

Bayesian & Sample Complexity

\hat{h} : Selected predictor (minimizes empirical risk), \mathbf{h}_{ML} : Best possible predictor in \mathcal{H} (minimizes true risk), $\mathbf{R}(\hat{h})$: True Risk (actual error) of the selected predictor \hat{h} , $\mathbf{R}_{\mathcal{V}^m}(\hat{h})$: Empirical Risk (measured error) on validation set \mathcal{V}^m , \mathcal{H} : Hypothesis Space

1.1 Type A: Single Model Test (Hoeffding)

Recognize: "Size of test set", "Fixed prediction strategy", "Confidence interval", "Estimate $R(h)$ ".

Key: One model only (\mathcal{H} is irrelevant). Watch the **Range R** .

Example: Sequence Prediction (Hamming Loss)

Given: Predict seq. length $n = 10$. Loss = Hamming distance (sum of errors).

Range R : Loss $\in [0, n]$, so $R = n = 10$. (If "capped at 5", then $R = 5$).

a) Find Confidence δ (Failure Prob)

Params: $l = 250$ (samples), $\epsilon = 1$ (error margin).

Formula: $\delta \leq 2e^{-2l\epsilon^2/R^2}$

Calc: $\delta \leq 2 \exp\left(-\frac{2 \cdot 250 \cdot 1^2}{10^2}\right) = 2e^{-5} \approx 0.1315$

b) Find Sample Size l (Test Set)

Params: $\epsilon = 1, \delta = 0.05$, **Formula:** $l \geq \frac{R^2}{2\epsilon^2} \ln\left(\frac{2}{\delta}\right)$

Calc: $l \geq \frac{100}{2 \cdot 0.05^2} \ln(40) = 50 \cdot 3.69 \approx 185$

c) Find Error Margin ϵ

Params: $l = 20,000, \delta = 0.05$. (Assume Binary Loss $R = 1$).

Formula: $\epsilon = R \sqrt{\frac{\ln(2/\delta)}{2l}}$, **Calc:** $\epsilon = 1 \cdot \sqrt{\frac{3.69}{40,000}} \approx 0.0096$ (0.96%)

1.2 Type B: Uniform Guarantee (ULLN)

Recognize: "For every $h \in \mathcal{H}$ ", "Size of validation set" (checking all), "Uniformly".

Key: $|\mathcal{H}|$ matters. We ensure the entire list is accurate.

Example: CNN Validation (100 Epochs)

Given: $|\mathcal{H}| = 100$ classifiers, $\delta = 0.05$. Range $R = 1$.

a) Sample Size m (Uniform Interval)

Goal: Ensure $|R_{val} - R_{true}| \leq \epsilon$ for all h .

Params: $\epsilon = 0.01$.

Formula: $m \geq \frac{R^2}{2\epsilon^2} \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)$

Calc: $m \geq \frac{1}{2(0.01)^2} \ln\left(\frac{200}{0.05}\right) = 5000 \cdot \ln(4000) \approx 41,470$

1.3 Type C: Model Selection (PAC / Excess Risk)

Recognize: "Selected vs Best", "Estimation Error", " $R(\hat{h}) \leq R(h^*) + \epsilon$ ", "Size of validation set".

Key: Requires tighter precision ($\epsilon/2$) or deals with Parameter Learning.

Example 1: CNN Selection (Validation)

Goal: Selected \hat{h} is at most 1% worse than best h^* .

Params: $|\mathcal{H}| = 100, \epsilon = 0.01, \delta = 0.05$.

Formula: $m \geq \frac{2R^2}{\epsilon^2} \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)$ (Derived from Type B with $\epsilon/2$)

Calc: $m \geq \frac{2}{(0.01)^2} \ln(4000) = 20,000 \cdot 8.29 \approx 165,880$

Example 2: Fixed Tree (Parameter Learning)

Given: Input space partitioned into $M = 128$ regions. $h(x) = \sum c_r \mathbb{1}_{x \in R_r}$.

