CMPUT 654: Mathematical Foundations of Modern AI Systems Fall 2026

Lecture 7: Distribution shift and helping models learn

September 24, 2026

Lecturer: Csaba Szepesvári *Note prepared by: Csaba Szepesvári*

Reading guide. The guiding question is: when does a change in the data law change what should be learned, and when can a change in the supplied data make learning easier? The first part distinguishes covariate shift from general distribution shift and explains why correct specification and coverage determine whether a shift is harmful. The second part organizes data selection as communication between a teacher and a student. The third part uses parity to illustrate a difficulty in teaching a neural network a desired behavior. Direct final-answer training can fail even when the network can represent the target. A modified training task exposes intermediate computations while preserving the desired final answer. The lecture stated the negative result, explained its proof mechanism, and previewed the intermediate-supervision construction. Lecture 8 supplies the complete gradient and trajectory argument and develops the positive constructions. The prerequisites are conditional expectation, KL divergence, total variation, Pinsker's inequality, Hoeffding's inequality, ReLU networks, and elementary linear algebra. The brief finite-field and Fourier explanations below supply the additional background used in the parity proof.

7.1 Distribution shift depends on the prediction problem

Let P_{tr} and P_{dep} be the training and deployment laws on $\mathcal{X} \times \mathcal{Y}$. Write μ_{tr} and μ_{dep} for their input marginals.

For a fixed predictor f, let $c_f : \mathcal{X} \times \mathcal{Y} \to [0,1]$ be its downstream cost and write

$$J_P(f) = \mathbb{E}_P[c_f(X,Y)].$$

Proposition 7.1.1 (Total variation controls every fixed bounded cost)**.** *For every fixed predictor f,*

$$J_{P_{\mathrm{dep}}}(f) - J_{P_{\mathrm{tr}}}(f) \le \mathrm{TV}(P_{\mathrm{dep}}, P_{\mathrm{tr}}). \tag{7.1}$$

Consequently,

$$\left|J_{P_{\mathrm{dep}}}(f) - J_{P_{\mathrm{tr}}}(f)\right| \le \mathrm{TV}(P_{\mathrm{dep}}, P_{\mathrm{tr}}). \tag{7.2}$$

Proof. Take a measure ν dominating both laws and let p_{dep} and p_{tr} be their densities. Then

$$\begin{aligned} J_{P_{\mathrm{dep}}}(f) - J_{P_{\mathrm{tr}}}(f) &= \int c_f(z)\big(p_{\mathrm{dep}}(z) - p_{\mathrm{tr}}(z)\big)\,\nu(dz) \\ &\le \int_{\{p_{\mathrm{dep}} \ge p_{\mathrm{tr}}\}} \big(p_{\mathrm{dep}}(z) - p_{\mathrm{tr}}(z)\big)\,\nu(dz) \\ &= \mathrm{TV}(P_{\mathrm{dep}}, P_{\mathrm{tr}}). \end{aligned}$$

Interchanging the two laws gives the absolute-value bound. □

The bound is uniform over all costs in $[0,1]$. For the particular cost of a particular predictor, the exact difference is the signed integral in the first line of the proof. It can be small even when the total-variation distance is large, because the cost can be small where the laws differ.

Definition 7.1.2 (Covariate shift and general distribution shift)**.** There is *covariate shift* when the two joint laws admit the factorizations

$$P_{\mathrm{tr}}(dx, dy) = \mu_{\mathrm{tr}}(dx)K(dy \mid x), \qquad P_{\mathrm{dep}}(dx, dy) = \mu_{\mathrm{dep}}(dx)K(dy \mid x) \tag{7.3}$$

for one conditional kernel K, while $\mu_{\mathrm{tr}} \neq \mu_{\mathrm{dep}}$. This definition also specifies the conditional target at inputs that occur under only one of the two marginals. There is *general distribution shift* when $P_{\mathrm{tr}} \neq P_{\mathrm{dep}}$; the conditional law of Y given X may change as well.

