COMP5318 Machine Learning and Data Mining

UNIVERSITY OF SYDNEY · POSTGRADUATE COURSEWORK · SEMESTER 2, 2026

Full guide + ask Sia about COMP5318

SIDE 1/2 WHAT EARNS SPACE · rules → formulas → assumptions → decision cues

MEMORY AID PLANNER

Final Exam **60%** · paper-based · 2 hours

0 · Exam Map

COURSE DATA · L13

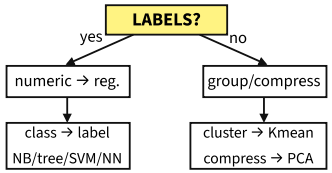
5 | A1 15 | A2 20 | FINAL EXAM 60%

ASSESSMENT	WT	FORMAT
Weekly Homework Quizzes ×10	5%	Canvas, W2–11
Assignment 1	15%	group program + report
Assignment 2	20%	group program + report
Final Exam	60%	paper, 2 h; calculator

Permitted materials during the exam: One sheet of your own notes - double-sided, A4 size, handwritten or typed *(COMP5318 lecture 13 slides)*

Hurdle: score ≥40% in the final or the final course mark is capped at 45; ≥50% overall is required to pass. *(course data)*

★ **Use this as a selection plan:** copy only what you can explain, prioritising procedures, assumptions, denominators and traps.



Task fork: target type selects regression/classification; no labels selects clustering or PCA. *Ch 1 topic map*

1 · Scale, Distance, kNN

CH 1–2

SCALING + SIMILARITY

```
min-max: x'=(x-min)/(max-min)
z=(x-x̄)/s
D₂(A,B)=√Σ(aᵢ-bᵢ)² · D₁=Σ|aᵢ-bᵢ|
cos(A,B)=A·B/(‖A‖‖B‖)
```

Scale before distance: otherwise the largest-range feature dominates. Euclidean = straight-line; Manhattan = axis steps; cosine compares direction rather than magnitude.

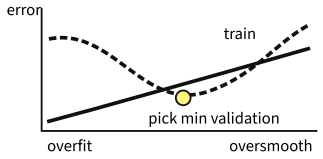
kNN: distance → sort → take k → majority vote (classification) or mean (regression). Compare squared Euclidean distances to rank. Choose k by validation error; small k = low bias/high variance, large k = high bias/low variance; k=n predicts the majority.

EXAM BITE → *Lazy learning stores all rows; one query over m rows × n features costs about m×n distance operations. Prefer an eager model for real-time prediction.*

1b · Risk + Rule Baselines

CH 1–2

ITEM	DECISION CUE
empirical risk	average loss on training data
true risk	expected unseen-data loss; estimate on held-out data



Pick k at the validation-error minimum, never at the training-error minimum. *Source: knn-rule-based fig 2*

2 · Linear Regression

CH 3

LEAST SQUARES

```
ŷ=b₀+b₁x
b₁=[Σxy-ΣxΣy/n]/[Σx²-(Σx)²/n]
b₀=ȳ-b₁x̄ · eᵢ=yᵢ-ŷᵢ · SSE=Σeᵢ²
```

FIT SCORE

```
SST=Σ(yᵢ-ȳ)² = SSR+SSE
R²=SSR/SST=1-SSE/SST · r=sign(b₁)√R²
```

Where it bites: show Σx, Σy, Σxy, Σx², then substitute; predict only after fitting. Least squares minimises *squared vertical residuals*, not their raw sum. Test-set R^2 can be negative when predictions are worse than using ȳ.

3 · Logistic + Penalties

CH 3

CLASSIFICATION PROBABILITY

```
z=w·x+b · p(y=1|x)=σ(z)=1/(1+e⁻ᶻ)
p≥0.5 ⇔ z≥0 → class 1
```

METHOD	OBJECTIVE/EFFECT	USE
Linear	least-squares SSE	numeric target
Logistic	maximum likelihood	binary target
Ridge L2	+αΣw_j^2; shrink	retain all features
Lasso L1	+α$\Sigma\|w_j\|$; zeros	feature selection

Larger α → heavier penalty → simpler model. Never claim logistic regression minimises SSE.

