

Advanced Machine Learning

Silvan Stadelmann - 3. Februar 2026 - v0.1.1

github.com/silvasta/summary-aml



Created with Grok and Gemini

Sigm. $\sigma(x) = \frac{1}{1+e^{-x}}$, $\sigma'(x) = \sigma(x)(1-\sigma(x))$

Variance $\mathbb{E}[(\hat{y} - \mathbb{E}[\hat{y}])^2]$

Loss Functions

Logistic $\ell(y, p) = -y \log p - (1-y) \log(1-p)$

Cross-Entropy loss $\mathcal{L} = -\sum_i y_i \log(\hat{y}_i)$

1 Representations

1.1 Empirical Risk Minimization (ERM)

$R(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(f(x), y)]$ \mathcal{D} data distrib.

Empirical Risk: $\hat{R}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$

CRLB Lower bound on variance of unbiased estimators $\hat{\theta}$ of param θ . Assumes regularity: differentiable log-likelihood, finite variance.

Trick CRLB achieved iff estimator is efficient, Multivariate: Use inverse Fisher matrix.

Hint for exams Always check unbiasedness first; compute via Hessian or score function.

1.2 Formulas: Fisher Information

$$\mathbb{E} \left[\left(\frac{\partial}{\partial \theta} \log p(X|\theta) \right)^2 \right] = -\mathbb{E} \left[\frac{\partial^2}{\partial \theta^2} \log p(X|\theta) \right]$$

Multivariate ($\theta \in \mathbb{R}^k$): Matrix form

$$[I(\theta)]_{ij} = \mathbb{E} \left[\frac{\partial \log p}{\partial \theta_i} \frac{\partial \log p}{\partial \theta_j} \right] = -\mathbb{E} \left[\frac{\partial^2 \log p}{\partial \theta_i \partial \theta_j} \right]$$

Trick: For iid X_1, \dots, X_n , $I_n(\theta) = nI(\theta)$

Example: Gaussian $\mathcal{N}(\mu, \sigma^2 = 1)$

Score: $\frac{\partial \log p}{\partial \mu} = x - \mu$. Fisher: $I(\mu) = 1$.

1.3 Rao-Cramér Lower Bound (CRLB)

For unbiased $\hat{\theta}(X)$: $\text{Var}(\hat{\theta}) \geq \frac{1}{I(\theta)}$ (scalar)

Multiv. $\text{Var}(g(\hat{\theta})) \geq \left(\frac{\partial g}{\partial \theta} \right)^T I(\theta)^{-1} \left(\frac{\partial g}{\partial \theta} \right)$

General CRLB: $\text{Cov}(\hat{\theta}) \succeq I(\theta)^{-1}$

Equality if $\hat{\theta} = a(\theta) \cdot s(X) + b(\theta)$, $s(X)$ is suff. stat. (Gauss. sample mean givs CRLB)

1.4 Calculus Recipes & Derivations

- Compute $I(\theta)$: (1) Write $\log L(\theta|X) = \sum \log p(x_i|\theta)$. (2) Take 2nd deriv or score sq. (3) Expectation over $p(X|\theta)$.
- For representations: Info in feature space: $I_\phi(\theta) = \mathbb{E}[\phi(X)^T \phi(X)]^{-1}$
- Exam hint: CRLB bounds learning rates (e.g., variance in param est. for neural nets).

2 Gaussian Processes (GPs)

Distribution over functions $f(\mathbf{x})$, defined by mean function $m(\mathbf{x}) = 0$ (zero-mean prior) and covariance (kernel) function $k(\mathbf{x}, \mathbf{x}')$.