Goal: Learn Params $c \in \{-1, +1\}^M$. Estimation error $\leq \epsilon$.

a) ERM Algorithm & Formula

Algo: Majority vote in each region.

Formula: $c_r = \text{sign}\left(\sum_{i: x_i \in R_r} y_i\right)$

b) Sample Size m (Training)

Params: $\epsilon = 0.05, \delta = 0.1$.

Hypothesis Size: $|\mathcal{H}| = 2^M = 2^{128}$.

Formula: $m \geq \frac{2}{\epsilon^2} \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)$ (Std. Realizable/Agnostic Bound)

Calc: $\frac{2}{0.0025} \ln(2^{128} \cdot 2) \approx 800(3 + 88.7) \approx 73,360$

c) PAC Learnable?

YES. Finite hypothesis class (2^M). ERM on finite \mathcal{H} is always PAC.

d) Approx. Error = 0? (Multivariate Normal)

NO. Normal dists have quadratic boundaries. Tree regions are axis-aligned boxes. A box cannot perfectly fit a curve.

Parameter Estimation

2.1 Radial Exp. Distribution

Given: $p(x) = \frac{1}{2\pi} e^{-\|x-\mu\|}$ in \mathbb{R}^2 .

a) Maximum Likelihood Estimator

The MLE is $\hat{\mu}_{MLE} = \arg \max_{\mu} \prod_{i=1}^n \frac{1}{2\pi} e^{-\|x_i - \mu\|}$.

Taking the log: $\sum_{i=1}^n \left(\ln\left(\frac{1}{2\pi}\right) - \|x_i - \mu\|\right) \rightarrow \max_{\mu}$.

Ignoring constants: $\sum_{i=1}^n (\|x_i - \mu\|) \rightarrow \min_{\mu}$.

Gradient: $\nabla_{\mu} \ell = \sum_{i=1}^n \frac{x_i - \mu}{\|x_i - \mu\|}$.

Setting to zero yields the **Geometric Median** (spatial median).

Does not have a closed form solution

b) Bayes Predictor & VC Dim

The Bayes predictor is $h(x) = \arg \max_k e^{-\|x-\mu_k\|} = \arg \min_k \|x - \mu_k\|$.

Boundary: Perpendicular bisector of μ_1, μ_2 (linear).

VC Dim: Linear classifiers in \mathbb{R}^d have VC = $d + 1 = 3$.

2.2 Bernoulli Distribution (MLE & Bias)

Given: Dataset $\mathcal{T}_m = \{x_1, \dots, x_m\}$ where $x_i \in \{0, 1\}$.

Model: $p(x) = \beta^x (1 - \beta)^{1-x}$ (Bernoulli)

a) MLE Estimation, Likelihood: $L(\beta) = \prod_{i=1}^m \beta^{x_i} (1 - \beta)^{1-x_i}$

Log-Likelihood: $\ell(\beta) = \sum_{i=1}^m (x_i \ln \beta + (1 - x_i) \ln(1 - \beta))$

Derivative: $\frac{\partial \ell}{\partial \beta} = \frac{\sum x_i}{\beta} - \frac{m - \sum x_i}{1 - \beta}$

Optimal β : Set $\frac{\partial \ell}{\partial \beta} = 0 \Rightarrow \hat{\beta}_{MLE} = \frac{1}{m} \sum_{i=1}^m x_i$ (Sample Mean).

b) Bias Check: Task: Determine if unbiased, i.e., check if $\mathbb{E}[\hat{\beta}] = \beta$.

Proof: $\mathbb{E}[\hat{\beta}] = \mathbb{E}\left[\frac{1}{m} \sum x_i\right] = \frac{1}{m} \sum \mathbb{E}[x_i]$.

Since $x_i \sim \text{Bernoulli}(\beta)$, we know $\mathbb{E}[x_i] = \beta \Rightarrow \mathbb{E}[\hat{\beta}] = \frac{1}{m} (m \cdot \beta) = \beta$.