9 · PCA

CH 6

Procedure: centre → covariance → eigenvectors/eigenvalues (or SVD $X=U\Sigma V^T$) → order λ descending → project onto first k components.

VARIANCE RETAINED

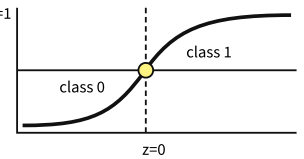
```
first k PCs = Σᵢ₌₁ᵏλᵢ / Σⱼλⱼ
```

PCA is unsupervised, ignores labels and yields orthogonal PCs. Kernel SVM maps upward to separate; PCA projects downward to retain variance.

9b · Regression Checks

CH 3

TWO FAST TRAPS → *R^2 is a fraction of variation explained, not the slope. Logistic p is nonlinear in x, but log-odds are linear; z=0 is the default boundary.*



Sigmoid threshold: p=0.5 exactly at z=0; log-odds are linear although p is not. *Source: regression fig 2*

4 · Naïve Bayes

CH 4

FACTOR, SCORE, ARGMAX

```
P(H|E)=P(E|H)P(H)/P(E)
score(c)=P(c)∏ⱼP(xⱼ|c) · choose
argmax c
```

Nominal feature: likelihood = class-conditional count/class count. **Zero frequency:** Laplace add-one smoothing stops one unseen value making the whole product zero. **Numeric:** use the Gaussian density with each class's mean and sample SD.

GAUSSIAN LIKELIHOOD

```
p(x|c)=1/(√(2π)σc) · exp[-(x-μc)²/(2σc²)]
```

Assumption: features are conditionally independent given the class. Compare unnormalised class scores; P(E) is common.

5 · Honest Evaluation

CH 4

CHOICE	RULE
Holdout	fast; train/test once
k-fold CV	each row tested once; average k scores
Stratified	preserve class proportions
LOOCV	k=n; costly, low bias

Train weights; **validation/CV** tunes k, α, depth; **test** estimates final generalisation. Never tune on test.

METRICS

```
accuracy=(TP+TN)/N
precision=TP/(TP+FP) ·
recall=TP/(TP+FN)
F1=2PR/(P+R)=2TP/(2TP+FP+FN)
```

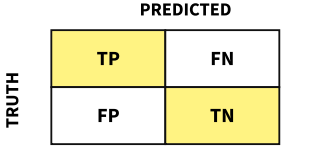
EXAM BITE → *Imbalanced classes: accuracy can hide failure. Precision answers "flagged: how many right?" Recall answers "actual positives: how many caught?"*

11 · Deep Architecture Chooser

CH 8

PROBLEM	PICK	REASON
image	CNN	local + shared weights
short sequence	RNN	hidden state
long dependency	LSTM	cell + 3 gates
whole sequence	Transformer	self-attention

GRU is the lighter two-gate alternative. Pooling: max keeps the strongest response; average keeps the mean.



Rows are truth, columns prediction; diagonal cells are correct. Build counts before metrics. *Source: Naïve Bayes/evaluation fig 3*

6 · Trees: Split + Stop

CH 5

IMPURITY

```
H(S)=-Σᵢpᵢlog₂pᵢ · 0log₂0:=0
Gini(S)=1-Σᵢpᵢ²
```

INFORMATION GAIN

```
Gain(S,A)=H(S)-Σₖ(|Sₖ|/|S|)H(Sₖ)
```

choose the largest gain

Entropy: pure=0; binary 50/50=1 bit. Numeric feature: sort values, try midpoint thresholds where labels change, score each. Weight every child by its row share.

High-cardinality trap: an ID-like attribute makes singleton pure leaves and memorises training data. Drop it or use gain ratio. **Pre-pruning** stops early; **post-pruning** grows then cuts using validation performance.