For any finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$, $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Intuition: GPs sample smooth functions; non-parametric, infinite-dimensional Bayesian linear regression. Prior: Multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = \mathbf{0}$, $\boldsymbol{\Sigma} = \mathbf{K} + \sigma_n^2 \mathbf{I}$ (noisy) Posterior: $\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$.

$$\text{Joint: } \begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{k}_* \\ \mathbf{k}_*^T & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right)$$

$$f_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(\mu_*, \sigma_*^2), \mu_* = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*$$

3 Ensemble Methods

3.1 Bagging (Bootstrap Aggregating)

Average B models: $\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B \hat{f}_b(\mathbf{x})$

Reduction: $\text{Var}(\hat{f}) \approx \frac{1}{B} \text{Var}$ if uncorrelated

Random Forest: Bagging + random feature

Importance $I(f) = \sum_{\text{nodes}} \Delta \text{impurity} \cdot p(\text{node})$

3.2 Boosting

Sequential, weight misclassified points.

Final: $H(\mathbf{x}) = \text{sign}(\sum_m \alpha_m h_m(\mathbf{x}))$,

$\alpha_m = \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}$. Reduces bias.

AdaBoost

Weights $w_i^{(t+1)} = w_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i))$,

normalized. $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$, where $\epsilon_t =$

weighted error.

Error bound: $\epsilon \leq 2^M \prod_m \sqrt{\epsilon_m(1-\epsilon_m)}$.

Gradient Boosting min $L = \sum_i \ell(y_i, F(\mathbf{x}_i))$, update $F_m = F_{m-1} + \nu h_m$, where h_m fits pseudo-residuals $r_{im} = -\frac{\partial \ell}{\partial F_{m-1}(\mathbf{x}_i)}$.

4 Support Vector Machines (SVMs)

4.1 Hard-Margin SVM (Linearly Separable)

Primal: min $\frac{1}{2} \|\mathbf{w}\|^2$ s.t. $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \forall i$
Dual: max $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ s.t. $\alpha_i \geq 0, \sum_i \alpha_i y_i = 0$ Margin: $\gamma = \frac{2}{\|\mathbf{w}\|}$

Decision: $f(\mathbf{x}) = \text{sign}(\sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b)$

Support vectors: Points w. $\alpha_i > 0$ (on margin)

4.2 Soft-Margin SVM

Primal: min $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i$
s.t. $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$

Hinge loss: $\ell(y, \hat{y}) = \max(0, 1 - y\hat{y})$.

Dual: Same as hard-margin but $0 \leq \alpha_i \leq C$.

C trades bias/var. large $C \rightarrow$ hard-margin

Kernel Trick Replace $\mathbf{x}_i^T \mathbf{x}_j$ with $k(\mathbf{x}_i, \mathbf{x}_j)$

• Polynomial: $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + c)^d$

• RBF: $k(\mathbf{x}, \mathbf{z}) = \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2} \right)$

• Matérn: $k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right)$
($\nu = 3/2$ or $5/2$ for smoothness).

Mercer's condition Kernel matrix ≥ 0 (PSD)

5 Neural Networks: Basics

5.1 Propagation

Forward: $z^l = W^l a^{l-1} + b^l, a^l = \sigma(z^l)$

Backward: $\delta^L = \nabla_a L \odot \sigma'(z^L), \delta^l = (W^{l+1})^T \delta^{l+1} \odot \sigma'(z^l)$

Weight update: $\frac{\partial L}{\partial W^l} = \delta^l (a^{l-1})^T$

6 Attention Mechanisms

Attention(Q, K, V) = $\text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$

Q/K/V: Linear projections of input. Scaled for stability (prevents large dot-products).

6.1 Multi-Head Attention

Concat(head₁, ..., head_h) W^O , head_i =

Attention(QW_i^Q, KW_i^K, VW_i^V) - h heads

project to subspaces (e.g., h=8). - Advantage:

Captures multiple dependency types.

7 Transformers

Architecture: Encoder (self-attn + FFN) stack; Decoder (masked self-attn + enc-dec attn + FFN) stack. - FFN: Two linear layers with ReLU: FFN(\mathbf{x}) = $\max(0, \mathbf{x}W_1 + \mathbf{b}_1)W_2 + \mathbf{b}_2$. - Residual: $\mathbf{x} \leftarrow \mathbf{x} + \text{Sublayer}(\mathbf{x})$. LayerNorm after.