Result: The estimator is **Unbiased**.

c) Exponential Family Form

Rewrite: $p(x) = \exp(x \ln \frac{\beta}{1-\beta} + \ln(1-\beta)) = \frac{1}{Z}$

Match: $\exp(\theta \phi(x) - A(\theta))$.

Params: $\phi(x) = x, h(x) = 1, \theta = \ln \frac{\beta}{1-\beta}, A(\theta) = \ln(1 + e^\theta)$.

2.3 Gaussian KL Approx

Task: Find μ, σ for $q(x)$ to approx fixed $p(x)$ (mean μ_0 , var σ_0^2) by min $D_{KL}(p||q)$.

Logic: Minimizing $D_{KL}(p||q)$ is equivalent to maximizing $\mathbb{E}_p[\ln q(x)]$. For Gaussian q , this requires moment matching.

Result: $\mu = \mu_0, \sigma = \sigma_0$

2.4 1-NN Regression: Bias-Variance Decomposition

Given: Regression problem $y = f(x) + \epsilon$, noise $\epsilon \sim (0, \sigma^2)$.

Model: 1-Nearest Neighbor $h_m(x) = y_{n(x)} = f(x_{n(x)}) + \epsilon_{n(x)}$, where $n(x)$ is the index of the nearest neighbor.

Assumption: Fixed design (inputs x_i are fixed/deterministic, randomness is only in ϵ).

a) Expected Predictor $g_m(x)$

Def: $g_m(x) = E_{\mathcal{T}^m} [h_m(x)]$

Calc: $E_{\mathcal{T}^m} [f(x_{n(x)}) + \epsilon_{n(x)}] = f(x_{n(x)}) + E[\epsilon] = f(x_{n(x)})$.

Note: $x_{n(x)}$ is fixed, so $f(x_{n(x)})$ is constant.

b) Squared Bias $(E_x[(g_m(x) - f(x))^2])$

Result: $E_x[(f(x_{n(x)}) - f(x))^2]$.

Interpretation: Bias depends entirely on how far the nearest neighbor is from the query point x . If $x_{n(x)} \approx x$ (dense data), bias is small.

c) Variance $(\text{Var}_{\mathcal{T}^m}(h_m(x)))$

Def: $E[(h_m(x) - g_m(x))^2] + \text{Var}(f(x_{n(x)}) + \epsilon_{n(x)})$

Calc: $\text{Var}(\text{const} + \epsilon) = \text{Var}(\epsilon) = \sigma^2$.

Result: σ^2

Insight: 1-NN does not average data (unlike k-NN), so it does not reduce noise variance. The error variance equals the irreducible noise.

2.5 Sequence Boundary Detection

Given: Sequence x_1, \dots, x_n with changepoint k .

Model: $x_{1:k} \sim \mathcal{N}(\mu_1, \sigma_1^2), x_{k+1:n} \sim \mathcal{N}(\mu_2, \sigma_2^2)$, prior $p(k) = \pi_k$.

Task 1: MLE with observed boundaries (x^j, k^j) .

$\pi_k = \frac{\text{count}(k=j)}{m}, \mu_1 = \text{avg}(x_i^j \text{ where } i \leq k_j), \mu_2 = \text{avg}(x_i^j \text{ where } i > k_j)$

$\sigma_r^2 = \frac{1}{N_r} \sum (x_i^j - \mu_r)^2$ for $r = 1 (i \leq k_j)$ and $r = 2 (i > k_j)$

Task 2: Optimal k^* under quadratic loss $\ell = (k - k^*)^2$.