Covariate shift changes how often inputs occur while preserving the desired prediction at each input. General distribution shift may also change the desired prediction. This classification depends on what is included in X.

Proposition 7.1.3 (A source indicator can turn conditional shift into covariate shift)**.** *Suppose that X, Y, and the source indicator J take values in finite sets. For each source j, let μ_j be its input law and K_j its conditional kernel, so that $P_j(x, y) = \mu_j(x)K_j(y \mid x)$. For mixture weights α_j, define the joint law*

$$P_\alpha(x, y, j) = \alpha_j P_j(x, y), \qquad \alpha_j \geq 0, \qquad \sum_j \alpha_j = 1.$$

If the predictor observes only X, its conditional kernel is, at every x with positive P_α-probability,

$$K_\alpha(y \mid x) = \sum_j K_j(y \mid x)P_\alpha(j \mid x), \tag{7.4}$$

which generally changes with the mixture weights. If the predictor observes (X, J), then

$$\widetilde{K}_\alpha(y \mid x, j) = K_j(y \mid x), \tag{7.5}$$

which is independent of the mixture weights.

The same corpus change can therefore be general shift for one prediction problem and covariate shift for another. Source metadata is useful only if the model receives it, preserves it, or can infer it from the context.

Example 7.1.4 (Benign and harmful changes)**.** Increasing the proportion of mathematics prompts is covariate shift if the desired answer to every prompt is unchanged. Combining sources that use incompatible conventions changes the conditional law of Y given X when the source is hidden. A recommender system can create a further effect: its outputs change what users consume, which can later change their preferences and hence the conditional law of feedback.

Evaluation has the same dependence on its law. For a loss ℓ, an evaluation set drawn from P_{eval} yields an empirical average that estimates

$$L_{P_{\mathrm{eval}}}(f) = \mathbb{E}_{P_{\mathrm{eval}}}[\ell(f(X), Y)].$$

The empirical average estimates deployment risk when the construction of the evaluation set represents the deployment law, or when a separate argument relates the two laws. Curation is therefore part of evaluation as well as training.

7.2 Correct specification, misspecification, and coverage

For a joint law P on $\mathcal{X} \times \mathcal{Y}$, let X, Y denote its coordinate maps and define the population risk

$$L_P(f) = \mathbb{E}_P[\ell(f(X), Y)].$$

Definition 7.2.1 (Correct specification and misspecification)**.** A conditional model class $\mathcal{F}$ is *correctly specified* for P if it contains a predictor that attains the optimal conditional decision at every input $x \in \mathcal{X}$. It is *misspecified* if no member of the class does so.

Correct specification is a claim about the interaction among the target conditional law, the loss, and the model class. It is stronger than saying that the class contains a predictor with small average training risk.

Proposition 7.2.2 (A common pointwise optimum survives covariate shift)**.** *Suppose P and Q share a conditional kernel $K(dy \mid x)$ as in Eq. (7.3), and write μ_P and μ_Q for their input marginals. If $f^\star \in \mathcal{F}$ satisfies, for every $x \in \mathcal{X}$ and every $f \in \mathcal{F}$,*

$$\int \ell(f^\star(x), y) K(dy \mid x) \le \int \ell(f(x), y) K(dy \mid x),$$

then $f^\star$ minimizes both L_P and L_Q.

Proof.

$$\begin{aligned} L_Q(f) - L_Q(f^\star) &= \int_{\mathcal{X}} \left[\int_{\mathcal{Y}} \big(\ell(f(x), y) - \ell(f^\star(x), y)\big) K(dy \mid x) \right] \mu_Q(dx) \\ &\ge 0. \end{aligned}$$

Replacing μ_Q by μ_P proves the same claim for P. □

For conditional log loss, let $p^\star(\cdot \mid x)$ be the true conditional law and $q_f(\cdot \mid x)$ the model prediction under P.

$$L_P(f) = H_P(Y \mid X) + \mathbb{E}_P\big[D\big(p^\star(\cdot \mid X) \| q_f(\cdot \mid X)\big)\big]. \tag{7.6}$$

If one model represents $p^\star(\cdot \mid x)$ at every deployment input, changing the input mixture does not change this population-risk minimizer. The mixture can still change the number of observations at each input, the conditioning of the optimization problem, and which minimizer is selected in an overparameterized class.

