1 MACHINE LEARNING Midterm Review

1.1 History of AI

- Turing test, 1950: Let computers do **imitation game**; no longer used for testing AI.

- First "Artificial Neural Network", 1951, Marvin Minsky: **SNARC**.

- Name "Aritificial Intelligence" is given in 1956 by John McCarthy. **Dartmouth Conference** marks the BIRTH of AI, as the name was accepted.

- **First AI Winter (1974-1980)**: Funding for AI disappeared. Reason: low computing power, logic reasoning is not powerful, search space explosion.

- **AI Boom (1980-1987)**: Expert System; Backprop; Hopfield Net.

- **Second AI Winter (1987-1993)**: Expert systems are domain-specific.

- 1993-2011: **steady development**, e.g. Deep Blue. Methods: SVM, RL.

- **Era of DL (2012-2022)**, AlexNet; AlphaGo. **Foundation Models (2018-)**, CLIP; ChatGPT; Copilot.

1.2 Framework

- **Empirical Distribution**: $P((x_i, y_i)) = 1/N$.

- Training, validation, test set are all sampled from **population distribution** D. The ultimate goal in SL is minimize $L_{\text{population}}$.

- **Memorization**: $f(x_i) = y_i$, and random otherwise. Achieves 0 training loss, but useless.

- **Pipeline of SL**: Identify Task $\rightarrow$ Create Dataset $\rightarrow$ Define Loss $\rightarrow$ Learn $f \rightarrow$ Generalization

- Overfitting; Underfitting. **Regularization**: CLASSICAL VIEW: restrict representation power of f. MODERN VIEW: Sometimes explicit regularization is not necessary, because there are implicit ones (SGD).

- Create Dataset: **Mechanical Turk**: crowd sourcing (split to micro-tasks, take majority voting, assembly line). Sometimes workers are strategic, this is the main bottleneck in practice.

- **UnS L**: learning distribution of X. **Clustering**, **Principle Component Analysis (PCA)**, **Generative Model** (maps Gaussian to the target distribution, also learn the hidden structure of X), **Anomaly Detection**, **Dimension Reduction**.

- **SemiS L**: Need assumptions (**Continuity Assumption**: close points share a label; simple decision boundaries).

- **Optimization**: *Zeroth-order method*: Hyperparameter tuning, when function is not differentiable; *Second-order method*: Hessian matrix (time-consuming) [NOTE: 1.5-th order method: BFGS, Hessian-vector Product]; **GD**.

1.3 Optimization

1.3.1 GD

- Smoothness Assumption: Gradient is Lipschitz. Two equivalent statements: $||\nabla f(w) - \nabla f(w')|| \leq L||w - w'||$; $|f(w') - f(w) - \langle \nabla f(w), w' - w \rangle| \leq \frac{L}{2}||w - w'||^2$.

- $f(w') - f(w) \leq -\eta(1 - L\eta/2)||\nabla f(w)||^2$. So, set $\eta < 2/L$ to make sure $f(w') < f(w)$.

- Strongly Convex: $|f(w') - f(w) - \langle \nabla f(w), w' - w \rangle| \geq \frac{\mu}{2}||w - w'||^2$.

- If L-smooth and convex,

$$f(w_t) \leq f(w^\star) + \frac{||w_0 - w^\star||^2}{2\eta t},$$

so we only need $T = \frac{L}{2\epsilon} \cdot ||w_0 - w^\star||^2$ to find $f(w_T) \leq f(w^\star) + \epsilon$.

1.3.2 SGD

$w_{t+1} = w_t - \eta G_t$, $E[G_t] = \nabla L(w_t)$. We can set

$$G_t = \frac{1}{|S|} \sum_{i \in S} \nabla l(\theta, x_i, y_i).$$

S is **mini-batch**. $[N]$ is **full-batch**, or **batch**.

- L-smooth, convex, $\mathrm{Var}(G_t) \le \sigma^2$, $\eta \le 1/L$, then

$$E[f(\bar{w}_t)] \le f(w^\star) + \frac{||w_0 - w^\star||^2}{2\eta t} + \eta\sigma^2.$$

In this situation, SGD converges at rate $\frac{1}{\sqrt{t}}$, when t, η is set properly according to σ, ϵ. (Comparison: GD is $1/t$.)

