FINAL PREP:

KNN: Select a random point. Find the K nearest neighbors, classify appropriately.

→ Find distance using a distance metric $d = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$

→ Define K (Sometimes just $K < \sqrt{N}$ and $K \% 2 = 1$)

Small K:
- → Good at capturing patterns
- → May overfit (sensitive to noise)

Large K:
- → Stable/smooth predictions
- → May underfit, fail to capture regularities.

Evaluating Classifier:
- → What's the accuracy?
- → Which ones did it get wrong?
- → How "badly" did it get it wrong?

Generalization error → How well does it perform on unseen data?

+++ No assumptions abt data distribution.
- Easy to implement
- Few hyper parameters
- Adaptive

Assumptions
- Similar data points are close to each other.
- Dataset is representative and sufficiently large to capture variations.

−−
- Bad with scale (becomes slow, lazy)
- Curse of Dimensionality
- Overfits.

→ Difficult to define "nearest" neighbor.

Cross Validation:
Split data into train and validation sets, test various K values.
Pick the K with the largest validation accuracy.

Made with Goodnotes

Information Theory:

Expected value of surprise = entropy.

Quantify the surprise of individual events → $-\log(P(x))$

$E[\text{surprise}] = -\sum P(x)\log(P(x))$

$-P(x)\log(P(x))$

Entropy of Joint Distribution:

$X = \{\text{Rain, Not Rain}\}$
$Y = \{\text{Cloud, Not Cloud}\}$

	Cloudy	Not Cloudy
Raining	0.24	0.01
Not Rain	0.25	0.5

$H(X,Y) = -0.24 \times \log(0.24) - 0.01 \times \log(0.01) - 0.25 \times \log(0.25) - 0.5 \times \log(0.5)$

$= 1.56$ bits

Entropy of Conditional:

Entropy of cloudiness, given it's raining?

$$H(Y|X=x) = -\frac{24}{25} \times \log\left(\frac{24}{25}\right) - \frac{1}{25} \times \log\left(\frac{1}{25}\right)$$

Entropy of cloudiness, given knowledge abt whether it's raining or not.

$$H(Y|X) = P(\text{Rain}).H(\text{Cloudy}|\text{rain}) + P(\text{Not Rain}).H(\text{Cloudy}|\text{Not Rain})$$

$$\frac{1}{4}.H(C|R) + \frac{3}{4}.H(C|NR)$$

→ H is always non negative.
→ $H(x,y) = H(x|y) + H(y) = H(y|x) + H(x)$

If X and Y are independent, meaning x doesn't tell us anything about Y and vice versa.

$H(Y|X) = H(Y)$

$H(Y|Y) = 0$

Information Gain:

How much IG about cloudiness do we get by discovering whether its raining.

$IG(Y|X) = H(Y) - H(Y|X) =$
0.25 bits

If X is uninformative about Y, then $IG(Y|X) = 0$
completely informative $IG(Y|X) = H(Y)$

Made with Goodnotes

Entropy is the measure of uncertainty in a probability distribution. $H(x) = -\sum P(x) \cdot \log(P(x))$

Chain Rule of entropy → $H(X,Y) = H(Y|X) + H(X)$

Unit of measure in entropy → bits

Information gain = Reduction of entropy when splitting a dataset.

Stats:

$P(a,b) = P(b|a) \cdot P(a) = P(a|b) \cdot P(b)$

$P(a,b,c) = P(a|b,c) \cdot P(b,c) = P(a|b,c) \cdot P(b|c) \times P(c)$

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$E[X] = \sum x \cdot P(x)$ $\quad$ $Var(x) = E[(X-\mu)^2]$

$E[X+Y] = E[X] + E[Y]$ $\quad$ $Var[cX] = c^2 var(X)$

$E[cX] = cE[X]$

If X and Y are independent, then

$Var(X+Y) = Var(X) + Var(Y)$

$E[XY] = E[X] \cdot E[Y]$

Advantages:
- Easy to visualize/interpret
- Can do classification and regression
- Non-linear relationships can be captured.

Disadvantages:
- Can overfit
- Sensitive to small changes in data.

Pruning → Cutting back on the size of a decision tree by removing nodes that provide little predictability info, helps reduce complexity and overfitting.

Random forests work by builds multiple decision trees, the final prediction is made by averaging (regression) or voting (classification) across multiple trees, reducing overfitting. It simplifies the tree while maintaining predictive performance.