7 · Ensembles

CH 5

	BAGGING / RF	BOOSTING
Train	parallel, independent	sequential
Data	bootstrap rows	reweight mistakes
Combine	equal vote	weighted vote
Main gain	↓ variance	↓ bias
Noise	more robust	more sensitive

Random forest = bagging + random feature subset at each split, decorrelating trees. Bootstrap means sample *with replacement*.

7b · Tree Decision Checks

CH 5

SITUATION	RESPONSE
numeric feature	try midpoints where class changes
deep tree overfits	pre-stop or post-prune using validation
unstable high variance	bag independent bootstrap trees
correlated trees	random feature subset per RF split
persistent bias	sequential boosting

Extra checks: Gini is 0 at a pure node and 0.5 at binary 50/50. Gain ratio counters information gain's bias toward many-valued splits. Bagging combines by majority vote or averaging; boosting combines weighted learners and is more noise-sensitive.

8 · SVM + Kernels

CH 6

HARD-MARGIN GEOMETRY

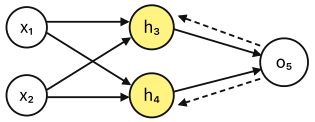
```
boundary: w·x+b=0
supports: w·x+b=±1
constraint: yᵢ(w·xᵢ+b)≥1
margin=2/‖w‖ ⇔ minimise ½‖w‖²
```

Support vector iff $y_i(w\cdot x_i+b)=1$; a point strictly beyond the margin is not one. Solve the ±1 support equations: direction alone does not fix ‖w‖ or margin.

C: larger C penalises training errors more; smaller C tolerates errors for a wider/softer margin. **Kernel trick:** $K(x,y)=\Phi(x)\cdot\Phi(y)$ computes a feature-space inner product without constructing Φ.

COMMON KERNELS

```
polynomial: K(x,y)=(x·y+1)ᵈ
RBF: γ controls boundary smoothness
```



forward → · ← backward

A 2→2→1 sigmoid net: forward activations, then backward error signals. *Source: neural networks fig 2*

10 · Neurons + Backprop

CH 7

FORWARD

```
n=Σwᵢxᵢ+b · o=σ(n)=1/(1+e⁻ⁿ)
σ'(n)=o(1-o)
```

BACKWARD + UPDATE

```
δout=(t-o)o(1-o)
δh=oh(1-oh)Σout(δout·who)
Δw=η·δreceiver·osource · Δb=ηδ
```

Perceptron: step output; e=t−a; w←w+ex, b←b+e. It converges only for linearly separable data; one unit cannot learn XOR.

Dense parameter count: $n_{in}n_{out}+n_{out}$ (one bias per receiving neuron). Sigmoid saturation gives tiny derivatives → vanishing gradients; course fixes include ReLU, better initialisation and dropout.

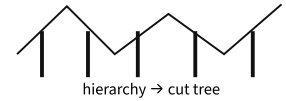
10b · Boundary / Update Checks

CH 6–7

CLAIM	CHECK
SVM support	$y_i(w\cdot x_i+b)=1$ exactly
outside margin	strictly >1; not a support vector
perceptron correct	e=0 ⇒ no weight moves
input $x_i=0$	its perceptron weight does not move
backprop edge	δ belongs to receiving unit

Show direction and scale: solving the support equations fixes w and b; guessing only the boundary direction can give the wrong margin.

K-means	GMM	DBSCAN
hard / round	soft assign	shape/noise



hierarchy → cut tree

Choose by geometry and output: hard means, soft memberships, density/noise or hierarchy. *Source: clustering figs 1–2*

12 · Clustering

CH 9

K-means: choose k centroids → assign nearest → replace each centroid by member mean → repeat until unchanged. Minimises within-cluster SSE; sensitive to start, scale and outliers; prefers compact spherical clusters.

GMM/EM: soft probabilistic generalisation: E-step assigns responsibilities; M-step updates parameters.

Agglomerative: start singleton, merge closest clusters; single=min pair distance, complete=max, average=mean.

DBSCAN: with Eps and MinPts, core has enough neighbours, border is reachable from a core but not core, noise is neither. Finds arbitrary shapes/noise; varying densities are hard.