- If assume f is also **strongly convex**, GD is linear convergence, SGD with rate $1/t$.

1.3.3 SVRG

Update rule: in round s, repeat m steps,

$$w_t = w_{t-1} - \eta \left(\nabla l_i(w_{t-1}) - \nabla l_i(\tilde{w}^{s-1}) + \tilde{u} \right),$$

i is randomly picked. At the end of the round, set $\tilde{w}^s = w_t$ where t is randomly chosen in $\{0, 1, \cdots, m-1\}$.

Assume L-smooth, u-strongly convex,

$$\frac{E[f(\tilde{w}^s) - f(w^\star)]}{E[f(\tilde{w}^{s-1}) - f(w^\star)]} \le \frac{1}{u\eta(1 - 2L\eta)m} + \frac{2L\eta}{1 - 2L\eta}.$$

Choose a η to let RHS < 1, achieving linear rate.

1.3.4 Mirror Descent (MD)

$R : \mathbb{R}^n \to \mathbb{R}$ strongly-convex.

$$x_{t+1} = \arg\min_y \{\eta \sum_{s=1}^{t} \langle y, \nabla f(x_s) \rangle + R(y)\},$$

i.e. find the minimum of the sum of all t planes (each is a plane approximated on a point x_i), adding regularization R. It is equivalent to

$$x_{t+1} = \arg\min_y \{V_{x_t}(y) + \alpha \langle y - x_t, \nabla f(x_t) \rangle\},$$

$V_x(y) = R(y) - R(x) - \langle y - x, \nabla R(x) \rangle$.

- If $||\nabla f(x)|| \le \rho$, $T = O(\rho^2/\epsilon^2)$.

1.3.5 Linear Coupling

Update rule:

$$x_{k+1} = \tau z_k + (1 - \tau) y_k,$$
$$y_{k+1} = \mathrm{GD}(x_{k+1}),$$
$$z_{k+1} = \mathrm{Mirr}_{z_k}(\alpha \nabla f(x_{k+1})),$$

where $\mathrm{Mirr}_x(\xi) = \arg\min_y \{V_x(y) + \langle \xi, y - x \rangle\}$.

- Set $\frac{1 - \tau}{\tau} = \alpha L$.

Let $f(x_0) - f(x^\star) \le d$, $V_{x_0}(x^\star) \le \Theta$.
- Set $\alpha = \sqrt{\Theta/Ld}$,

$$f(\bar{x}) - f(x^\star) \le \frac{2\sqrt{L\Theta d}}{T},$$

so in $T = 4\sqrt{L\Theta/d}$ steps, $f(\bar{x}) - f(x^\star) \le d/2$. Thus we need

$$T = O(\sqrt{L\Theta/\epsilon}).$$

1.3.6 Matrix Completion

- Netflix Grand Challenge
- *Assumption* of Matrix: Low Rank; Known Entries are Uniformly Distributed; Incoherence (not too sparse).
- Minimizing rank(A) is hard. Alternatives: **Nuclear Norm** (sum of abs of singular values)

Assume $A = UV^T$. Solve

$$\min_{U,V \in \mathbb{R}^{n\times r}} \left|\left|P_\Omega(UV^T) - P_\Omega(A)\right|\right|_F^2.$$

Ω are known entries. **Alternating Minimization**: $U^1 \to V^1 \to U^2 \to \cdots$. Each problem is **convex**, so is efficient.

1.3.7 Non-Convex Optim.

- **Saddle Point**: $\nabla^2 L(w)$ has both + / - eigenvalues, say **strict SP**; if $\nabla^2 L(w) \ge 0$ but has eigenvalue 0, say **flat saddle point**.

SGD escapes all saddle points for strict saddle functions. Assumingeach local minima are equally good, SGD finds the optimal point.