Decision Trees:

High entropy (Max uncertainty)
- Variable is a uniform-like distribution
- Histogram is flat
- Not predictable

Low entropy
- Distribution has peaks and valleys
- Histogram has highs and lows.
- Sampled values are predictable

Ex

100 Red
49 Blue

50 Red
0 Blue

50 Red
49 Blue

$H(Root) = -\frac{49}{149} \times \log\left(\frac{49}{149}\right) - \frac{100}{149} \times \log\left(\frac{100}{149}\right) \approx 0.91$

$H(Left) = 0$

$H(Right) = -\frac{50}{99} \times \log\left(\frac{50}{99}\right) - \frac{49}{99} \times \log\left(\frac{49}{99}\right) \approx 1$

$IG(split) = 0.91 - \left(\frac{1}{3} \times 0 + \frac{2}{3} \times 1\right) \approx 0.24$ ✓

Good Tree

→ Not too small: Should handle subtle distinctions in data

→ Not too big:
- Overfits
- Expensive

Decision Tree
- Good when lots of attributes
- Can deal with missing values
- Robust to scale
- Fast
- Interpretable

KNN
- Few hyperparams
- Typically makes better predictions

Made with Goodnotes

Linear Regression:

Loss → $\frac{1}{2}(y-t)^2$

$y = wx + b$ (dep - ind. ind. → dep.)

Cost = Loss averaged over all examples = $\frac{1}{2N}\sum(y-t)^2$

$$\frac{dJ}{dw_j} = \frac{1}{n}\sum(y-t)\cdot x^i$$

$$\frac{dJ}{db} = \frac{1}{n}\sum(y-t)$$

Gradient Descent = Used to optimize hyperparams, minimize cost function.

$$\boxed{\begin{aligned} w &\to w - \alpha \frac{d}{dw} J(w,b) \\ b &\to b - \alpha \frac{d}{db} J(w,b) \end{aligned}}$$

↑ bias ↓ Variance → under fit
↓ bias ↑ Variance → overfit.

bias: error due to assumptions (underfitting)

Weight: error due to model complexity and sensitivity to data (overfitting)

$$w = w - \alpha \cdot \frac{dJ}{dw_j}$$

Vectorization → Use parallel processing to make calculations faster.

$$\boxed{\text{np.dot}(w, x) + b}$$

Minimum of a smooth function occurs at a critical point, where the derivative is 0.
Multivariate generalization → Set partial derivatives $\frac{dj}{dw_j}$ to zero.

If $\frac{dJ}{dw_J} > 0$, then $w\uparrow$ $J\uparrow$
< 0, then $w\uparrow$ $J\downarrow$

L_1 Reg: Forces some coefficients to be exactly zero $J(w) = \text{Loss} + \lambda \sum |w_i|$ (Sum of absolute values of w)
Used in cases where we think only a small number of features are relevant.
L_2 Reg = Reduces magnitude of all coefficients but doesn't set any to zero
Prevents one feature dominating others, used when a lot of features are relevant. Sum of squared value of w. Smoother. $J(w) = \text{Loss} + \lambda \sum w_i^2$

Assumptions:

Linearity = Relationship between predictors and outcome is linear.
Independence = Observations are independent of each other.

$2xy$

Overfit → Model is too complex, fits noise well, can't generalize to new data.
Under fit → Model is too simple, doesn't fit data well.

Regularization can be used to handle overfitting.

Batch GD: Use entire dataset on each iteration. (Smoother, slower, fails to capture irregularities)
Stochastic GD: Use one random dataset on each dataset (Struggles with noise, fast)
Mini-batch GD: Use mini-batches.

Slope of regression line indicates how much the dependent variable is expected to change for each unit increase in the independent variable.

Sample run of GD. $J(w,b) = \frac{1}{m}\sum(x-y)^2$
$x = [1,2,3,4,5]$
$y = [3,6,7,8,11]$
$h = wx + b$

$h(x_1) = 0\cdot 1 + 0 = 0$
$h(x_2) = 0\cdot 2 + 0 = 0$
⋮
$h(x_5) = 0\cdot 5 + 0 = 0$

Initial parameters, $w=0$ $b=0$ $\alpha = 0.1$ $m=5$

Calculate Gradients

$$\frac{dJ}{dw_J} = \frac{1}{5}\left[(0-3)\cdot 1 + (0-6)\cdot 2 + (0-7)\cdot 3 + (0-8)\cdot 4 + (0-11)\cdot 5\right] = -24.6$$

$$\frac{dJ}{db} = \frac{1}{5}\left[(0-3)+(0-6)+(0-7)+(0-8)+(0-11)\right] = -7$$

$$w = w - \alpha\frac{dJ}{dw_J} = 0 - 0.1\times(-24.6) = 2.46$$

$$b = b - \alpha\cdot\frac{dJ}{db} = 0 - 0.1(-7) = 0.7$$

Made with Goodnotes

Logistic Regression:

$\{x \mid w^T x = 0\}$ a set of all vectors x orthogonal to w.

If 2D → line

3D → hyperplane.

$w^T x = 0$ → orthogonal

> 0 → acute (less than 90°) → x lies in the same side of the hyperplane as w.

