AI4Good.

Curriculum cheat sheet

Probability distributions

Bernoulli(p) has 2 outcomes only and p is the probability of "success". Ex: getting head in one coin toss.

Binomial(n, p) is a repetition of Bernoulli(p) n times. Ex: number of heads in 100 coin tosses.

Uniform(a, b) is for data generated randomly and equiprobably between a and b. Ex: number generator.

Normal(μ, σ^2) is for data symmetrically distributed with a σ^2 variance and concentrated around μ.

Poisson(λ) counts the occurrences of an event in a timeframe, with λ being the average in one unit of time. Ex: number of calls during a day if λ = 5 calls per hour on average.

Exponential(λ) measures a lifespan/waiting period. Ex: time before being served by a bank clerk.

Most used probability laws

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

$P(A) = \sum_B P(A \cap B)$

$P(A \cap B) = P(A)\,P(B|A)$

$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)\,P(A)}{P(B)}$ (Bayes' rule)

Maximum Likelihood Estimation

The MLE of θ is the value of θ that is most likely to generate our observed dataset. Our data (x_i) needs to be iid for this. To find it, we solve :

$\nabla_\theta \log P(data|\theta) = 0$

$\nabla_w J(w) = 0 \quad \leftrightarrow w = (X^T X)^{-1} X^T y$

Bayesian Estimation

In Bayesian Estimation, we want to maximize the **a posteriori** distribution, given by :

A Posteriori ∝ A Priori × Likelihood $L(\theta)$

The **a priori** distribution is assumed based on the data and our intuition.

Feature scaling

Used in pre-processing, when the range of the data varies widely, to bring the data the same scale. Common techniques are:

Min-max normalization : $x' = \frac{x - \min(X)}{\max(X) - \min(X)}$

Mean normalization : $x' = \frac{x - \text{mean}(X)}{\max(X) - \min(X)}$

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

Unit length : $x' = \frac{x}{||x||}$

Dimensionality reduction

Used in pre-processing to reduce the number of features (dimensions) as much as possible without losing too much information on the data.

Principal Component Analysis (PCA) is the main linear method for reduction. It maps the data into a lower-dimensional space using the principal components that capture the most data variability. It is used after feature scaling.

Non-linear methods : **autoencoders** and **t-SNE**.

Bias-variance decomposition

$MSE = Variance + Bias^2 + Noise$

Optimization

Used to maximize or minimize a function (usually loss). All optimization algorithms are affected by the **bias-variance trade-off** (too much variance → overfitting).

Gradient descent, initialized at a point θ :

$$\theta \leftarrow \theta - \eta \times \nabla_\theta J(\theta)$$

where η is the **learning rate** and $J(\theta)$ is the **loss function** that we want to minimize. We stop when θ stops changing significantly.

Stochastic GD uses the gradient of only one instance of x_n for each step.

Batch GD uses the mean gradient of all n instances of x_n for each step.

Mini-batch GD uses the mean gradient of $n' < n$ instances of x_n for each step.

Cross-validation

Used to estimate the true error of a predictor without using the test data.

K-folds cross-validation splits the training data into K sets and trains the model on all but one set, then uses that last set to assess model performance. It repeats across all sets (so a total of K times) to average the error.

Regression algorithms

Ordinary Least Squares calculates the line that has the smallest sum of squared distances.

Ridge (L2) regression helps avoid overfitting by adding $\lambda \sum w_i^2$ to the error function. λ is chosen with cross-validation and w_i are the weights in the model. It will converge (but not equal) the weights of irrelevant parameters to 0.

Lasso (L1) regression helps avoid overfitting by adding $\lambda \sum |w_i|$ to the error function. λ is chosen with cross-validation and w_i are the weights in the model. It will equal the weights of irrelevant parameters to 0.

Classification algorithms

Logistic regression uses a logistic sigmoid function to plot the probability of Y = 1 or Y = 0 given X.

K-nearest neighbours identifies groups in the training data and assigns the test data to the group that the majority of its K neighbours belongs to. If there is a tie, it "flips a coin".

Support Vector Machines assigns a hyperplane (line, circle, etc.) that best separates the data. It allows for multidimensionality.

Other common algorithms are **Naïve Bayes**, **Decision Trees** and **Random Forest**.

Clustering algorithms

K-means clustering randomly assigns groups the data into K clusters, computes the centroids, then reassigns each point to its closest centroid using a distance metric (Euclidian, keyword, etc.). It is often used in data compression and encoding.

Gaussian Mixture Models assume that the data comes from a number of Gaussian distributions that each represent a cluster and tries to identify them.

Neural networks

CNNs, RNNs add text here.

Forward and backpropagation.

Activation functions

The **activation function** of a node in a NN defines the output of that node given an input. The most common ones are :

Identity : $f(x) = x$

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

Rectified Linear Unit : $f(x) = \max\{0, x\}$

Heaviside : $f(x) = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases}$

Add softmax function here.

Missing values

Gradient ascent

Expectation minimization (hard EM, soft EM)

Performance metrics

Common ways of measuring accuracy are :

Classification accuracy : $\frac{correct\ predictions}{total\ predictions}$

Confusion matrix : $\frac{TP+TN}{total\ sample}$

Area Under Curve : the area under the curve of TP Rate ($\frac{TP}{FN+TP}$) by FP Rate ($\frac{FP}{TN+FP}$).

F1 Score : $\frac{2}{\frac{TP+FP}{TP} + \frac{TP+FN}{TP}}$

Cross-entropy loss (or **log loss**) is used for classification models with a probability output ∈ [0,1]. For binary classification :

$$-(y\ log(p) + (1-y)\ log(1-p))$$

where y is the binary indicator and p the predicted probability.

Lastly, you can use **Mean Square Error** (L2 loss) or **Mean Absolute Error** (L1 loss).

Sources :

AI4Good curriculum

Wikipedia