c) Sigmoid function

Back Propagation Algorithm – Step 2 :backward pass

- Sum of squared errors:

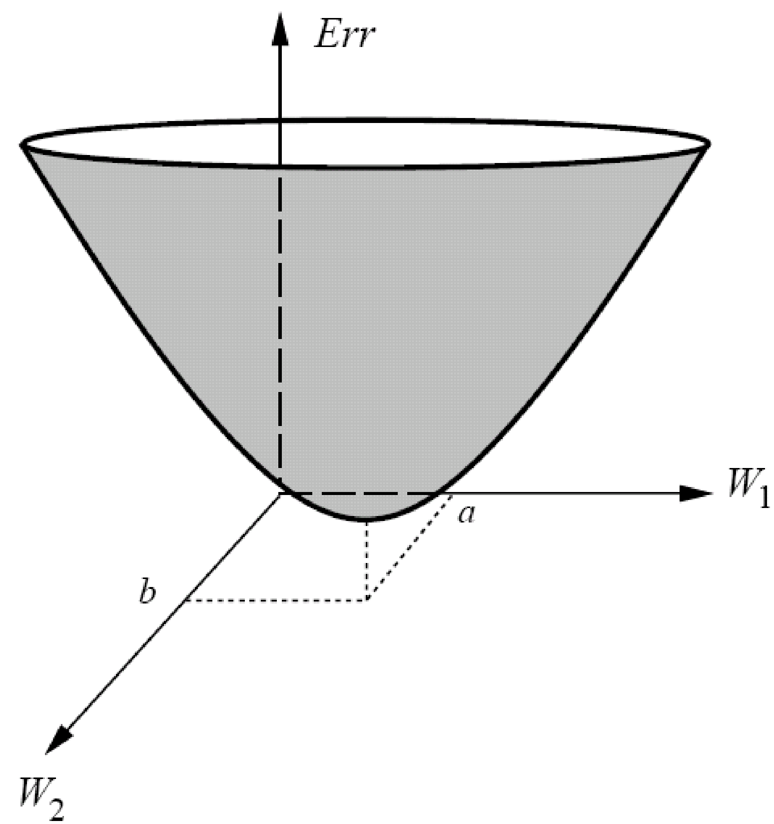
$$E_p = \sum (t_p - y_p)(1 - y_p)y_p$$

$$E_j = \left(\sum_p w_{kj} E_p\right)(1 - x_j)x_j$$

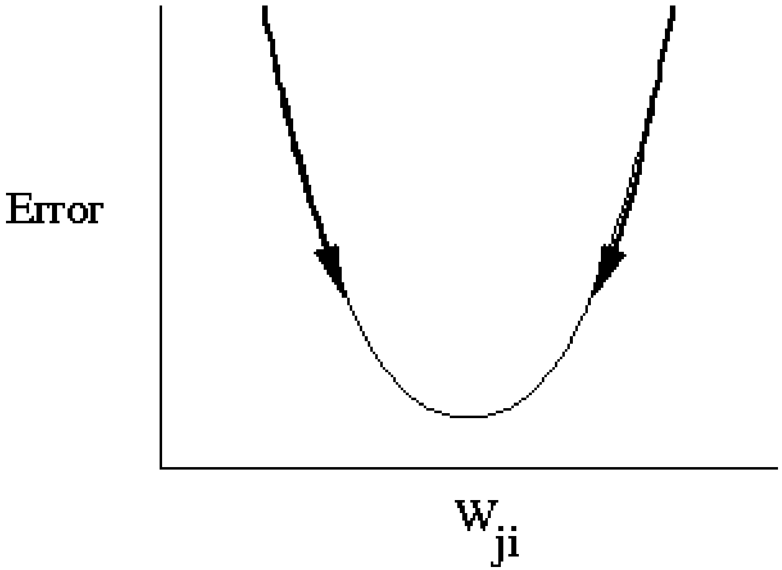


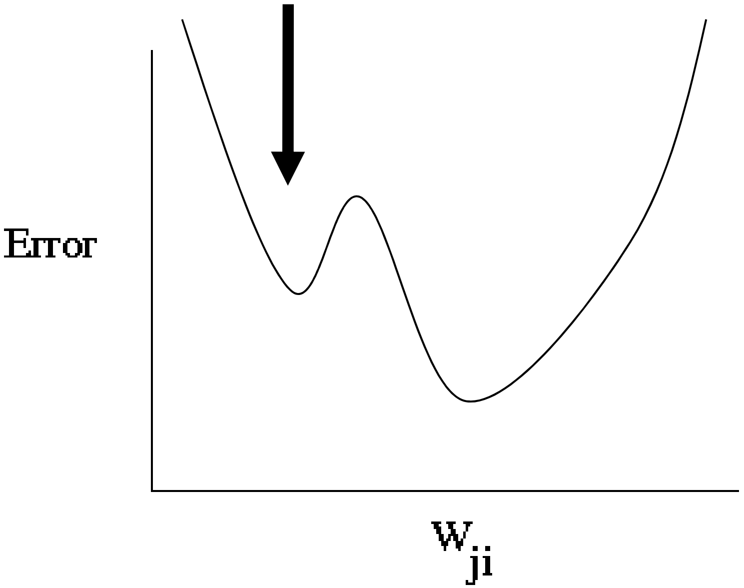
Gradient Descent

- To minimize E
- partial derivation $\frac{\partial E}{\partial W_i}$



Gradient Descent – local minimum

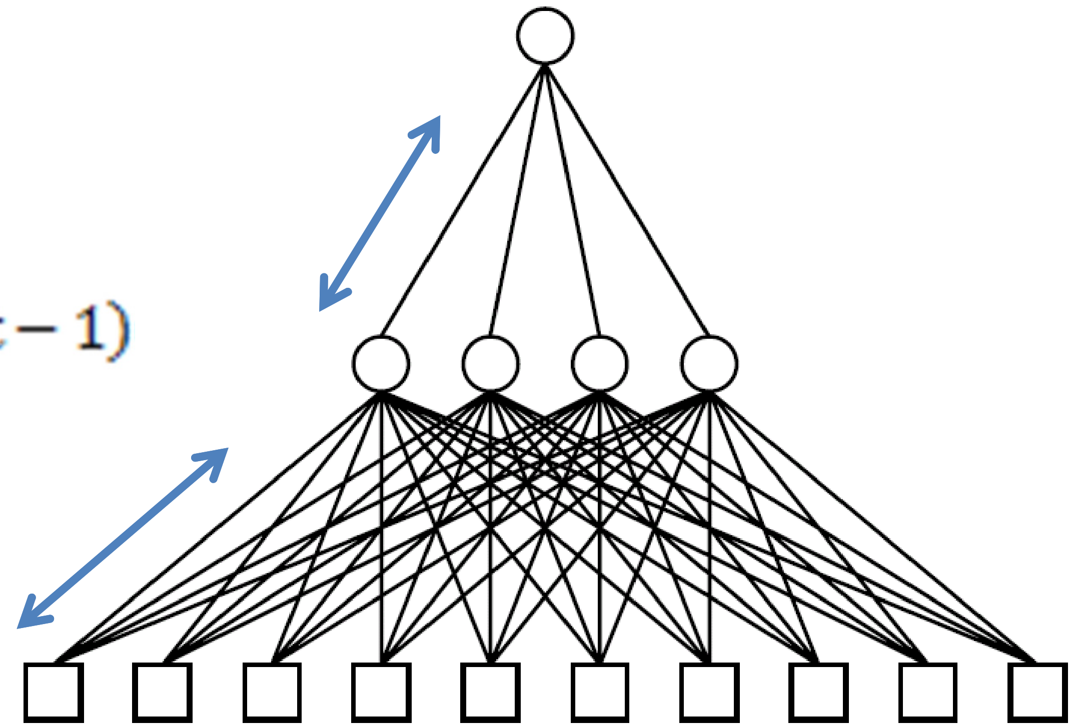




Step 3 : weight updates

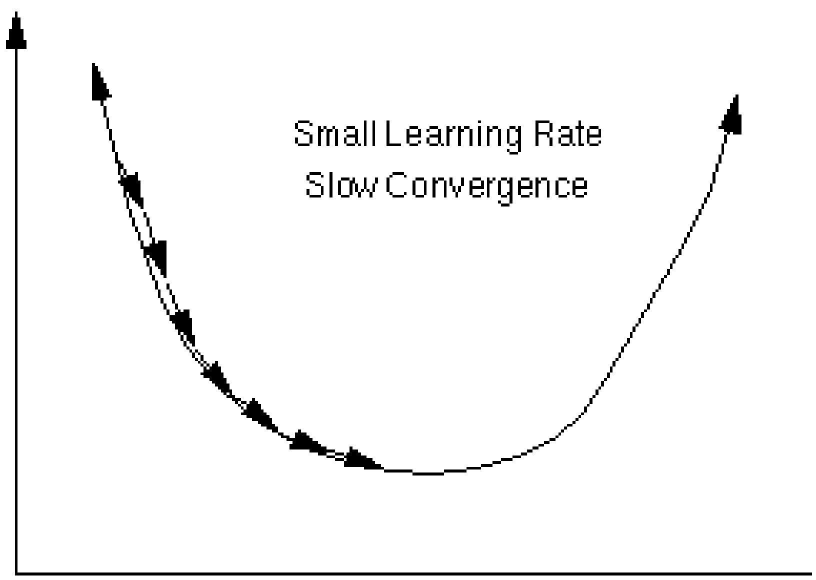
- Weight updating:

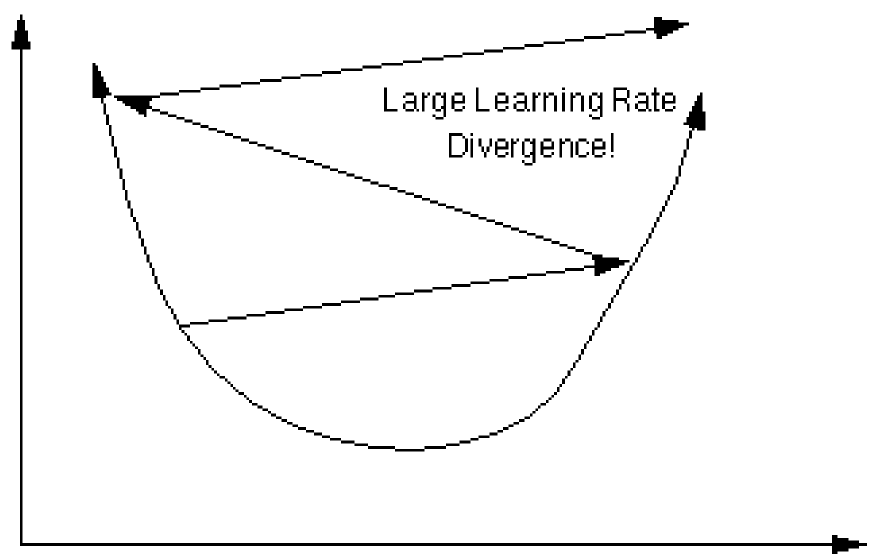
$$\Delta w(t) = -\eta \frac{\partial E}{\partial w} + \alpha \Delta w(t-1)$$



- learning rate : η
- momentum factor : α

Learning Rate & Momentum





Limitations