QUALITY

```
SSE=ΣclustersΣx∈C‖x-μC‖²
silhouette s=(b-a)/max(a,b)
```

13 · HMM + Reinforcement Learning

CH 10–11

MARKOV / HMM

```
P(π₁…πK)=P(π₁)∏ᵢP(πᵢ₊₁|πᵢ)
fₖ(1)=A₀(k)eₖ(x₁)
fₖ(t)=eₖ(xₜ)Σⱼ[fⱼ(t-1)aⱼₖ]
P(X)=Σₖfₖ(T)
```

Viterbi: same trellis but replace Σ with max, store back-pointers, trace argmax. Rows of A are *from*, columns *to*; each row sums to 1.

RETURN + BELLMAN/Q

```
Gₜ=Σₖ≥0γᵏrₜ₊ₖ₊₁
V*(s)=maxₐΣs'P(s'|s,a)[r+γV*(s')]
Q←Q+α[r+γmaxₐ'Q(s',a')-Q]
```

ε-greedy: explore with ε; otherwise exploit argmax Q. Terminal next state: max term=0. Experience replay stores (s,a,r,s'), samples random minibatches, breaks sequential correlation and reuses experience.

Exam split: Forward evaluates P(observations) by summing every hidden path; Viterbi decodes one most likely path by max plus back-pointers; Baum–Welch/EM learns A, E and A_0. In an MDP, (S,A,R,P,γ) names states, actions, reward, transitions and discount.

Your exam allows your own A4 notes only: build your own sheet from this plan; do not bring this printout.

asksia.ai

FORWARD THIS TO YOUR TUTORIAL GROUP · SHARED BY _____

https://www.asksia.ai/cheatsheet/usyd-comp5318

v2 · 2026-10-16 · Semester 2 formal examination period, from 16 November 2026 (the Exams Office publishes the exact date).

COMP5318 Machine Learning and Data Mining · Worked Drills

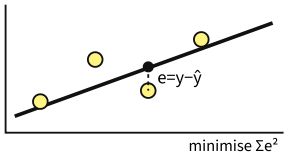
REPRODUCE EACH PIPELINE UNAIDED · THEN DIAGNOSE THE FIRST WRONG LINE

Full guide + ask Sia about COMP5318

SIDE 2/2 FIVE WORKED DRILLS · traps · fast recall · method chooser

DO → CHECK → EXPLAIN

Keep the guide's numbers · carry every calculation to the answer



Least squares minimises squared vertical residuals; show the residual column before R^2. *Source: regression fig 1*

Drill 1 · Fit + Score

CH 3 WORKED EXAMPLE

Given: (x,y)=(1,1),(2,3),(3,2),(4,4); n=4. Fit $\hat{y}=b_0+b_1x$, predict x=5, compute R^2.

1. $\Sigma x=10$, $\Sigma y=10 \Rightarrow \bar{x}=\bar{y}=2.5$; $\Sigma xy=29$; $\Sigma x^2=30$.
2. $b_1=[29-(10\cdot10)/4]/[30-100/4]=4/5=$**0.8**.
3. $b_0=2.5-0.8(2.5)=$**0.5**; $\hat{y}=0.5+0.8x$.
4. $x=5 \Rightarrow \hat{y}=$**4.5**.
5. $\hat{y}$: 1.3,2.1,2.9,3.7; e: −0.3,0.9,−0.9,0.3; SSE=.09+.81+.81+.09=**1.80**.
6. SST=$(-1.5)^2+.5^2+(-.5)^2+1.5^2=$**5.00**; $R^2=1-1.8/5=$**0.64**. Thus $r=+\sqrt{.64}=+0.8$.