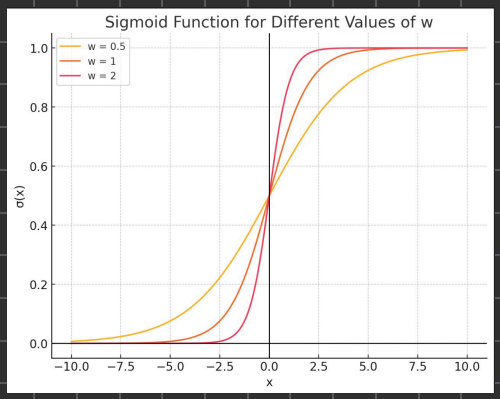
< 0 → more than 90°

$$\boxed{\sigma(x) = \frac{1}{1+e^{-wx}}}$$

approaches 1 as $x \to \infty$

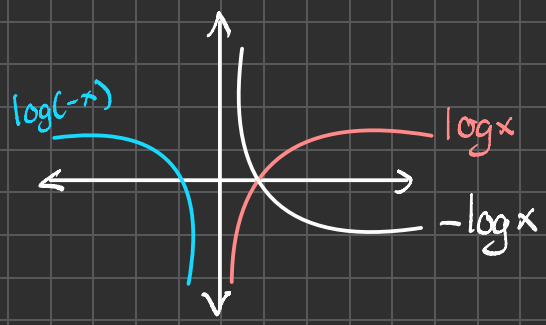
approaches 0 as $x \to -\infty$

0.5 if $x = 0$



As x or w increases:

- Slope at $x=0$ becomes steeper.
- Function transitions sharply between 0 and 1.



$$y = \frac{1}{1+e^{-(wx+b)}}$$

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} y \cdot \log y + (1-y)\cdot\log(1-y)$$

→ Cross entropy loss = Cost function for logistic classification, quantifies how well the predicted class matches the actual class.

$$CEL \to -t \cdot \log(y) - (1-t)\cdot\log(1-y)$$

Assumptions:

→ Log odds of the defendant variable is a linear combination of input features.

→ Independence.

→ Linearity of log odds.

$$\sigma^{-1}(p) = \text{logit}(p) = \ln \frac{p}{1-p}$$

→ Cross entropy is better than MSE because MSE is not convex in the case of logistic regression, meaning there could be a local minima

CEL is convex with respect to weights in logistic regression, which guarantees gradient descent will converge to a global minimum.

CEL is a probabilistic interpretation, it quantifies the difference of probabilities, MSE treats the predictions as continuous and doesn't account that we are modeling probabilities.

↳ penalizes incorrect answers exponentially, correct answers linearly.

Made with Goodnotes

Classifier Evaluation:

Accuracy is the ratio of correct/total predictions.
↓
Good measure if:
- Dataset is balanced
- Misclassifications have equal costs.

Precision (TP/AP) important when cost of false positives is high. Spam email.

Recall (TP/(TP+FN)) " " " " false negatives is high. Diagnosis.

CE loss

$CE = -t \log y - (1-t)\log(1-y)$

t	y
1	0.9
0	0.2
1	0.7
0	0.1

$L_1 = -1\log(0.9) - (1-1).\log(1-0.9) = 0.105$

$L_2 = -0\log(0.2) - (1-0).\log(0.8) = 0.223$

Total Loss $= \frac{L_1+L_2+L_3+L_4}{4}$

$y = wx+b$

$y = 5$ → $\max(0, 5) = 5$

Made with Goodnotes

$X = [4, 2, 3, 5] \quad \bar{x} = 3.5$

$y = [1, 3, 2, 4] \quad \bar{y} = 2.5$

$Var(x) = \frac{1}{3}\cdot(-0.5)^2 + (1.5)^2 + (0.5)^2 + (1.5)^2$

$\frac{1}{3} = 0.25 + 2.25$

$= \frac{1}{3}\cdot 5 = 1.66$

$Var(y) = 1.66$

$Cov(x,y) = (4-3.5)\cdot(1-2.5) + \ldots = 0.33$

$$\begin{bmatrix} 1.66 & 0.33 \\ 0.33 & 1.66 \end{bmatrix}$$

$\det(C - \lambda I) = 0$

$$\begin{matrix} 1.66-\lambda & 0.33 \\ 0.33 & 1.66-\lambda \end{matrix} \qquad (1.66-\lambda)^2 = 0.1111$$

$\lambda_1 = 2$

$\lambda_2 = 1.3$

$$\begin{bmatrix} 1.66-2 & 0.33 \\ 0.33 & 1.66-2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

$$v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.66-1.3 & 0.33 \\ 0.33 & 1.66-1.3 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

$v_2 = \begin{matrix} 1 \\ 1 \end{matrix}$

$$\begin{matrix} w \\ x \\ b \end{matrix} \rightarrow z \rightarrow y - L \quad t$$

