1 K-NN and /decision Trees

Distances, $d(x^{(i)}, x^{(q)}) =$

Manhattan	L1	$\sum_{k=1}^{K} \lvert x_k^{(i)} - x_k^{(q)} \rvert$
Euclidean	L2	$\sqrt{\sum_{k=1}^{K} (x_k^{(i)} - x_k^{(q)})^2}$
Chebyshev	L∞	$\max_{k=1}^{K} \lvert x_k^{(i)} - x_k^{(q)} \rvert$

$$\text{Types} = \begin{array}{cc} \text{Inverse} & \frac{1}{d(x^{(i)}, x^{(q)})} \\ \text{Gaussian} & \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{d(x^{(i)}, x^{(q)})^2}{2}\right) \end{array}$$

$$\text{Entropy} = H(X) = \begin{array}{c} -\sum_k^K P(x_k) \log_2(P(x_k)) \\ -\int_k f(x) \log_2(f(x)) \end{array}$$

Information Gain $=$ IG(dataset, subsets)

$$= H(dataset) - \sum_{S \in \text{subsets}} \frac{|S|}{|\text{dataset}|} H(S)$$

$$|\text{dataset}| = \sum_{S \in \text{subsets}} |S|$$

2 Evaluation of Machine Learning Systems

Confusion Matrix

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	**TP** True Positive	**FN** False Negative
Class 2 Actual	**FP** False Positive	**TN** True Negative

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

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

Recall $= \frac{TP}{TP+FN}$

F-measure $= F_1 = 2\frac{Precision*Recall}{Precision+Recall}$

MSE $= \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$

Sample Error:

$error_s(h) = \frac{1}{N}\sum_{x \in S} \delta(f(x), h(x))$

Confidence interval:

$error_s(h) \pm Z_N \sqrt{\frac{error_s(h)*(1-error_s(h))}{n}}$

3 Neural Networks

Activation Functions:

Function	$g(z)$	$g'(z)$
Sigmoid	$\frac{1}{1+e^{-x}}$	$g(z)(1-g(z))$
Tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - g(z)^2$
ReLU	0 for $x \leq 0$ x for $x > 0$	0 for $x \leq 0$ 1 for $x > 0$
Softmax	$\frac{e^{Z_i}}{\sum_k e^{Z_k}}$	$\frac{\delta L}{\delta z} = \frac{1}{N}(\hat{y} - y)$

Back Propagation:

$$\frac{\delta \text{Loss}}{\delta W} = \frac{\delta \text{Loss}}{\delta Z} \cdot \frac{\delta Z}{\delta W}$$
$$\frac{\delta \text{Loss}}{\delta b} = \frac{\delta \text{Loss}}{\delta Z} \cdot \frac{\delta Z}{\delta b}$$
$$\frac{\delta \text{Loss}}{\delta X} = \frac{\delta \text{Loss}}{\delta Z} \cdot \frac{\delta Z}{\delta X}$$
$$\frac{\delta \text{Loss}}{\delta X} = \frac{\delta \text{Loss}}{\delta Z} \cdot W^T$$
$$\frac{\delta \text{Loss}}{\delta W} = X^T \cdot \frac{\delta \text{Loss}}{\delta Z}$$
$$\frac{\delta \text{Loss}}{\delta b} = 1^T \cdot \frac{\delta \text{Loss}}{\delta Z}$$
$$\frac{\delta \text{Loss}}{\delta Z} = \frac{\delta \text{Loss}}{\delta A} \circ g'(Z)$$

where $A = g(Z)$

Regularisation

$L1 : J(\theta) = Loss(y, \hat{y}) + \lambda \sum_w w^2$

$w \leftarrow w - \alpha(\frac{\delta Loss}{\delta w} + 2\lambda w)$

$L2 : J(\theta) = Loss(y, \hat{y}) + \lambda \sum_w |w|$

$w \leftarrow w - \alpha(\frac{\delta Loss}{\delta w} + 2 \sin w)$

Dropout: drop 50% of Activation

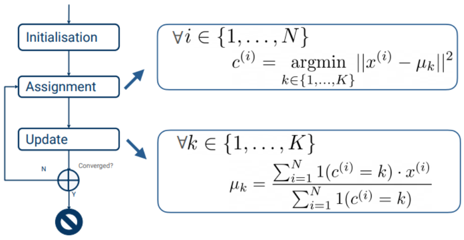
Data normalisation

Min-Max: $X' = a + \frac{(X - X_{min})(b-a)}{X_{max} - X_{min}}$

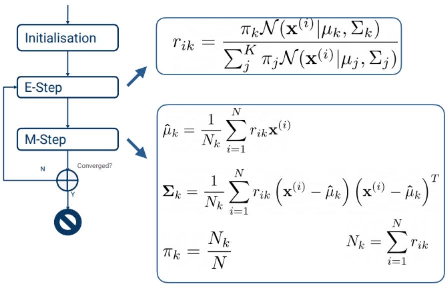
Standardization: $X' = \frac{X - \mu}{\sigma}$

4 Unsupervised Learning

K-means



GMM-EM



Mixture Models:
$P(x) = \sum_{k=1}^{K} \pi_k p_k(x)$
Gaussian Mixture models:
$P(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sum_k)$

5 Evolutionary Algorithms

6 General

Overfitting

1. high dimensionality $\rightarrow$ overfitting
2. Stop for decision trees (Pruning, early stopping (max depth))
3. General Solutions (more data, stop earlier and change complexity (use validation set to test))
4. Neural networks (dropout/regularisation/normalisation)