- if L is smooth, bounded, strict saddle, Hessian is smooth, SGD noise has non-negligible variance in every direction with constant probability: **SGD escapes all saddle points and local maxima**, converges to local minimum after **POLY number of steps**.
- Let $y_t = x_t - \eta\nabla L(x_t)$, then if the update rule is $x_{t+1} = x_t - \eta\nabla L(x_t) - \eta\omega_t$ (ω_t is noise), $\{y_t\}$ is doing GD using **avg gradient of neighborhood**,

$$E_{\omega_t}[y_{t+1}] = y_t - \eta\nabla E_{\omega_t} L(y_t - \eta\omega_t).$$

1.4 Generalization

Generalization Theory: Assuming training samples are i.i.d. from D, we can use L_{train} to get upper bound of L_D.

1.4.1 No Free Lunch

- **Theorem.** Let A be any learning algorithm for binary classification over domain X. Training set size $m \le |X|/2$. Then there exists a distribution D (**data**), such that: (i) there is an $f : X \to \{0,1\}$ with $L_D(f) = 0$ (there is a solution); (ii) w.p. $\ge 1/7$ over the choice of $S \sim D^m$, $L_D(A(S)) \ge 1/8$. (With a high probability of selecting training data, the algorithm fails to learn D)

1.4.2 PAC Learning

Hypothesis class H. ERM is the hypothesis with smallest training loss,

$$ERM_H(S) \in \arg\min_{h\in H} L_S(h).$$

- **Realizability Assumption**: There exists $h^\star \in H$ with 0 population error.

- **Theorem.** (ERM hypothesis is a good solution) H finite, $\delta \in (0,1), \epsilon > 0$, integer $m \geq \frac{\log(|H|/\delta)}{\epsilon}$. Then for any data (D, f) for which *realizability assumption* holds, w.p. $\geq 1-\delta$ over choice of $S \sim D^m$, ERM hypo h_S holds

$$L_{D,f}(h_S) \leq \epsilon.$$

- **PAC learnablility**: Hypo class H is **PAC learnable** if, there is a function $m_H : (0,1)^2 \to N$, and a learning alg. A, such that: $\forall \epsilon, \delta \in (0,1)$, distribution D over X, and $f : X \to \{0,1\}$, with *realizable assumption* holds, when running A on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples from (D, f), A returns $h \in H$, w.p. $\geq 1-\delta$, $L_{D,f}(h) \leq \epsilon$. (With enough data, the alg. learns well)

- **Agnostic PAC learnability** (without realizable assumption): For all dist. D over $X \times Y$, when running A on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples from D, w.p. $\geq 1-\delta$ we have the learned hypo h that

$$L_D(h) \leq \min_{h' \in H}\{L_D(h')\} + \epsilon =: \epsilon_{app} + \epsilon_{est}.$$

- H: class of threshold functions. Then H is PAC learnable, with $m_H(\epsilon, \delta) \leq \frac{\lceil \log(2/\delta) \rceil}{\epsilon}$.

1.4.3 VC Dim

- Suppose training set size is m, and VC dimension of H is $\geq 2m$. Then for any A, there is D over $X \times \{0,1\}$, such that there is $h \in H$ s.t. $L_D(h) = 0$ but w.p. $\geq 1/7$ over $S \sim D^m$, $L_D(A(S)) \geq 1/8$.

- If H has infinite VC-dim, H is not PAC learnable.

- **Fundamental Thm for Statistical Learning.** (quantitative ver.) Hypo class $H : X \to \{0,1\}$, $\text{VCdim}(H) = d < \infty$. Then there are absolute constants C_1, C_2, s.t. H is agnostic PAC learnable with

$$\min m_H(\epsilon, \delta) = \Theta(\frac{d - \log \delta}{\epsilon^2}),$$

and PAC learnable with

$$\min m_H(\epsilon, \delta) = \Theta(\frac{-d \log \epsilon - \log \delta}{\epsilon}).$$

1.5 Supervised Learning

1.5.1 Linear Regression

$f(x) = w^T x + b$, $L(f, x_i, y_i) = \frac{1}{2}(f(x_i) - y_i)^2$.

Exact minimizer: $w^\star = (X^T X)^{-1} X^T y$. Can also use (S)GD: L is a convex function, so always converges.

1.5.2 Perceptron (Classification)

$f(x) = \text{sgn}(w^T x + b)$. Learing process: find a misclassified pair (x, y). If $y = -1$, $w \leftarrow w - x$; if $y = 1$, $w \leftarrow w + x$. However, if data is not linearly seperable, it will not converge.