Bayes optimal = posterior mean: $k^* = \mathbb{E}[k|x] = \sum_{k=0}^n k \cdot p(k|x)$

where $p(k|x) \propto \pi_k \prod_{i=1}^k \mathcal{N}(x_i|\mu_1, \sigma_1^2) \prod_{i=k+1}^n \mathcal{N}(x_i|\mu_2, \sigma_2^2)$

2.6 Generative Gaussian vs Linear Classifier

Given: Class-conditional $p(x|y) = \mathcal{N}(\mu_y, \Sigma_y)$ (Multivariate Normal), hypothesis \mathcal{H} : linear classifiers, algo: ERM.

a) Approx. error if $C_+ = C_- = ?$ (LDA)

Log-ratio $\ln \frac{p(x|+)}{p(x|-)}$ cancels quadratic terms \rightarrow boundary is linear \rightarrow **Result:** $\epsilon_{app} = 0$ (ϵ_{app} is the err of best possible class. - bayesian)

b) Approx. error if $C_+ \neq C_-$? (QDA)
Quadratic terms don't cancel \rightarrow boundary is quadratic \rightarrow **Result:** $\epsilon_{app} > 0$

c) Statistically consistent? Result: No (model misspec) ERM converges to best linear $h_{\mathcal{H}}$, but $R(h_{\mathcal{H}}) > R(h^*)$ when $C_+ \neq C_-$.

d) Pac Learnable? Is hypothesis space finite \rightarrow **YES** Is VC dim finite \rightarrow **YES** (hypothesis space CAN then be infinite)

2.7 Poisson Exp. Family & MLE

Task: Show Poisson $p(k) = \frac{\lambda^k e^{-\lambda}}{k!}$ is Exp. Family. Find MLE for natural param η given avg \bar{k} .

a) Exp. Family Form (Rewrite via $\lambda^k = e^{k \ln \lambda}$):

$p(k) = \frac{1}{k!} \exp[k \log \lambda - \lambda] = h(k) \exp[\eta \theta - A(\eta)]$

Natural param: $\eta = \log \lambda$ Log-partition: $A(\eta) = e^\eta$ Sufficient Statistic: $T(k) = k$ Base: $h(k) = \frac{1}{k!}$

b) MLE Derivation (Maximize log-likelihood):

$\mathcal{L}(\eta) = \sum_{i=1}^n \log p(k_i|\eta), \log p(k_i|\eta) = \log(h(k_i)) + \eta k_i - e^\eta$

$A(\eta) \stackrel{\text{wrt } \eta}{\rightarrow} \sum_{i=1}^n (\eta k_i - A(\eta))$ Objective: $\eta \bar{k} - A(\eta) \rightarrow \max_{\eta}$

Differentiate: $\frac{\partial \mathcal{L}}{\partial \eta} (\eta \bar{k} - e^\eta) = \bar{k} - e^\eta = 0$, **Solution:** $\eta = \log \bar{k}$ **b2)**

MLE Derivation

(Maximize likelihood): $L(\lambda) = \prod_{j=1}^m \frac{\lambda^{x_j} e^{-\lambda}}{x_j!} \Rightarrow \mathcal{L}(\lambda) = \ln L(\lambda) = \sum_{j=1}^m (x_j \ln \lambda - \lambda - \ln x_j!)$.

Differentiate: $\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{j=1}^m \left(\frac{x_j}{\lambda} - 1\right) = \frac{1}{\lambda} \sum_{j=1}^m x_j - m = 0$.

Solution: $\hat{\lambda} = \frac{1}{m} \sum_{j=1}^m x_j$.

c) Compute Fisher Information

(assuming mean equals variance) 1. log-likelihood of one element: $\ln p(x) = x \ln \lambda - \lambda - \ln(x!)$

2. first derivation: $\frac{\partial \mathcal{L}}{\partial \lambda} \ln p(x; \lambda) = \frac{x}{\lambda} - 1$

3. second derivation: $\frac{\partial^2}{\partial \lambda^2} \ln p(x; \lambda) = -\frac{x}{\lambda^2}$

4. fisher information: $I(\lambda) = -\mathbb{E}\left[\frac{\partial^2}{\partial \lambda^2} \ln p(X; \lambda)\right] = \frac{1}{\lambda}$

d) Compute Minimum Sample Count

so that $\epsilon < 0.1, \delta > 0.99$: 1. Asymptotic Normality: $\hat{\lambda}_{MLE} \sim \mathcal{N}\left(\lambda, \frac{1}{m I(\lambda)}\right) = \mathcal{N}\left(\lambda, \frac{\lambda}{m}\right)$