CHECK → *SSR=5−1.8=3.2; SSR/SST=.64. A bare answer loses the auditable method trail.*

Fast Recall · Distance + kNN

CH 1–2

PROMPT	WRITE FIRST
normalise	(x−min)/(max−min), not x/max
rank Euclidean	compare squared distances
classify	scale → distance → sort → k → vote
select k	validation minimum; k≤√n is a rule of thumb

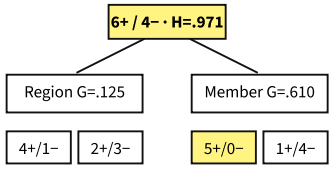
Foundation Checks

CH 1–2

STATEMENT	VERDICT + REASON
lowest train error is best	false; estimate true risk on unseen data
unlabelled grouping predicts a class	false; clustering discovers structure
cosine tracks magnitude	false; it compares vector direction
k=1 is always best	false; low bias, high variance
kNN has costly training	false; lazy training, costly prediction

1R builds one rule from the single best attribute.

PRISM learns a covering set of IF–THEN rules. They are baselines, not distance methods.



Member wins: its pure child drives weighted entropy down and gain up. *Source: Ch 5 worked example*

Drill 2 · Entropy + Gain

CH 5 WORKED EXAMPLE

Given: parent 6 Buy/4 No-buy. Region → North 4+/1−, South 2+/3−. Member → yes 5+/0−, no 1+/4−.

1. $H(parent)=-.6\log_2.6-.4\log_2.4=.6(.737)+.4(1.322)=$**.971 bits**.
2. $H(4/1)=-.8\log_2.8-.2\log_2.2=$**.722**; $H(2/3)=H(.4/.6)=$**.971**.
3. Weighted Region=.5(.722)+.5(.971)=**.846**; Gain=.971−.846=**.125**.
4. Member yes is pure ⇒ H=0; Member no (1/4) ⇒ H=.722.
5. Weighted Member=.5(0)+.5(.722)=**.361**; Gain=.971−.361=**.610**.
6. Choose Member: .610>.125.

CHECK → *Memorise anchors: pure=0; 50/50=1; 80/20≈.722; 60/40≈.971. The child average must be weighted.*

Tree / Ensemble Traps

CH 5 FAQ

WRONG MOVE	CORRECTION
plain mean of child H	weight by $\lvert S_k\rvert/\lvert S\rvert$
ID gives perfect gain	memorisation; drop/use gain ratio
RF = boosting	RF is bagging + feature subsets
bag without replacement	bootstrap is with replacement

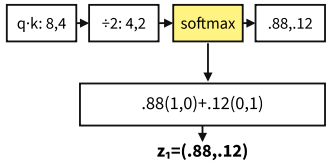
Split / Prune Recall

CH 5 TOPIC MAP

Numeric split: sort values; place candidate thresholds at midpoints where the class label changes; compute impurity reduction for each. A feature may be reused deeper with another threshold.

Pre-pruning halts growth by a limit such as depth or node size. **Post-pruning** grows the tree, then removes branches whose removal improves validation behaviour.

Sanity checks: gain cannot exceed parent entropy; pure children contribute zero; a split with unchanged class proportions has zero gain.



Attention's four lines: dot → scale → softmax → weighted values. *Source: deep learning fig 4*

Drill 3 · Self-Attention

CH 8 WORKED EXAMPLE

Given: $q_1=(2,0,2,0)$; $k_1=(2,1,2,1)$, $k_2=(1,1,1,1)$; $v_1=(1,0)$, $v_2=(0,1)$; $d_k=4$.

1. Scores: $q_1\cdot k_1=2\cdot2+0\cdot1+2\cdot2+0\cdot1=$**8**; $q_1\cdot k_2=2\cdot1+0\cdot1+2\cdot1+0\cdot1=$**4**.
2. Scale by $\sqrt{d_k}=\sqrt{4}=2 \Rightarrow$ scores **4, 2**.
3. Softmax: $e^4=54.60$, $e^2=7.39$; sum=61.99 ⇒ weights **.88, .12**.
4. $z_1=.88(1,0)+.12(0,1)=$**(.88,.12)**.

Interpret: token 1 contributes 88% because its key matches q_1 more strongly. Weights sum to 1; output dimension equals the *value* dimension.