Proposition 7.2.3 (Misspecification makes the mixture part of the target)**.** *Let $\mathcal{X} = \{a, b\}$, $\mathcal{Y} = \{0, 1\}$, and define a conditional kernel K by*

$$K(1 \mid a) = 0.1, \qquad K(1 \mid b) = 0.9.$$

For $\lambda \in [0, 1]$, let $\mu_\lambda(a) = \lambda$, $\mu_\lambda(b) = 1 - \lambda$, and $P_\lambda(x, y) = \mu_\lambda(x) K(y \mid x)$. Restrict the model to one probability q at both inputs. The log-loss minimizer under P_λ is

$$q^\star_\lambda = 0.9 - 0.8\lambda.$$

Proof. Write $r_\lambda = P_\lambda(Y = 1) = 0.9 - 0.8\lambda$. The risk is

$$L_{P_\lambda}(q) = -r_\lambda \log q - (1 - r_\lambda) \log(1 - q).$$

Therefore

$$\frac{d}{dq} L_{P_\lambda}(q) = -\frac{r_\lambda}{q} + \frac{1 - r_\lambda}{1 - q},$$

and the unique stationary point is $q = r_\lambda$. □

The restricted model must compromise between the two contexts. Their weights determine the compromise. In a sufficiently expressive language model, the same parameters may represent both mathematics and literature conditionals, so a moderate mixture change need not change the ideal predictor. This is an inductive-bias claim about the shared representation; it does not follow from model size alone.

A *minimum-coverage condition* supplies a separate requirement. The density-ratio condition below is one sufficient, model-independent form of coverage; it is stronger than necessary for structured model classes.

Proposition 7.2.4 (Minimum coverage transfers excess risk)**.** *Assume covariate shift with input marginals μ_{tr} and μ_{dep}, and let*

$$\Delta_f(x) = \int \big(\ell(f(x), y) - \ell(f^\star(x), y)\big) K(dy \mid x) \geq 0.$$

If $\rho \in (0, 1]$, μ_{dep} is absolutely continuous with respect to μ_{tr} and

$$\frac{d\mu_{\mathrm{dep}}}{d\mu_{\mathrm{tr}}}(x) \leq \rho^{-1} \quad \mu_{\mathrm{tr}}\text{-almost surely,}$$

then

$$L_{P_{\mathrm{dep}}}(f) - L_{P_{\mathrm{dep}}}(f^\star) \leq \rho^{-1}(L_{P_{\mathrm{tr}}}(f) - L_{P_{\mathrm{tr}}}(f^\star)) \,. \tag{7.7}$$

Proof. Let $\omega = d\mu_{\mathrm{dep}}/d\mu_{\mathrm{tr}}$. Then

$$\int \Delta_f(x)\, \mu_{\mathrm{dep}}(dx) = \mathbb{E}_{\mu_{\mathrm{tr}}}[\omega \Delta_f] \leq \rho^{-1} \mathbb{E}_{\mu_{\mathrm{tr}}}[\Delta_f].$$

□

If deployment assigns positive mass to a region absent from training, no finite ρ exists. This does not make correct prediction there impossible. A structured class can identify behavior at unseen inputs from observations elsewhere. For example, a linear predictor can be identified when the training covariates cover the parameter directions needed at deployment, even if the individual deployment inputs were never observed. Thus the necessary notion of coverage depends on the model and the prediction problem: it must identify the relevant degrees of freedom, rather than literally include every deployment input. The density-ratio condition avoids this structural analysis by imposing a stronger distribution-level requirement. The term *domain adaptation* commonly refers to methods that use source data together with information about a different target domain, often unlabeled target inputs, to obtain a predictor for the target law.