Positional Encoding Add to input embeddings.

$$\text{PE}_{(\text{pos}, 2i|+1)} = \sin \left| \cos \left(\frac{\text{pos}}{10000^{2i/d}} \right) \right|$$

Allows order awareness; fixed or learned.

8 Computer Vision

8.1 Convolutional Neural Networks (CNNs)

Discrete Convolution (2D)

Input I : (H, W, C) , Kernel K : $(k_h \times k_w \times C)$

$O[i, j] = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} \sum_{c=1}^C I[i+m, j+n, c] \cdot K[m, n, c] + b$, p = padding, s = stride:
Output size: $\lfloor (H - k_h + 2p)/s \rfloor + 1$,

Pooling (Max/Avg): Reduces dims, e.g., max-pool: $O[i, j] = \max_{m,n} I[i \cdot s + m, j \cdot s + n]$.

Backpropagation in CNNs: Gradients via chain rule. For conv layer: - Weight grad: $\frac{\partial \mathcal{L}}{\partial K[m, n, c]} = \sum_{i,j} \frac{\partial \mathcal{L}}{\partial O[i, j]} \cdot I[i+m, j+n, c]$. - Input grad: Rotate kernel 180° and convolve with output grad.

9 Graph Neural Networks (GNNs)

9.1 Basics & Notation

Graph $G = (V, E)$, $|V| = n$ nodes, adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ (symmetric for undirected). Feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ (node features). Degree matrix $\mathbf{D} = \text{diad}(\sum_j A_{ij})$.

Normalized adjacency: $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ (self-loops), $\hat{\mathbf{A}} = \mathbf{D}^{-1/2} \tilde{\mathbf{A}} \mathbf{D}^{-1/2}$ (symmetric normalization).

Message passing: Update node v as $h_v^{(l+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} m_{u \rightarrow v}^{(l)} \right)$, where m aggregates neighbor info.

9.2 Graph Convolutional Network (GCN)

Layer: $\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{A}} \mathbf{H}^{(l)} W^{(l)})$, with $\mathbf{H}^{(0)} = \mathbf{X}$.

Spectral view: Approximation of graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, normalized $\hat{\mathbf{L}} = \mathbf{I} - \hat{\mathbf{A}}$.