$L = \frac{1}{2}(y-t)^2$

$y \rightarrow \sigma(z)$

$z = wx + b$

$(y-t)\,\alpha$

$y - t$

$$\frac{dL}{dw} = \frac{dz}{dw}\cdot\frac{dy}{dz}\cdot\frac{dL}{dy}$$

$$\frac{dL}{dz} = (y-t)\cdot\sigma'$$

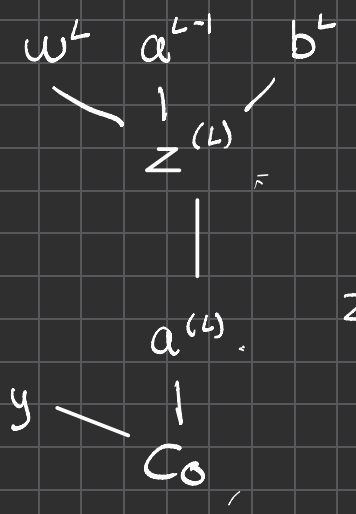
$$\frac{dL}{dw} = \frac{dL}{dz}\,x$$

~~$z = wx + b$~~

Made with Goodnotes

Neural Networks:

Weight



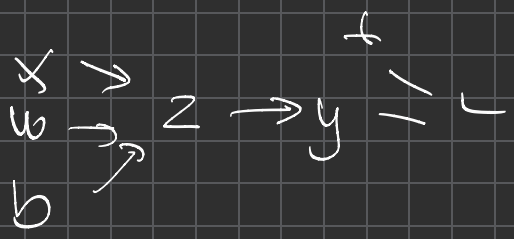
$$C_0 = (a^{(L)} - y)^2$$

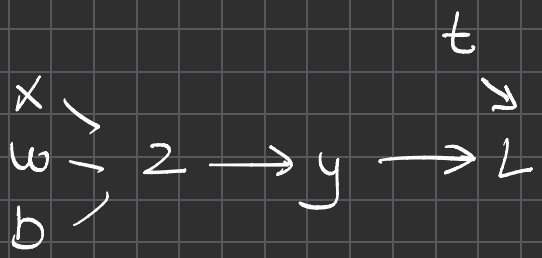
$$a^L = \sigma(z^L)$$

$$z^L = (w^L \times a^{L-1}) + b^L$$

$$\frac{dC_0}{dw_L} = \frac{dz^L}{dw^L} \cdot \frac{da_L}{dz^L} \cdot \frac{dC_0}{da_L}$$

$$a^{L-1} \times \sigma^{-1}(z^L) \times 2(a^L - y)$$





Made with Goodnotes

Univariate Gaussian

$$N(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi \cdot \sigma}} \cdot e^{-\frac{1}{2}\left(\frac{(x-\mu)^2}{\sigma}\right)}$$

$P(G_1) = 0.6 = 60\%$ of the points in the dataset come from G_1

$P(G_2) = 0.4 = 40\%$ of the points come from G_2.

$P(x|G_1)$ = Likelihood of point x under g_1s pdf. $= 0.5$

$P(x|G_2)$ = Likelihood of point x under g_2s pdf $= 0.5$

$$P(g_1|x) = \frac{P(x|g_1) \times P(g_1)}{P(x)} = \frac{0.5 \times 0.6}{P(x)}$$

$$P(x|G_1) \times P(G_1) + P(x|G_2) \times P(G_2)$$

$$P(x) = P(x|G_1) \times P(G_1) + P(x|G_2) \times P(G_2)$$

Made with Goodnotes

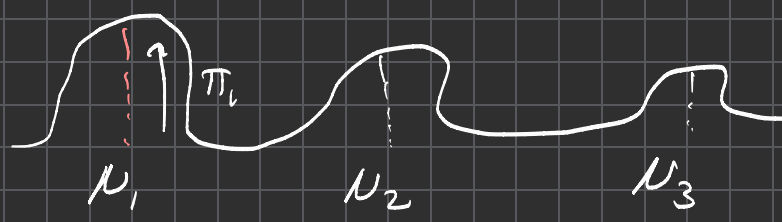
EM:

Mixture of Gaussians

Mean μ_c

Variance σ_c Pdf $p(x) = \sum_c \pi_c \cdot N(x, \mu_c, \sigma_c)$

Size π_c

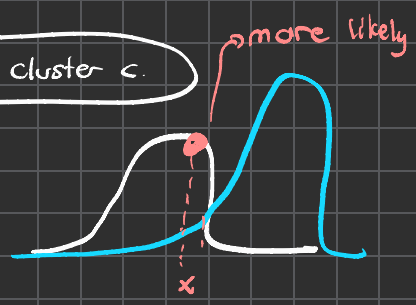


E step:

For each datum x_i,

Compute r_{ic}, probability it belongs to cluster c.

$$r_{ic} = \frac{\pi_c \cdot N(x_i, \mu_c, \Sigma_c)}{\sum_{\text{all } c\text{'s}} \pi_{c'} \cdot N(x_i, \mu_{c'}, \Sigma_{c'})}$$



M step:

For each cluster, update parameters μ_c, Σ_c, π_c

$m_c = \sum_i r_{ic}$ total responsibility of cluster c.