7.3 Learning as communication between teacher and student

Data selection can be described as a communication protocol. A *teacher* sends data or feedback; a *student* updates its internal state and may send queries or predictions. A useful classification separates four questions.

Dimension	Question
Objective information	Does the teacher know the target predictor, desired behavior, or only a verifier? What does the student know?
Student information	Does the teacher know the student's representation or hypothesis class, update algorithm, current state, or only observed outputs?
Control of communication	Who chooses the next input, label, intermediate state, correction, or evaluation: the teacher, the student, or both?
Number of turns	Is the data fixed in one batch, or can later messages depend on the transcript and the student's current state?

These dimensions are independent. In a *passive learning* protocol, the student receives data selected without using its queries or current state. *Data curation* selects examples or changes their weights; *task curation* changes the supervised task; and a *curriculum* controls the order or difficulty of the supplied examples or tasks. Any of these choices can be fixed in advance or adapted over several turns.

Active learning usually gives the student control over queries whose answers are supplied by an oracle. In *active teaching*, the teacher chooses the next message using the interaction so far. *Machine teaching* usually gives the teacher a target and a model of the student, and asks the teacher to choose effective training data. In a multi-turn protocol, either side or both sides can adapt to the transcript. Thus "active" identifies who selects the next message in the protocol.

The teacher's information is a resource. Knowing the target and the student's update rule can reduce the required number of examples. It can also make a teaching strategy brittle when the assumed student differs from the actual one. The student's ability to query is another resource. It can expose the specific uncertainty that the current data leave unresolved.

The communication channel also matters. Reordering final-answer examples changes the optimization path. Replacing

$$(x, y) \quad \text{by} \quad (x, z_1, \ldots, z_r, y)$$

adds information about intermediate computations. Mathematical reasoning is a favorable setting because problems can often be generated, final answers can often be verified, intermediate states may be checkable, and computation length can be controlled.

When learning is modeled through a hypothesis class, correct specification says that the desired behavior is represented. It does not imply that the student's update rule can find that behavior. Changing the communicated data or the supervised task can change this optimization problem while preserving the desired final output.

The parity example applies this distinction between the desired behavior and the training task used to teach it. The desired final output remains parity. What the teacher reveals during training will change.

7.4 Parity as a test case for teaching a computation

This example illustrates one difficulty in teaching a neural network a desired behavior and one way to overcome it. The direct approach supplies input–output pairs and trains the network to predict only the final answer. The result in Section 7.5 shows that a specified version of this approach can fail. This does not imply that every way of teaching a neural network to produce the same answer must fail. In Section 7.7, the training task will expose intermediate computations, and completing the modified task will still produce the original final answer.

The same parity problem has two useful representations. For $b \in \{0,1\}^d$, define

$$\mathrm{parity}_S(b) = \bigoplus_{j \in S} b_j.$$

Set $x_j = (-1)^{b_j}$. Then

$$(-1)^{\mathrm{parity}_S(b)} = \prod_{j \in S} x_j. \tag{7.8}$$

The bit representation makes XOR computation and linear algebra over $\mathbb{F}_2$ explicit. The sign representation makes products, correlations, and the ReLU gradient calculation explicit.

For $S \subseteq [d]$ and $x \in \{-1,1\}^d$, define

$$\chi_S(x) = \prod_{j \in S} x_j.$$

Fix S and define the data law P_S on $\{-1,1\}^d \times \{-1,1\}$ by

$$P_S(x,y) = 2^{-d}\mathbb{I}\{y = \chi_S(x)\}.$$

Let X, Y be the coordinate maps under P_S. Thus X is uniform on the Boolean cube and $Y = \chi_S(X)$ with P_S-probability one.

