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∞	$\max_{k=1}^{K} \lvert x_k^{(i)} - x_k^{(q)} \rvert$

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

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

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

2 Evaluation of Machine Learning Systems

Parameter Estimation *Case 1:* Plenty of data available - Held-out test set

1. Train algorithm on **training set**
2. Tune hyperparameters on **validation set** 60:20:20 or 80:10:10 split
3. Estimate generalisation performance using the **test set**

Case 2: Limited data available - Cross-validation

1. Separate dataset into **k folds**
2. Use 1 fold for testing and k-1 folds for **training+validation**
3. Repeat k times, using each fold as the **test set**
4. Estimate generalisation performance by **averaging results** across all the test folds

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

For Classification use: Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
Precision = $\frac{TP}{TP+FP}$
Recall = $\frac{TP}{TP+FN}$
F-measure = $F_1 = 2\frac{Precision*Recall}{Precision+Recall}$
For regression use: MSE or RMSE 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

Perceptron:
$\theta_i = \theta_i + \alpha(y - h(x))x_i$
Classification:
Binary: Only two possible classes
Multi-class: More than one possible class but each class is distinct
Multi-label: Each input can belong to more than one class

Activation Functions:

Function	$g(z)$	$g'(z)$	use
Threshold	1: $W^T x \geq 0$ 0 : otherwise	N/A	Perceptron
Linear	Duh	Duh	Regression
Sigmoid	$\frac{1}{1+e^{-x}}$	$g(z)(1-g(z))$	binary or multi-label classification
Tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - g(z)^2$	binary or multi-label classification
ReLU	0 for $x \leq 0$ x for $x > 0$	0 for $x \leq 0$ 1 for $x > 0$	Used in deep layers and regression
Softmax	$\frac{e^{Z_i}}{\sum_k e^{Z_k}}$	$\frac{\delta L}{\delta z} = \frac{1}{N}(\hat{y} - y)$	multi-class classification

Derivative of softmax only works for Cross entropy loss and N is the number of datapoints Loss Functions:

Name	Definition	Use
MSE	$\frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2$	Regression
Binary cross-entropy	$-\sum_{i=i}^{N} (y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$	binary or multi-label classification
Categorical cross-entropy	$-\sum_{i=1}^{N} \sum_{c=1}^{C} y_c^{(i)} \log(\hat{y}_c^{(i)})$	multi-class classification

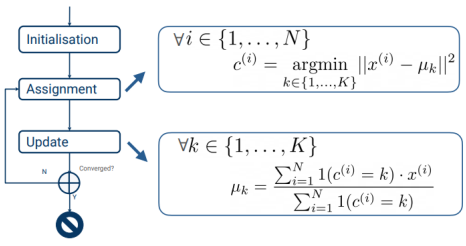
Back Propagation:
$\frac{\delta \text{Loss}}{\delta W} = \frac{\delta \text{Loss}}{\delta Z} \cdot \frac{\delta Z}{\delta W}$
$\frac{\delta \text{Loss}}{\delta b} = \frac{\delta \text{Loss}}{\delta Z} \cdot \frac{\delta Z}{\delta b}$
$\frac{\delta \text{Loss}}{\delta X} = \frac{\delta \text{Loss}}{\delta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

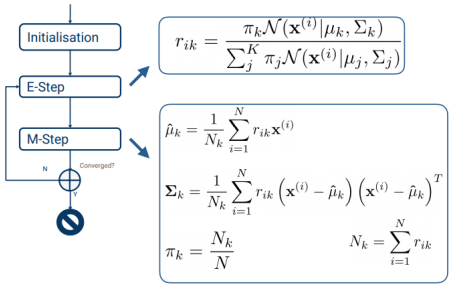
Data normalisation
Min-Max: $X' = a + \frac{(X - X_{min})(b-a)}{X_{max} - X_{min}}$
Standardization: $X' = \frac{X - \mu}{\sigma}$

4 Unsupervised Learning

K-means



GMM-EM



Mixture Models:
$p(x) = \sum_{k=1}^K \pi_k p_k(x)$
Gaussian Mixture models:
$p(x|\theta) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \sum_k)$
Negative Log likelihood
$\mathcal{L} = -\sum_{i=1}^N \log P(x^{(i)}|\theta)$
Bayesian Information Criterion (BIC)
$BIC_K = \mathcal{L}(K) + \frac{P_K}{2} \log(N)$
GMM-EM vs K-means:

K-means	GMM-EM
Objective function: Minimize average mean squared distance	Objective function: Maximize log-likelihood
EM-like algorithm - Assignment: Assign points to cluster - Update: Optimise cluster	EM algorithm - E-step Compute posterior probability of membership - M-step: Optimise parameters
Performs hard assignment during Assignment step	Performs soft assignment during E-step
Assume spherical clusters with equal probability for each cluster	Can be used for non-spherical clusters Can generate clusters with different probabilities

5 Evolutionary Algorithms

genotype vs phenotype: gene representation vs what's shown to environment genetic algorithm: intialize(bitmaps for genomes), evaluate, select with selection operator(random roulette biased on fitness, tournament by picking 1 from random 2), crossover using crossover operator(single-point crossover: swap a prefix of parents for children), mutate using mutation operator(random bit flips) elitism (GA): fix the top n% of the population and ensure it survives (the fitness of the best individual cannot decrease) stopping criterion: individual with max fitness or fixed number of epochs or no improvement visible

$\mu + \lambda$-ES algorithm: initialize $\mu + \lambda$ individuals (float array genotype), evaluate, select best μ, generate λ offsprings by gaussian perturbations from random parents; usually $\frac{\lambda}{\mu} \approx 5$ main challenge is fixing $\sigma 2$ for the gaussian distribution one solution for fixing σ is to add separate σs for each gene and update them based on $\sigma \text{new} = \sigma \exp(\tau_0 \mathcal{N}(0,1))$ where τ_0 is the learning rate (usually $\tau_0 = \sqrt{n}$ where n is the number of genes)

novelty search: instead of optimizing for fitness optimize for novelty(using novelty archive composed of all genotypes; average distance from kNN in novelty archive) behavioral descriptor(BD): defines the "type of solution" that is generated (not necessarily linked to task); several solutions can have same BD but different fitnesses; evaluation AFTER selection

Quality Diversity optimization: mix of novelty search and ES (interesting and high-performing solutions) - highest performing solutions for each region of novelty archive; general pipeline: select population from collection with selection operator, random mutation, evaluation, tenative addition to the collection MAP-Elites - Multidimensional Archive of Phenotypic Elites: discretize the BD into a grid to try to fill it with best solutions; hyperparameter is the size of cells quantifying performance: the diversity of the container (archive size), the performance of the container (max fitness), the convergence speed of the 2 points, QD score (sum of fitness of all solutions in the archive) selector in MAP-Elites: usually uniform random gradient descent can also be applied here using a critic or gradient estimation

6 General

Overfitting
- high dimensionality → overfitting
- Stop for decision trees (Pruning, early stopping (max depth))
- General Solutions (more data, stop earlier and change complexity (use validation set to test))
- Neural networks (dropout/regularisation/normalisation)

Criteria	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	The machine learns by using labeled data	The machine is trained on unlabeled data without any guidance	An agent interacts with its environme by performing actio and learning for errors or rewards
Types of problems	Regression and classification	Association and clustering	reward based
Type of data	Labeled data	Unlabeled data	No predefined dat
Training	External supervision	no supervision	no supervision
Approach	Maps the labeled inputs to the known outputs	Understand patterns and discovers the outputs	Follows the trial and error method