- Assume exists $||w^\star|| = 1$ and $\gamma, R > 0$, $y_i \langle w^\star, x_i \rangle \geq \gamma$ (**Linearly seperable**), $||x_i|| \leq R$. Then Perceptron makes $\leq \frac{R^2}{\gamma^2}$ mistakes.

1.5.3 Ridge Regression

Minimize

$$\frac{1}{2N} \sum_{i=1}^{N} (w^T x_i - y_i)^2 + \frac{\lambda}{2} ||w||_2^2.$$

(Add a L2 regularization term to Linear regression). Hessian matrix

$$\frac{1}{N} \sum x_i x_i^T + \lambda I,$$

which is strongly convex. When training, the first step is identical to linear regression

$$w'_{t+1} = w_t - \frac{\eta}{N}\sum(w_t^T x_i - y_i)x_i,$$

then do shrinking

$$w_{t+1} = w'_{t+1} \cdot (1 - \eta\lambda).$$

The second part is called **weight dacay** in DL. `weight-decay = a` means weight $\times(1-a)$.

1.5.4 LASSO

When having a lot of features, we want to just use **important features**. i.e. let $||w||_0$ (number of non-zero elements) be smaller. Using $||w||_1$ to approximate this, the objective becomes

$$\frac{1}{2N}\sum_{i=1}^{N}(w^T x_i - y_i)^2 + \lambda||w||_1.$$

Training step: first step is linear rgn, the second step:

1. if $w'^i_{t+1} > \eta\lambda$, $w^i_{t+1} \leftarrow w'^i_{t+1} - \eta\lambda$;
2. if $w'^i_{t+1} < -\eta\lambda$, $w^i_{t+1} \leftarrow w'^i_{t+1} + \eta\lambda$;
3. if $|w'^i_{t+1}| < \eta\lambda$, $w^i_{t+1} \leftarrow 0$.

- **Intuition**: Linear rgn term can be seen as a *quadratic* function, and $||w||_2, ||w||_1$ give ball and diamond contour, respectively. *Diamond gives sparse intersections.*

1.5.5 Compressed Sensing

- **Nyquist Thm**: For a signal with frequency f, we need $2f$ *sampling rate* to fully reconstruct the signal.

- However, in real world scenarios, signals are **sparse** (e.g. in Fourier space / DWT basis), so we need fewer samples. **Advantage**: low cost, high performance, e.g. magnetic resonance imaging, Imaging outside visible spectrum, wireless communication.

- Formally: x is long vector but structured (sparse), $A = [a_1, a_2, \cdots, a_n]^T \in \mathbb{R}^{n\times d}$ is **measurement matrix** (data points). Given $y = Ax$, we want to recover x using A (designed by ourself) and **observation** y.

- **Note**: Comparing with LASSO, we are given $(X_{\text{train}}, Y_{\text{train}})$ to learn w: X_{train} is picked by ourselves, Y_{train} is observed, then w is learned, knowing that $w^\star$ is sparse.

- **RIP Condition**: $W \in \mathbb{R}^{n\times d}$ is (ϵ, s)-RIP, if for all $x \neq 0$ such that $||x||_0 \leq s$, we have

$$(1-\epsilon)||x||_2^2 \leq ||Wx||_2^2 \leq (1+\epsilon)||x||_2^2.$$

(**restricted isometry property**)

- **Lemma**: W is $(\epsilon, 2s)$-RIP. I, J is disjoint sets of size $\leq s$. Then, for any vector u, $\langle Wu_I, Wu_J\rangle \leq \epsilon||u_I|| \cdot ||u_J||$.

- **Thm**: Let $\epsilon < 1/(1+\sqrt{2})$, W is an $(\epsilon, 2s)$-RIP matrix. x is an arbitrary vector,

$$x_s \in \arg\min_{||v||_0 \leq s} ||x - v||_1.$$

Let $y = Wx$, and

$$x^\star \in \arg\min_{Wv=y} ||v||_1$$

be the reconstructed vector. Then,

$$||x^\star - x||_2 \leq \frac{2(1+\rho)s^{-1/2}}{1-\rho}||x - x_s||_1,$$

where $\rho = \dfrac{\sqrt{2}\epsilon}{1-\epsilon}$. When $x = x_s$ is s-sparse, we can reconstruct $x^\star = x$.