CHECK → *Scale before softmax. Softmax on raw 8,4 is the classic lost line.*

Neural / Deep Fast Recall

CH 7–8

PROMPT	MARK-WINNING PHRASE
why CNN?	local receptive fields + shared weights + translation invariance
RNN weakness	products of derivatives → vanishing/exploding gradient
why LSTM?	cell state; forget/input/output gates preserve long context
parameters	incoming weights + one bias per receiving unit

Backprop / Convolution Checks

CH 7–8

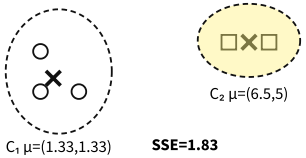
BACKPROP ORDER

```
forward hidden → forward output →
δout → update hidden→output →
δhidden → update input→hidden
```

For sigmoid, derivative is $o(1-o)$, using the neuron's already-computed output. Each weight change is η × receiver delta × source output; bias uses source 1.

Convolution: multiply each image patch elementwise by the shared filter and sum. With a 3×3 input, 2×2 filter, stride 1 and no padding, the guide's map is 2×2 [[2,4],[4,2]]; max pool gives **4**, average pool gives **3**.

Transformer check: positional encoding adds order because attention is order-agnostic. Multi-head attention uses distinct Q/K/V projections of one embedding, then concatenates the outputs.



SSE=1.83

Means need not be data points; recompute, reassign, stop only when membership is unchanged. *Source: clustering fig 1*

Drill 4 · K-Means

CH 9 WORKED EXAMPLE

Given: $P_1(1,1)$, $P_2(1,2)$, $P_3(6,5)$, $P_4(7,5)$, $P_5(2,1)$; k=2; initial $c_1=(1,1)$, $c_2=(6,5)$; squared Euclidean.

1. Distance pairs $d^2(c_1)$,$d^2(c_2)$: P_1 0,41; P_2 1,34; P_3 41,0; P_4 52,1; P_5 1,32.
2. $C_1=\{P_1,P_2,P_5\}$; $C_2=\{P_3,P_4\}$.
3. Means: $c_1=((1+1+2)/3,(1+2+1)/3)=(4/3,4/3)\approx$**(1.33,1.33)**; $c_2=((6+7)/2,(5+5)/2)=$**(6.5,5)**.
4. Epoch 2: no reassignment. Example P_2: $d^2(c_1)=.11+.44=.56$ vs $d^2(c_2)=30.25+9=39.25 \Rightarrow$ converged.
5. SSE $C_1=.22+.56+.56=1.33$; $C_2=.25+.25=.50 \Rightarrow$ **SSE=1.83**.

CHECK → *Compare squared distances for assignment; take square roots only when an actual distance is requested.*

Clustering Traps

CH 9

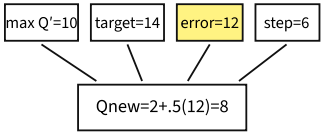
TRAP	FIX
centroid = real row	centroid is mean; medoid is real point
K-means finds noise	DBSCAN labels noise
single linkage	minimum pair distance
complete linkage	maximum pair distance
silhouette near 1	good cohesion/separation

Cluster Family Chooser

CH 9 TOPIC MAP

NEED / GEOMETRY	METHOD	TRADE-OFF
known k, compact groups	K-means	fast; start/outlier sensitive
soft membership	GMM + EM	probabilistic, more parameters
nested grouping	agglomerative	dendrogram; linkage matters
arbitrary shape + noise	DBSCAN	Eps/MinPts; varied density hard

Silhouette: a = mean distance within own cluster; b = best alternative-cluster mean distance. Near +1 is well placed, near 0 is on a boundary, negative suggests a better neighbouring cluster.



α=.5 · γ=.9 · r=5 · Qold=2

Keep target, TD error and α-scaled step as separate lines. *Source: Ch 11 worked example*

Drill 5 · Q-Learning

CH 11 WORKED EXAMPLE

Given: α=.5, γ=.9, Q(s,a)=2, r=5; next-state Q values are 4,10,6.