$\pi_c = \frac{m_c}{m}$ Fraction of total responsibility assigned to cluster c

$$\mu_c = \frac{1}{m_c} \sum r_{ic} x^{(i)}$$

$$\Sigma_c = \frac{1}{m_c} \sum r_{ic} \cdot (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c)$$

$$\frac{1}{2} \cdot e^{\left(\left(\frac{x-\mu}{\sigma}\right)^2\right)}$$

$$r_{ic} = \frac{\pi_1 N(x, \mu, \sigma^2)}{\text{all}}$$

$$-\frac{1}{2}(\|x-\mu\|^2$$

Made with Goodnotes

Q1

$N_1 \quad \mu_1 = 2 \quad \sigma_1^2 = 1 \quad \pi_1 = 0.6$

$N_2 \quad \mu_2 = 4 \quad \sigma_2^2 = 2 \quad \pi_2 = 0.4$

$-\frac{1}{2}\|x-\mu\|^2$

$x_n = 3$

$$N_1 = \frac{1}{\sqrt{2\pi \cdot 1}} \cdot e^{-\frac{1}{2}\left(\frac{(3-2)^2}{1}\right)} = a$$

$$N_2 = \frac{1}{\sqrt{2\pi \cdot 2}} \cdot e^{-\frac{1}{2}\left(\frac{(3-4)^2}{2}\right)} = b$$

$$\Gamma_{1n} = \frac{0.6 \times a}{0.6 \times a + 0.4 \times b}$$

Multivariate

μ_k = mean vector of k

Σ_k = Covariance of k

$$N(x, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}} \cdot \exp\left(-\frac{1}{2}\|x-\mu\|^2\right)$$

$$-\|x-\mu\|^2 = -\frac{1}{2}\left((2-1)^2 + (3-2)^2 + (4-3)^2\right) = 3$$

G1 $\quad N(x_1, \mu_1, \Sigma_1) = \frac{1}{\sqrt{2\pi}^{3/2}} \cdot \exp\left(-\frac{1}{2} \cdot 3\right)$

G2 $\quad N(x, \mu, \Sigma) = \frac{1}{\sqrt{2\pi}^{d/2}} \cdot \exp\left(-\frac{1}{2} \cdot \|x-\mu\|^2\right)$

$$\|x-\mu\|^2 = -\frac{1}{2} \cdot \left((5-1)^2 + (6-2)^2 + (7-3)^2\right) = 48$$

$$\frac{1}{\sqrt{2\pi}^{3/2}} \cdot \exp\left(-\frac{1}{2} \cdot 48\right) =$$

$$p_1 = \pi_1 \cdot N(x, \mu_1, \Sigma_1)$$

$$p_2 = \pi_2 \cdot N(x, \mu_2, \Sigma_2)$$

$-\frac{1}{2} \cdot \|x-\mu\|^2$

$(1, 2, 3)$

$(4, 5, 6)$

$(4-1)^2$

Made with Goodnotes

$z = wx + b$

$0.5 \times 2 + 1 = 2$ → larger than 0

$\frac{dL}{dw}$

$\frac{dL}{dz} \cdot \frac{dz}{dw}$

$\text{Loss} = \frac{1}{2}(z-4)^2$

$\frac{dL}{dw} = \frac{dz}{dw} \cdot \frac{dL}{dz}$

$x \cdot (y-t)$

$x \cdot$

$2 \times (-2) = -4$

$\frac{dL}{dz} \cdot \frac{dz}{dw}$

$\frac{dL}{db} = \frac{dL}{dz} \cdot \frac{dz}{db} = -2$

$(y-t) \cdot 1$

$\frac{dL}{db} = \frac{dL}{dz} \cdot \frac{dz}{db}$

$y-t$

1

$P(\text{Rain}) \cdot H(\text{cloudy} \mid \text{is rain})$

$+ P(\text{Not rain}) \cdot H(\text{cloudy} \mid \text{not rain})$

$-P(t) \cdot \log(P_t)$

$P($

$P(P, B) = P(A \mid B) + P(B)$

Draft/Collaborative Notes for 3151 Final Exam

Cheat Sheet

- Conditional Entropy Equation. (specific and Expected)
- $L_{CE}(y,t) = -t\ln(y) - (1-t)\ln(1-y)$
- Normalization xj = (xj - μj)/ σ j
- Regularization term
- Softmax function:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- Mv = v
- Spectral Decomposition
- IG = H(D) - H(D|split)
- H(split) = p(left)H(left) + P(right)H(right)
-
- MSE: loss = |y - t|^2
- Regularization: λ ∑ w^2
- Sigmoid function: 1 / (1 + e^{-z})
- Tanh function: (e^z – e^{-z}) / (e^{z} + e^{-z})

Standard form of an ellipse: x^2/a^2 + y^2/b^2 = 1

I'll provide eq's for sensitivity & specificity

Topics

- kNN
- decision trees
- basics of information theory
    - If i ask you to compute the entropy of a distribution, you can generally leave it as a sum of logs, i.e. you do

Made with Goodnotes

not need to simplify or compute it exactly, e.g. you can write it as “-5/8 log (5/8) + etc”...