<p>9.3 Graph Attention Network (GAT)</p> <p>Attention: $\alpha_{ij} = \text{softmax}_j (\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}h_i \parallel \mathbf{W}h_j]))$</p> <p>Update: $h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i) \cup i} \alpha_{ij} \mathbf{W}h_j^{(l)} \right)$.</p> <p>Multi-head: Concat or average heads.</p> <p>10 Information Theory</p> <p>10.1 Key Measures</p> <p>Entropy: $H(X) = -\mathbb{E}_{p(x)}[\log p(x)]$</p> <p>Joint entropy: $H(X, Y) = -\mathbb{E}[\log p(x, y)]$.</p> <p>Conditional: $H(Y X) = H(X, Y) - H(X)$.</p> <p>Mutual information: $I(X; Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X Y) = \text{KL}(p(x, y) \parallel p(x)p(y)) \geq 0$.</p> <p>Cross-entropy: $H(p, q) = -\mathbb{E}_p[\log q] = H(p) + \text{KL}(p \parallel q)$.</p> <p>KL divergence $\text{KL}(p \parallel q) = \mathbb{E}_p[\log(p/q)] \geq 0$</p> <p>Tricks: Jensen-Shannon divergence for stability: $\text{JSD}(p \parallel q) = \frac{1}{2} \text{KL}(p \parallel m) + \frac{1}{2} \text{KL}(q \parallel m)$, $m = (p + q)/2$.</p> <p>11 Anomaly Detection</p> <p>11.1 Statistical Methods</p> <p>Z-Score: Score $z_i = \frac{x_i - \mu}{\sigma}$. Anomaly if $z_i > \theta$</p> <p>Mahalanobis Distance Accounts for cov. $D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$</p> <p>Anomaly if $D_M > \theta$ (e.g., from χ^2 dist.).</p> <p>11.2 Proximity-Based Methods</p> <p>11.3 Isolation Forest</p> <p>Randomly partition until isolation. Anomaly Score: $s(\mathbf{x}, n) = 2^{-\frac{E(h(\mathbf{x}))}{c(n)}}$, where $h(\mathbf{x})$ = path length, $E(\cdot)$ = avg over trees, $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$ (H = harmonic number). Anomaly if $s \approx 0.5$ (normal) or $s \rightarrow 1$ (anomaly). Works well in high dims.</p> <p>11.4 One-Class SVM</p> <p>Hyperplane maximizing margin from origin $\min_{\mathbf{w}, \xi_i, \rho} \frac{1}{2} \ \mathbf{w}\ ^2 + \frac{1}{\nu n} \sum_i \xi_i - \rho$</p> <p>s.t. $\mathbf{w}^T \phi(\mathbf{x}_i) \geq \rho - \xi_i$</p> <p>Decision: $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) - \rho)$. $\nu \in (0, 1]$</p>	<p>12 RL & Active Learning</p> <p>12.1 Markov Decision Processes (MDPs)</p> <p>$V^\pi(s) = \mathbb{E}_\pi [\sum_{t=0}^\infty \gamma^t R(s_t, a_t) \mid s_0 = s]$</p> <p>Action-value $Q^\pi(s, a) = \mathbb{E}_\pi [\sum_{t=0}^\infty \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a]$</p> <p>Advantage $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$.</p> <p>Bellman Optimality: $V^*(s) = \max_a \sum_{s', r} P(s', r \mid s, a)[r + \gamma V^*(s')]$</p> <p>Discounted Return: $G_t = \sum_{k=t}^\infty \gamma^{k-t} R_{k+1}$.</p> <p>Exploration ϵ-greedy (random w.p. ϵ), UCB ($a = \arg \max [Q(s, a) + c \sqrt{\frac{\ln t}{N(s, a)}}]$)</p> <p>Policy Gradient Thm: $\nabla_\theta J(\theta) = \mathbb{E}_\pi [\nabla_\theta \log \pi_\theta(a \mid s) Q^\pi(s, a)]$</p> <p>REINFORCE: $\hat{\nabla} J = \sum_t \nabla_\theta \log \pi(a_t \mid s_t) G_t$. Var. reduct. Subtract baseline $b(s_t) \approx V(s_t)$</p> <p>Actor-Critic: $A = r + \gamma V(s') - V(s)$.</p> <p>13 Reproducing Kernel Hilbert Spaces (RKHS)</p> <p>A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space \mathcal{H} of functions $f : \mathcal{X} \rightarrow \mathbb{R}$ with a kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that:</p> <ul style="list-style-type: none"> k is positive semi-definite (PSD): For any x_i, the Gram matrix $K_{ij} = k(x_i, x_j) \succeq 0$. Reproducing $\langle f, k(x, \cdot) \rangle_{\mathcal{H}} = f(x) \forall f \in \mathcal{H}$. <p>Counterfactual invariance In causal ML, models invariant under interventions (e.g., do-calculus). For a structural causal model (SCM) $Y = f(X, U)$, counterfactuals ask "What if?" (e.g., $Y_{x'}$ where x' is intervened). Invariance ensures predictions stable across envs.