1. Write $Q \leftarrow Q+\alpha[r+\gamma\max_{a'}Q(s',a')-Q]$.
2. Best next value=max(4,10,6)=**10**.
3. TD target=5+.9(10)=**14**.
4. TD error=14−2=**12**.
5. Qnew=2+.5(12)=**8**.

Interpret: α=.5 moves the estimate halfway from 2 toward 14. If s' is terminal, the max-next term is 0 and target=r.

CHECK → *γ multiplies the maximum next Q, not a sum/mean. α multiplies the entire TD error, not the target alone.*

HMM Recall Drill

CH 10 · SOURCE FIG 2

Forward trellis values for Write, Browse: t=1 gives F=.48, D=.12; t=2 gives F=.0768, D=.1512; therefore P(X)=**.228**.

QUESTION ASKS	OPERATOR	FINISH
P(observations)	Forward: Σ incoming	Σ last column
most likely states	Viterbi: max incoming	argmax + backtrack
sequence probability	initial × transitions	K states ⇒ K−1 transitions

Do not reverse $a_{j\,k}$: row=from, column=to. Multiply the current observation's emission once, outside the incoming sum/max.

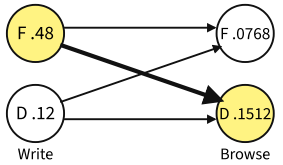
Sequence / RL Checks

CH 10–11

Complexity: Forward/Viterbi dynamic programming is $O(N^2T)$, versus enumerating $O(N^T)$ paths. HMM parts are hidden states, transition A, emission E and initial distribution A_0.

Return: γ near 0 is myopic; γ near 1 values distant rewards. **ε-greedy:** random action with probability ε, otherwise current argmax. The action changes the next state, unlike a fixed supervised dataset.

Deep Q: a network approximates Q; replay memory stores (s,a,r,s') and random minibatches break successive-transition correlation while reusing experience.



Forward sums all incoming paths; Viterbi selects one and records its predecessor. *Source: HMM fig 2*

Final-Pass Method Chooser

ALL TOPIC MAPS

IF THE PROMPT SAYS...	REACH FOR...	FIRST LINE
closest labelled points	kNN	scale + distance
numeric prediction	linear regression	$\hat{y}=b_0+b_1x$
class probability	logistic	σ(w·x+b)
priors + likelihoods	Naïve Bayes	$P(c)\prod P(x_j\mid c)$
best split	tree	parent H−weighted child H
widest boundary	SVM	w·x+b=0; supports ±1
variance compression	PCA	centre → covariance/SVD
small neural update	backprop	forward → δout → δhidden
image	CNN	filter → ReLU → pool
token interactions	attention	$QK^T/\sqrt{d_k}$ → softmax → V
unlabelled groups	cluster	geometry/noise selects family
P(observation seq.)	Forward	Σ incoming × emission
hidden path	Viterbi	max incoming + pointers
rewarded action update	Q-learning	target → error → α step

Common Error Radar

FAQS + STUDY STRATEGIES

- **Data leakage:** tuning on test or scaling with information from test.
- **Wrong denominator:** precision uses predicted positives; recall uses actual positives.
- **Wrong objective:** logistic ≠ least squares; PCA ≠ classifier; clustering has no labels.
- **Wrong operator:** NB multiplies likelihoods; Forward sums paths; Viterbi and Q-learning take maxima.
- **Missing assumption:** scale for distance, conditional independence for NB, linearly separable for hard-margin SVM/perceptron convergence.
- **Missing reason:** after the number, write one sentence interpreting the model choice or failure mode.

LAST CHECK → *Formula → substituted values → result → one-line reason. That four-part trace makes partial credit visible.*

Your exam allows your own A4 notes only: build your own sheet from this plan; do not bring this printout.

asksia.ai

FORWARD THIS TO YOUR TUTORIAL GROUP · SHARED BY _____

https://www.asksia.ai/cheatsheet/usyd-comp5318

v2 · 2026-10-16 · Semester 2 formal examination period, from 16 November 2026 (the Exams Office publishes the exact date).