We note in passing that for any $0 < \delta < 1$, after seeing $\Omega(d + \log(1/\delta))$ labeled examples where the inputs come from the uniform distribution, S can be determined with probability $1 - \delta$ in a computationally efficient manner (see Exercise 2). As we shall see, training a neural network with gradient descent to do the same does not succeed. However, before that, we establish that the target lies in the network class considered in the negative result: without this result, the negative result would not make sense.

7.4.1 A small ReLU network represents the fixed parity

Consider the one-hidden-layer ReLU network

$$N_\theta(x) = \sum_{r=1}^{m} u_r [w_r^\top x + b_r]_+. \tag{7.9}$$

Proposition 7.4.1 (Parity has a small ReLU representation)**.** *For every S, a network of the form Eq. (7.9) with width at most $|S| + 1$ represents χ_S on the Boolean cube.*

Exercise 3 asks you to verify an explicit width-$(|S| + 1)$ construction.

7.5 Direct gradient training can fail for a fixed parity

For a score f, define its *signed correlation loss*

$$\mathcal{C}(f) = -\mathbb{E}_{P_S}[Yf(X)],$$

and define the parameter objective

$$F(\theta) = \mathcal{C}(N_\theta) = -\mathbb{E}_{P_S}[YN_\theta(X)]. \tag{7.10}$$

For a hard predictor $h : \{-1,1\}^d \to \{-1,1\}$, define its classification risk by $R_{01}(h) = P_S(h(X) \neq Y)$. For binary-valued predictions, $\mathcal{C}(h) = 2R_{01}(h) - 1$. The identity does not extend to a real-valued score f: an arbitrarily small score can have the correct sign everywhere. Exercise 4 asks you to prove these claims and derive the corresponding decompositions for squared and logistic loss. The main result below analyzes perturbed gradient descent on F itself. Its classification conclusion follows from symmetry of the entire parameter law, rather than from the numerical value of $\mathcal{C}(N_\theta)$.

On an ambient probability space, let $\Theta_0, \Xi_0, \dots, \Xi_{T-1}$ be mutually independent, with

$$\mathcal{L}(\Theta_0) = \mathcal{N}(0, \sigma^2 I), \qquad \mathcal{L}(\Xi_t) = \mathcal{N}(0, \sigma^2 I).$$

Starting from Θ_0, run *perturbed gradient descent*

$$\Theta_{t+1} = \Theta_t - \eta \nabla F(\Theta_t) - \Xi_t. \tag{7.11}$$

Define the comparison process obtained by deleting the gradient:

$$\Theta_0^{(0)} = \Theta_0, \qquad \Theta_{t+1}^{(0)} = \Theta_t^{(0)} - \Xi_t. \tag{7.12}$$

The independent Gaussian increments make the comparison process Gaussian at every time. This symmetry is the reason the small-gradient estimate can be applied repeatedly along the comparison path.

The proof of the main result uses the following comparison between the trained parameter path and the Gaussian walk.

Lemma 7.5.1 (Path comparison)**.** *Let $d > 30$, $k = |S|$, and $A > 0$. Let $m, T \in \mathbb{N}$ and $\eta, \sigma > 0$, and assume*

$$m, T, \eta, \sigma, \sigma^{-1} \leq d^A, \qquad k \geq (288A + 144)\log d. \tag{7.13}$$

Then

$$\mathrm{TV}\Big(\mathcal{L}(\Theta_0, \dots, \Theta_T), \mathcal{L}(\Theta_0^{(0)}, \dots, \Theta_T^{(0)})\Big) \leq e^{-k/36}. \tag{7.14}$$

The comparison process is independent of S. The lemma says that the gradient changes the law of a polynomial-length parameter path by only an exponentially small amount, even though the network class contains the target.

The symmetry of the comparison process now gives the main result.