</p> <p>Moore-Aronszajn Every PSD kernel k defines a unique RKHS where $\text{span}\{k(x, \cdot)\}$ is dense.</p> <p>Mercer for continuous PSD kernels on compact \mathcal{X}, $k(x, x') = \sum_{i=1}^\infty \lambda_i \phi_i(x) \phi_i(x')$, with $\lambda_i \geq 0$, enabling eigen-decomposition.</p> <p>SVMs K decision $f(x) = \sum \alpha_i y_i k(x_i, x) + b$</p> <p>GP Prior $f \sim \mathcal{GP}(m, k)$ in RKHS, posterior mean $\bar{f}(x_*) = k_*^\top (K + \sigma^2 I)^{-1} y$</p> <p>Counterfactuals in ML: Use invariant risk min-</p>	<p>imization (IRM) to minimize risk invariant to spurious correlations (e.g., Arjovsky et al.).</p> <p>Solve Use reproducing property for f evaluation: $f(x) = \langle f, k(x, \cdot) \rangle$ Show PSD via Mercer.</p> <p>Counterfactual Trick: For invariance, compute $P(Y do(X = x'))$ using causal graphs; compare to observational $P(Y X)$.</p> <p>14 Variational Autoencoders (VAEs)</p> <p>14.1 Evidence Lower Bound (ELBO)</p> <p>$\mathbb{E}_{q_\phi(z x)}[\log p_\theta(x z)] - \text{KL}(q_\phi(z x) \parallel p(z))$</p> <p>15 Non-Parametric Bayesian Methods</p> <p>15.1 Dirichlet Processes & Non-Param. Bayes</p> <p>$\sum_{k=1}^\infty \pi_k \delta_{\theta_k}, \pi_k = \beta_k \prod_{j=1}^{k-1} (1 - \beta_j)$</p> <p>16 PAC Learning</p> <ul style="list-style-type: none"> Realizable $\exists h^* \in \mathcal{H}$ with true risk $L(h^*) = 0$. PAC Learnable: \exists learner s.t. \forall distributions $\mathcal{D}, \forall \epsilon, \delta > 0$, with prob. $\geq 1 - \delta$, outputs h with $L(h) \leq \epsilon$ using $m = m(\epsilon, \delta)$ samples. Agnostic PAC: No assumption on h^*; minimize excess risk over \mathcal{H}. True Risk: $L(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)]$ (e.g., 0-1 loss: $\ell = \mathbf{1}_{h(x) \neq y}$). Empirical Risk: $\hat{L}(h) = \frac{1}{m} \sum_{i=1}^m \ell(h(x_i), y_i)$ <p>16.1 VC Dimension & Shattering</p> <ul style="list-style-type: none"> Shattering: \mathcal{H} shatters set $S \subseteq \mathcal{X}$ if $\{\mathbf{y} \in \{0, 1\}^{ S } : \exists h \in \mathcal{H} \text{ realizes } \mathbf{y} \text{ on } S\} = 2^{ S }$. Growth Function: $\Pi_{\mathcal{H}}(m) = \max_{S: S =m} \{h _S : h \in \mathcal{H}\} \leq \left(\frac{em}{d}\right)^d$ (Sauer-Shelah, if VC-dim $d < \infty$). VC Dimension $d = \text{VC}(\mathcal{H})$: Largest S s.t. \mathcal{H} shatters S (infinite if no such max). Trick for VC Calc: Find largest shatterable set (e.g., for half-planes: 3 points not collinear shatter, 4 do not). <p>Fundamental Thm of PAC (Realizable, Finite \mathcal{H}): $m \geq \frac{1}{\epsilon} (\ln \mathcal{H} + \ln(1/\delta))$ samples suffice for $L(h) \leq \epsilon$ w.p. $\geq 1 - \delta$ via ERM.</p> <p>Infinite \mathcal{H} (VC-based): For VC-dim d, $m \geq C \frac{d + \ln(1/\delta)}{\epsilon}$ (lower bound); upper: $m =$</p>	<p>$O\left(\frac{d \ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}\right)$.</p> <p>Agnostic PAC (Uniform Convergence): w.p. $\geq 1 - \delta$, $L(h) - \hat{L}(h) \leq \sqrt{\frac{2d \ln(em/d) + \ln(2/\delta)}{m}}$</p> <p>Sample Complexity (Agnostic): $m = O\left(\frac{d \ln(1/\epsilon) + \ln(1/\delta)}{\epsilon^2}\right)$ for excess risk $\leq \epsilon$.</p> <p>Tricks:</p> <ul style="list-style-type: none"> Use Hoeffding for finite \mathcal{H}: $\Pr(L - \hat{L} > \epsilon) \leq 2 \mathcal{H} e^{-2m\epsilon^2}$. For VC, bound $\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^d \binom{m}{i} \leq (em/d)^d$. ERM is PAC if \mathcal{H} has finite VC-dim.
--	---	--	---