- **Thm (Random Matrix is likely RIP).** U is an arbitrary fixed $d \times d$ orthonormal matrix, $\epsilon, \delta \in (0,1)$. $s \in [d]$, $n \geq 100 \cdot \frac{s \log(40d/\delta\epsilon)}{\epsilon}$ is an integer. $W \in \mathbb{R}^{n \times d}$, each element of $W \sim \mathcal{N}(0, 1/n)$. Then, w.p. $\geq 1-\delta$, WU is (ϵ, s)-RIP.

- When the sparsity is not on a linear basis, i.e. $y = A \cdot G(z)$, G being a non-linear function, mapping z (low-dimension) into x (high-dimension) - G is given, but z is unknown: to recover z, minimizing $||y - AG(z)||_2$ is great, where $y = Ax^\star + \eta$:

- **Thm.** $G : \mathbb{R}^k \to \mathbb{R}^n$ be a generative model from a d-layer NN using RELU activations, $A \in \mathbb{R}^{m \times n}$ with $m = O(kd \log n)$, $A_{i,j} \sim \mathcal{N}(0, 1/m)$. For any $x^\star \in \mathbb{R}^n$, and observation $y = Ax^\star + \eta$, let $\hat{z}$ minimize $||y - AG(z)||_2$ to within additive ϵ of the optimum, then w.p. $\geq 1 - e^{-\Omega(m)}$,

$$||G(\hat{z}) - x^\star||_2 \leq 6 \cdot \min_{z^\star \in \mathbb{R}^k} ||G(z^\star - x^\star)||_2 + 3||\eta||_2 + 2\epsilon.$$

In the right hand side, the three terms are: the *best approximation*, *observing noise*, and *optimization accuracy*, respectively.

1.5.6 SVM

Hard Margin Constraints: $y_i(w^T x_i - b) \geq 1$, need to minimize $||w||_2$. Can be solved using *quadratic programming* in POLY time.

Soft Margin Minimize $\frac{1}{2}||w||_2^2 + \lambda \cdot \sum_i \xi_i$, satisfying

$$\xi_i \geq 0, \quad y_i \cdot (w^T x_i - b) \geq 1 - \xi_i.$$

Call ξ_i **slack variable**. Intuitively, instead using $1_{y_i \cdot (w^T x_i - b) \leq 0}$ as loss fn (not differentiable), we use **Hinge loss** (*soft version*): $\max\{0, 1 - y_i \cdot (w^T x_i - b)\}$.

Dual Form (Soft Margin)

$$\arg \max_{c_1, \cdots, c_n} \sum_{i=1}^n c_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} y_i y_j c_i c_j \langle x_i, x_j \rangle,$$

subject to $\sum_{i=1}^n c_i y_i = 0$, $0 \leq c_i \leq \lambda$ for all i. The original w can be obtained by

$$w = \sum_{i=1}^n c_i y_i x_i.$$

If $c_i \neq 0$, call x_i a **support vector**.

Kernel Method Transform the data x_i into another space: $x_i \to \Phi(x_i)$. We want $K(x_i, x_j) := \langle \Phi(x_i), \Phi(x_j) \rangle$ (K is the **kernel** of Φ) can be computed easily. In fact, let $K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^p$, then K can be computed in $O(d)$ time, while $\Phi(x_i)$ itself lies in $O(d^p)$ space. When doing rgn,

$$w^T \Phi(x) = \left(\sum_{i=1}^n c_i y_i \Phi(x_i) \right)^T \Phi(x) = \sum_{i=1}^n c_i y_i K(x_i, x).$$

- **Mercer's Thm.** If the matrix $(K(x_i, x_j))_{i,j \geq 1}$ is *positive semidefinite* for any $\{x_i\}$, then there exists Φ such that K is the kernel of Φ.

- **Gaussian Kernel**: $K(x_i, x_j) = \exp\{-||x_i - x_j||^2 / 2\sigma^2\}$.