- basics of probability
- linear regression
- logistic classification
    - Includes evaluating a classifier
- neural networks
- multivariate gaussians
- density modeling
- gaussian mixture model and EM algorithm

Resources to study:

- Class slides (posted on brightspace)
- Tutorial notes (posted on brightspace)
- Selected Readings by R. Grosse (posted on brightspace)
- Assignments

Questions (contributed by students/TA’s/instructor)

Put your questions here

1.

What is the decision boundary of the gaussian? (draw a few different lines)

- I think I know what the intention of this question , but also please note that as phrased here, the question doesn’t *quite* make sense, i.e. a single gaussian doesn’t really have a decision boundary.

(Taking formula with given matrix and mean to calculate and draw the circles on a graph using exp[½(x-mu)Σ(x-mu)] (that formula not totally accurately written here))

Maybe identity matrix so we dont have to calculate inverse, or just given an inverse as that could take a lot of time

2.

Made with Goodnotes

Calculate the information gain at the split (a picture of divided dots, something able to do by hand) [see my comment about this question]

3

Draw KNN decision boundary on points in a space (midterm 1 redemption)

4.

What does the curse of dimensionality imply?

AI generated - Q3 is not possible

Made with Goodnotes

1. **Single Neuron:**
A single neuron has the input $x = 2$, weight $w = 3$, and bias $b = -1$. The activation function is ReLU. Compute the output y of the neuron.

2. **Two Neurons in Parallel:**
A layer contains two neurons. For the first neuron, the weights are $\boldsymbol{w}_1 = [1, -2]$ and bias $b_1 = 0$. For the second neuron, the weights are $\boldsymbol{w}_2 = [0, 3]$ and bias $b_2 = 1$. Input to the layer is $\boldsymbol{x} = [1, 2]$. Both neurons use ReLU activation. Compute the outputs of both neurons.

3. **Two Layers:**
A two-layer network has:

- First layer: $\boldsymbol{w}_1 = [1, 0]$, $b_1 = -1$, ReLU activation.
- Second layer: $\boldsymbol{w}_2 = [2, 3]$, $b_2 = 0$, linear activation.

Input $\boldsymbol{x} = [3, -1]$. Compute the final output.

4. **Vectorized Neuron:**
A neuron takes inputs $\boldsymbol{x} = [2, -1, 0]$, has weights $\boldsymbol{w} = [0.5, -1.5, 2]$, and bias $b = 1$. Use ReLU activation and compute the output.

Backward Pass and Gradient Descent Questions

6. **Single Neuron Gradient:**
For a neuron with $w = 1$, $b = 0$, input $x = 2$, and target output $y_{\text{true}} = 3$:

- Activation function: linear.
- Loss function: MSE $L = \frac{1}{2}(y_{\text{pred}} - y_{\text{true}})^2$.

Compute the gradient of the loss with respect to w and b.

7. **Chain Rule:**
Consider a neuron with ReLU activation and output y. The loss L depends on y as $L = \frac{1}{2}(y - 5)^2$. Input to the neuron is $x = 3$, weight $w = 2$, and bias $b = -1$. Compute $\frac{\partial L}{\partial w}$ using the chain rule.

8. **Multi-Layer Backpropagation:**
A network has two layers:

Made with Goodnotes

- First layer: $\boldsymbol{w}_1 = [1, -1]$, $b_1 = 0$, ReLU activation.
- Second layer: $\boldsymbol{w}_2 = [2]$, $b_2 = -1$, linear activation.

Input $\boldsymbol{x} = [1, 1]$, target output $y_{\text{true}} = 3$. Use MSE as the loss function and compute the gradient $\frac{\partial L}{\partial \boldsymbol{w}_1}$.

9. **Gradient Descent Update:**
For a neuron with $w = 2$, $b = 1$, input $x = 4$, and target output $y_{\text{true}} = 7$:

- Activation function: ReLU.
- Loss function: MSE.