2. Confidence Interval: $P(|\hat{\lambda} - \lambda| \leq \epsilon) \geq 0.99 \Rightarrow \sqrt{\frac{\epsilon}{\lambda}} \geq 2.576$

3. Solve for m : $\sqrt{m} \geq \frac{2.576 \sqrt{\lambda}}{\epsilon} \Rightarrow m \geq \lambda \left(\frac{2.576}{\epsilon}\right)^2$

4. Worst Case Assumption: $\lambda < 1 \Rightarrow \max(\lambda, 1) = 1$

5. Calculation: $m \geq 1 \cdot \left(\frac{2.576}{0.1}\right)^2 \approx 663.57 \Rightarrow m = 664$

Neural Networks

3.1 Backpropagation

Given: $h = \tanh(s), a_j = \text{relu}(z_j)$, Loss $\ell = \frac{1}{2} (y - h)^2$.

Hints: $\tanh' = 1 - \tanh^2(s), \text{relu}(s) = \max(0, s)$.

Derivation:

1. **Output Error Signal:** $\delta_{out} = \frac{\partial \ell}{\partial h} \cdot \frac{\partial h}{\partial s} = -(y - h)(1 - \tanh^2(s))$

2. **Layer 2 ($w_j^{(2)}$):** Input is a_j .

$-\frac{\partial \ell}{\partial w_j^{(2)}} = \delta_{out} \cdot a_j = -(y - h)(1 - \tanh^2(s)) \cdot \text{relu}(z_j)$

3. **Layer 1 ($w_{ij}^{(1)}$):** Backprop error through weights $w_j^{(2)}$ and activation.

(Note: Deriv of $\text{relu}(z_j)$ is 1 if $z_j > 0$, else 0).

$-\frac{\partial \ell}{\partial w_{ij}^{(1)}} = -(y - h)(1 - \tanh^2(s)) \cdot w_j^{(2)} \cdot \begin{cases} 1 & \text{if } z_j > 0 \\ 0 & \text{else} \end{cases} \cdot x_i$

3.2 Median Pooling (2x2 Backward)

Given: Window size 2 \times 2 (4 even). Median is avg of two middle sorted vals $x_{(2)}, x_{(3)}, f_{k1} = 0.5(x_{(2)} + x_{(3)})$.

Gradient: Splits between the middle elements. $\frac{\partial f}{\partial x_{ij}} = \begin{cases} 0.5 & \text{if } x_{ij} \in \{x_{(2)}, x_{(3)}\} \\ 0 & \text{else} \end{cases}$

3.3 Neural Module (Linear + ELU)

Task: Define module with n inputs, k linear+ELU units.

ELU: $f(z) = z$ if $z > 0$, else $e^z - 1$. Deriv: $f'(z) = 1$ if $z > 0$, else e^z .

Forward Message (layer outputs as function of inputs): $z_k = \sum_{j=1}^m w_{kj} x_j + b_k, y_k = f(z_k)$ for $k = 1, \dots, K$.

Backward Message (derivs of all outputs w.r.t. all inputs):

$\frac{\partial y_k}{\partial x_j} = \frac{\partial y_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial x_j} = f'(z_k) \cdot w_{kj}$ for all k, j .

Parameter Message (derivs of all outputs w.r.t. all params):

$\frac{\partial y_k}{\partial w_{kj}} = f'(z_k) \cdot x_j, \frac{\partial y_k}{\partial b_k} = f'(z_k)$ for all k, j .

3.4 Backprop (Sigmoid & BCE)

Given: Input $x \in \mathbb{R}^n$, target $y \in \{0, 1\}$, weights $w \in \mathbb{R}^n$.

Forward: $s = \sum w_i x_i, \hat{y} = \sigma(s) = \frac{1}{1 + e^{-s}}$ (network output).

Loss (BCE): $\mathcal{L} = -y \$