Theorem 7.5.2 (Classification advantage occurs with probability at most one half)**.** *Let $d > 30$, $k = |S|$, and $A > 0$. Let $m, T \in \mathbb{N}$ and $\eta, \sigma > 0$, and assume*

$$m, T, \eta, \sigma, \sigma^{-1} \leq d^A, \qquad k \geq (288A + 144)\log d.$$

Let $h_\theta(x) = \mathrm{sign}(N_\theta(x))$, with fair tie-breaking when $N_\theta(x) = 0$. Equivalently,

$$R_{01}(h_\theta) = P_S(YN_\theta(X) < 0) + \frac{1}{2}P_S(N_\theta(X) = 0).$$

Then, for every $\gamma > 0$,

$$\mathbb{P}\left(R_{01}(h_{\Theta_T}) \le \frac{1}{2} - \gamma \right) \le \frac{1}{2} + e^{-k/36}, \tag{7.15}$$

and

$$\left| \mathbb{E}[R_{01}(h_{\Theta_T})] - \frac{1}{2} \right| \le e^{-k/36}. \tag{7.16}$$

The expectation bound permits a symmetric mixture of very accurate and very inaccurate classifiers. The probability bound gives the corresponding reliability statement: the chance of achieving any fixed positive advantage over random guessing is at most one half, up to the exponentially small term.

7.6 Why the comparison theorem is plausible

The proof of Lemma 7.5.1 has two parts. First, when the parameters have an isotropic Gaussian law, the expected norm of the parity-dependent gradient is exponentially small in k. The calculation expresses each gradient coordinate through correlations between a random threshold and a degree-k Boolean Fourier character. Low-degree terms cancel, leaving only an exponentially small remainder.

Second, the comparison process in Eq. (7.12) is Gaussian at every time. The small-gradient estimate can therefore be applied repeatedly along this path. A KL chain-rule calculation compares the actual update kernel with an auxiliary kernel that deletes sufficiently small gradients. Pinsker's inequality and a coupling of the auxiliary process with the Gaussian walk then give Eq. (7.14). The sign symmetry of the Gaussian walk yields the classification statement in Theorem 7.5.2.

This argument explains the obstruction. The network class contains the fixed parity, but typical Gaussian parameters lie in a region where the objective supplies exponentially little information about the direction in which the parameters should move. Lecture 8 gives the complete calculation.

7.7 Changing the training task

The lower bound concerns direct training on final parity labels. It does not apply to every way of teaching a neural network to compute parity. Write $x_i = (-1)^{b_i}$ with $b_i \in \{0, 1\}$ and define the running parities

$$z_1 = b_1, \qquad z_i = z_{i-1} \oplus b_i.$$

The final value z_d determines the original parity, while every intermediate label demonstrates the same two-bit XOR rule. Thus intermediate supervision changes the learning problem while preserving the required final answer.

Lecture 8 develops three versions of this idea: composing an explicitly learned XOR gate, exposing the hidden first decision in the path-star task, and a constructed transformer result in which successive gradient updates learn retrieval and combination.

7.8 Take-home messages

- Total variation uniformly controls the change in every fixed bounded downstream cost. A particular cost can remain stable even when this uniform bound is large.
- Covariate shift preserves the conditional target; general distribution shift need not. The distinction depends on the information included in the input.

- Under correct specification, a common pointwise optimum survives a change in input frequencies. Under misspecification, mixture weights determine the best compromise.
- Correct inductive bias and coverage address separate requirements. A density-ratio bound quantifies one sufficient minimum-coverage condition.
- Teaching and active learning describe different control structures. The relevant dimensions are what each party knows, who selects each message, what the communication contains, and whether later messages adapt to earlier ones.
- Noiseless examples identify an unknown parity efficiently, and a small ReLU network represents every fixed parity. For the specified perturbed population-gradient method, the expected zero–one risk remains within $e^{-k/36}$ of $1/2$, and the probability of achieving any fixed positive advantage is at most $1/2 + e^{-k/36}$.
- The lower bound concerns one direct training method. Running-parity labels preserve the final computation while supplying repeated supervision for a two-bit rule.