Compute the updated w and b after one step of gradient descent with a learning rate of $\eta = 0.01$.

$$\frac{1}{\sqrt{2\pi}\cdot\sigma} \cdot e^{-\frac{1}{2}\left(\frac{(x-\mu)^2}{\sigma}\right)}$$

$$\frac{0.6 \times a}{0.6 \times a + 0.4 \times b}$$

Mechanical EM Algorithm Problems

Expectation Step (E-Step)

1. **Responsibility Calculation for 2 Components:**
A GMM has two components:

- $\mathcal{N}_1(x \mid \mu_1 = 2, \sigma_1^2 = 1)$ with $\pi_1 = 0.6$.
- $\mathcal{N}_2(x \mid \mu_2 = 4, \sigma_2^2 = 2)$ with $\pi_2 = 0.4$.

For a single data point $x_n = 3$, compute the responsibility $\gamma_{1n} = p(z_1 \mid x_n)$ symbolically. Express your answer in terms of the Gaussian density functions.

2. **Responsibilities for Multiple Data Points:**
Given the same GMM setup as in Problem 1, calculate the symbolic responsibilities γ_{1n} for the data points $x = \{2, 3, 5\}$. Leave your answers in terms of Gaussian probabilities.

Complete EM Iteration

4. **Single Iteration (Univariate GMM):**
Given a dataset $x = \{1, 3, 5\}$ and a GMM with:

- Initial means: $\mu_1 = 2, \mu_2 = 4$.

Made with Goodnotes

- Variances: $\sigma_1^2 = 1, \sigma_2^2 = 2$.
- Mixture weights: $\pi_1 = 0.5, \pi_2 = 0.5$.

Perform a single E-step by computing the responsibilities γ_{1n} and γ_{2n}. Then perform the M-step to compute the updated μ_1, μ_2, π_1, and π_2. Leave intermediate steps symbolic.

5. **Full EM Example (Multivariate Gaussian):**

For a GMM with two 2D Gaussian components, the dataset consists of three points:

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}.$$

Initial parameters are:

$$\mu_1 = [1, 1], \quad \mu_2 = [5, 5], \quad \pi_1 = \pi_2 = 0.5.$$

Covariance matrices are identity matrices ($\Sigma_1 = \Sigma_2 = I$).
Perform the E-step by calculating responsibilities, and show the symbolic updates for μ_1 and μ_2 in the M-step.

I thought it would be handy to have practice problems with a breakdown or answer key of some kind. These are AI generated, Haven't checked for validity just yet – I put subjects there so you can ctrl f

Made with Goodnotes

K-Means Clustering

Problem Set

1. **Understanding K-means**: Describe how the K-means clustering algorithm works, including the steps involved in the algorithm.
2. **Cluster Initialization**: Why is the choice of initial centroids important in K-means? Explain one method to improve the selection of initial centroids.
3. **Practical Application**:
   - You are given a dataset with the following 2D points:
     $$\{(1,1),(1.5,2),(3,4),(5,7),(3.5,5),(4.5,5),(3.5,4.5)\}$$
     Perform one iteration of K-means clustering for $K=2$. Assume initial centroids are $(1,1)$ and $(5,7)$.
4. **Choosing K**: Discuss the "elbow method" for selecting the number of clusters, and explain its advantages and limitations.
5. **Distance Metric**: In K-means, Euclidean distance is commonly used. How might the results differ if you used Manhattan distance instead?

`-means kmeans k means – I could be wrong, but I don't believe the k-means++ algorithm was mentioned in class, so I don't think question 2 is applicable to the material. Q3 has two correct answers, explained under the corresponding segment of the answer key below. Elbow method isn't mentioned either so question 4 not applicable.

Linear Regression

Problem Set

6. **Regression Line**: Given the data points $(1,2),(2,3),(3,5)$, derive the equation of the best-fit line using the least squares method.
7. **Interpretation**: Explain the difference between the coefficient and the intercept in a linear regression model.
8. **Overfitting**: How can overfitting be identified in a regression model? Suggest two methods to address it.

linear regression linreg lin-reg

Made with Goodnotes

Logistic Regression

Problem Set

9. **Sigmoid Function**: Derive the expression for the sigmoid function and explain how it transforms linear outputs into probabilities.
10. **Decision Boundary**: In logistic regression, what determines the decision boundary? How would you modify the model if the boundary is not linear?

logistic regression logreg log-reg

Dimensionality Reduction

Problem Set

11. **PCA Mechanics**: Explain the steps involved in Principal Component Analysis (PCA). Why is it important to normalize the data before applying PCA?
12. **Variance Explained**: If the eigenvalues of the covariance matrix are $3.5, 2.0, 0.5$, how many principal components would you retain to explain at least 90% of the variance?

Pca dimensionality principal component analysis eigenvalues variance

Training testing subsets matrix matrices binary classification cross-validation cross validation

Model Evaluation

Problem Set

13. **Train-Test Split**: Why is it necessary to divide the dataset into training and testing subsets? What is a common ratio used?
14. **Confusion Matrix**: Define and calculate precision, recall, and F1-score for a binary classification problem with the following confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	40	10
Actual Negative	5	45

15. **Cross-Validation**: Explain the concept of k-fold cross-validation. What are its advantages compared to a simple train-test split?