- requires lots of supervised training, with lots of input-output examples
- there is no guarantee that the system will converge to an acceptable solution

  ~local minimum

Experiment

- The learning rate :0.25 to 0.5 ,and
- Momentum set to WEKA’s default:0.2
- the size of training set: 500 to 400000.

Experiment

- Confusion matrix

		Predicted	
		Positive	*Negative*
True	*Positive*	*TP*	*FN*
	Negative	*FP*	*TN*

Experiment - Result

Size of training set	500	5000	10000	20000	30000	400000
Learning rate = 0.25	$\begin{bmatrix}399 & 69\\111 & 44\end{bmatrix}$	$\begin{bmatrix}402 & 66\\109 & 43\end{bmatrix}$	$\begin{bmatrix}400 & 68\\107 & 45\end{bmatrix}$	$\begin{bmatrix}400 & 68\\110 & 42\end{bmatrix}$	$\begin{bmatrix}405 & 63\\103 & 38\end{bmatrix}$	N/A
Learning rate = 0.3	$\begin{bmatrix}445 & 23\\148 & 4\end{bmatrix}$	$\begin{bmatrix}402 & 66\\106 & 46\end{bmatrix}$	$\begin{bmatrix}401 & 67\\106 & 46\end{bmatrix}$	$\begin{bmatrix}399 & 69\\106 & 46\end{bmatrix}$	$\begin{bmatrix}392 & 76\\106 & 46\end{bmatrix}$	$\begin{bmatrix}392 & 76\\112 & 40\end{bmatrix}$
Learning rate = 0.35	$\begin{bmatrix}435 & 33\\145 & 7\end{bmatrix}$	$\begin{bmatrix}393 & 75\\109 & 43\end{bmatrix}$	$\begin{bmatrix}402 & 66\\114 & 38\end{bmatrix}$	$\begin{bmatrix}402 & 66\\113 & 39\end{bmatrix}$	$\begin{bmatrix}404 & 64\\115 & 37\end{bmatrix}$	$\begin{bmatrix}402 & 66\\114 & 38\end{bmatrix}$
Learning rate = 0.5	$\begin{bmatrix}393 & 73\\129 & 23\end{bmatrix}$	$\begin{bmatrix}351 & 117\\91 & 61\end{bmatrix}$	$\begin{bmatrix}349 & 119\\91 & 61\end{bmatrix}$	$\begin{bmatrix}356 & 112\\92 & 60\end{bmatrix}$	$\begin{bmatrix}360 & 108\\94 & 58\end{bmatrix}$	N/A