Made with Goodnotes

Answer Key

K-Means Clustering

Take w/ grain of salt!

1. **Answer**: K-means works by:
   - Initializing K centroids.
   - Assigning each data point to the nearest centroid.
   - Updating centroids by calculating the mean of assigned points.
   - Repeating steps 2 and 3 until centroids converge or max iterations reached.
2. **Answer**: Poor initialization can lead to suboptimal clustering or slow convergence. The **K-means++** algorithm improves centroid selection by choosing initial points farthest apart.
3. **Answer**:
   Assign points to clusters:
   - Cluster 1: $\{(1,1),(1.5,2)\}$
   - Cluster 2: $\{(3,4),(5,7),(3.5,5),(4.5,5),(3.5,4.5)\}$
     Update centroids:
   - Cluster 1: $(1.25,1.5)$
   - Cluster 2: $(3.9,5.1)$
4. **Answer**: The elbow method plots K vs. the sum of squared distances (SSD). The "elbow" is where SSD decreases less significantly. Limitation: Subjective and might not work well for non-spherical clusters.
5. **Answer**: Manhattan distance would lead to clusters shaped by horizontal/vertical distances, affecting centroid placement in asymmetric data.

For k-means, if a point is equidistant to multiple centroids assign randomly, so for question 3, you could arbitrarily assign to cluster 1, and get centroids (11/6,7/3) or to cluster 2, and get the two shown here.

Made with Goodnotes

Linear Regression 6. **Answer:** Equation of the best-fit line: $y = 1.5x + 0.33$.

7. **Answer:** Coefficient represents the slope (change in y for 1 unit change in x). The intercept is the y-value when $x = 0$.
8. **Answer:** Identified by high variance on the test set or low R^2. Use regularization (e.g., L1/L2 penalties) or simplify the model.

Logistic Regression 9. **Answer:** Sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$. It maps z to [0, 1], suitable for probabilities.

10. **Answer:** The decision boundary depends on the model's weights. Use polynomial features or kernel methods for non-linear boundaries.

Dimensionality Reduction 11. **Answer:** PCA steps:

- Normalize data.
- Compute covariance matrix.
- Calculate eigenvalues/eigenvectors.
- Project data onto top eigenvectors.
  Normalization ensures features contribute equally.

12. **Answer:** Variance explained: $\frac{3.5}{6} + \frac{2.0}{6} = 91.7\%$. Retain 2 components.

Model Evaluation 13. **Answer:** To assess generalization. Common ratio: 80:20 (train:test).

14. **Answer:**
    Precision: $\frac{40}{40+5} = 0.89$
    Recall: $\frac{40}{40+10} = 0.8$
    F1-score: $2 \cdot \frac{0.89 \cdot 0.8}{0.89+0.8} = 0.84$.
15. **Answer:** k-fold splits data into k subsets, rotating validation and training. Advantage: Uses all data for validation, reducing variance in results.

Info Theory

- How could you use bayes theory to create a very simple language generator?
- Could knowledge of an event increase the entropy of another event, if so give an example, otherwise, explain why not.
- Why must probabilities sum to one, and how does this

Made with Goodnotes

constraint affect entropy?

- What probability distribution is the most random?
- Describe the difference between conditional entropy $H(Y|X)$ and specific conditional entropy $H(Y|X=x)$
- Given some set of discrete values or classes calculate the optimal split.
- How do we estimate probabilities from finite samples to compute entropies and what are the risks in doing so?
- Given the following profile of a woman in the US, guess whether she is a librarian or a farmer and explain your reasoning. She wears glasses, she enjoy's reading, she is introverted.

Neural Networks

- You have a simple single-layer neural network (a perceptron) with one input neuron, one output neuron, and no hidden layers. The activation function is ReLU.

Given:

  - Input: $x=2$
  - Weight: $w=0.5$
  - Bias: $b=1$
  - Target output: $y_true=4$
  - Loss function: Mean Squared Error (MSE)
  - $\alpha=0.5$

Calculate the updated parameters of the model

- Design and justify the architecture for a neural network to determine what species of animal is shown in a 5x5 image.
- Under what conditions would you choose L1 over L2 regularization

Made with Goodnotes

GMM and EM

- write down and justify the parameters of the GMM which would model the following dataset

- estimate the covariance matrix of the following dataset and draw it's eigenvectors

Explain the concept of backpropagation in neural networks. How does it help in training the model, and what role does the learning rate play in this process?

- suppose a GMM has two gaussians with parameters
    - $\mu_1=0, \sigma_1^2=1,\pi_1=0.6$
    - $\mu_1=4, \sigma_1^2=1,\pi_1=0.4$

calculate the responsibility of each component for each point.

Made with Goodnotes