Result analysis

		Predicted	
		Positive	*Negative*
True	*Positive*	*TP*	*FN*
	Negative	*FP*	*TN*

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

Result analysis

	RSI(14)	Accuracy	Precision
4 day bottom	$\begin{bmatrix} 577 & 451 \\ 372 & 316 \end{bmatrix}$	0.5209627	0.6080084
8 day bottom	$\begin{bmatrix} 438 & 404 \\ 405 & 465 \end{bmatrix}$	0.4274533	0.5185729
12 day bottom	$\begin{bmatrix} 451 & 431 \\ 379 & 447 \end{bmatrix}$	0.5257611	0.54337349
16 day bottom	$\begin{bmatrix} 355 & 420 \\ 413 & 516 \end{bmatrix}$	0.5111502	0.46223958
20 day bottom	$\begin{bmatrix} 314 & 379 \\ 423 & 564 \end{bmatrix}$	0.5226190	0.42605156

Size of training set	Learning rate = 0.35	Accuracy	Precision
500	$\begin{bmatrix} 435 & 33 \\ 145 & 7 \end{bmatrix}$	0.716129032	0.75
5000	$\begin{bmatrix} 393 & 75 \\ 109 & 43 \end{bmatrix}$	0.754838709	0.782868525
10000	$\begin{bmatrix} 402 & 66 \\ 114 & 38 \end{bmatrix}$	0.709677419	0.779068767
20000	$\begin{bmatrix} 402 & 66 \\ 113 & 39 \end{bmatrix}$	0.711290322	0.780582524
30000	$\begin{bmatrix} 404 & 64 \\ 115 & 37 \end{bmatrix}$	0/711290322	0.778420038
40000	$\begin{bmatrix} 402 & 66 \\ 114 & 38 \end{bmatrix}$	0.709677419	0.779069767

Result analysis - Problem?

- False Positive

$$\text{Special FP rate} = \frac{FP}{FP+TN+TP+FN}$$

		Predicted	
		Positive	*Negative*
True	*Positive*	*TP*	*FN*
	Negative	*FP*	*TN*

False Positive

	RSI(14)	Special FP rate
4 day bottom	$\begin{bmatrix} 577 & 451 \\ 372 & 316 \end{bmatrix}$	0.216783216
8 day bottom	$\begin{bmatrix} 438 & 404 \\ 405 & 465 \end{bmatrix}$	0.23656542
12 day bottom	$\begin{bmatrix} 451 & 431 \\ 379 & 447 \end{bmatrix}$	0.221896955
16 day bottom	$\begin{bmatrix} 355 & 420 \\ 413 & 516 \end{bmatrix}$	0.242370892
20 day bottom	$\begin{bmatrix} 314 & 379 \\ 423 & 564 \end{bmatrix}$	0.251785714

Size of training set	Learning rate = 0.35	Special FP rate
500	$\begin{bmatrix} 435 & 33 \\ 145 & 7 \end{bmatrix}$	0.233870967
5000	$\begin{bmatrix} 393 & 75 \\ 109 & 43 \end{bmatrix}$	0.175806451
10000	$\begin{bmatrix} 402 & 66 \\ 114 & 38 \end{bmatrix}$	0.183870967
20000	$\begin{bmatrix} 402 & 66 \\ 113 & 39 \end{bmatrix}$	0.182258064
30000	$\begin{bmatrix} 404 & 64 \\ 115 & 37 \end{bmatrix}$	0.185483871

Discussion

- 100% accuracy is impossible in practice
- Effectiveness of classification model also depends on how it is used, i.e. coupled with trading rules

Further Improvement