Bibliographical notes

The distinction among covariate shift, other forms of dataset shift, and domain adaptation is developed in the collection edited by Quiñonero-Candela et al. [2009]. The density-ratio calculation in Proposition 7.2.4 is elementary; stronger domain-adaptation results replace the pointwise ratio by discrepancies adapted to a hypothesis class.

The teacher–student terminology follows the separation between active learning and machine teaching used by Zhu [2015] and Liu et al. [2017]. The dimensions in Section 7.3 are the organizational scheme used in this lecture. Other taxonomies use different dimensions and terminology.

Lemma 7.5.1 and Theorem 7.5.2 are based on the fixed-parity lower bound of Shoshani and Shamir [2025]. Lecture 8 gives the Gaussian threshold and trajectory calculations at the center of their argument. The discussion of teacher-forced intermediate parity refers to Wen et al. [2025]; the path-star example and the forward-prediction shortcut are from Hu et al. [2025].

Glossary

Active learning
A protocol in which the student selects queries and receives their labels or answers from an oracle.

Active teaching
A protocol in which the teacher adapts its next message to the interaction so far.

Covariate shift
A change in the input marginal while the conditional law of the target given the input remains fixed.

Correct specification
The model class contains a predictor attaining the optimal conditional decision at every input in the input space.

Curriculum
A choice of the order or difficulty of training examples or tasks, fixed in advance or adapted during training.

Data and task curation
Selection or weighting of training examples, or modification of the supervised task, to influence what the student learns.

Domain adaptation
Learning for a target domain using source data and some information about the target domain.

Downstream cost
The expected cost of using a fixed predictor under a specified evaluation or deployment law.

General distribution shift
A change in the joint law of input and target, possibly including the conditional law of the target given the input.

Intermediate supervision
Training labels reveal intermediate computations whose final result is the original target behavior.

Inductive bias
Structure in the model class or learning rule that determines how observations at some inputs constrain predictions at other inputs.

Machine teaching
Selection of training information by a teacher that has a target and a model of the student's learning process.

Minimum coverage
A condition ensuring that training data identify the degrees of freedom needed at deployment. Proposition 7.2.4 uses a stronger, model-independent bounded density-ratio condition.

Misspecification
The model class cannot attain the optimal conditional decision at every input in the input space.

Perturbed gradient descent
The population-gradient update in Eq. (7.11), with an independent Gaussian perturbation added at every step.

Passive learning
A protocol in which the student receives data selected without using its queries or current state.

Representation capacity
The set of input–output functions expressible by a model class, independently of whether a specified training procedure can find them.

Signed correlation loss
The objective $\mathcal{C}(f) = -\mathbb{E}_{P_S}[Yf(X)]$, which isolates the target-dependent term in the squared-loss and logistic-loss decompositions derived in Exercise 4.

Teacher and student
The teacher supplies data or feedback. The student updates its state and may send queries or predictions in a multi-turn protocol.

Total variation
The uniform discrepancy over event probabilities. Equivalently, it is the largest possible expectation difference over costs taking values in $[0, 1]$.

Zero–one risk
The probability that a binary-valued predictor differs from the binary label.

7.9 Exercises

General instructions. Justify every answer, including yes/no answers, even when the question does not explicitly ask for a justification.

Exercise profiles. Filled mountains show difficulty on a four-mountain scale: one is a direct application, two uses a familiar method, three combines ideas, and four is a substantial challenge. Filled cogs show technical load on a three-cog scale: one is light, two is sustained, and three is dense. Filled hourglasses show estimated time: one is under 20 minutes, two is 20–45 minutes, three is 45–90 minutes, and four is more than 90 minutes. Pale pictograms show the unused part of each scale. The work row uses the canonical order proof, calculation, construction, computation, and interpretation; pale icons mark work types not used. ◎ marks the intended learning outcome. Time estimates cover the non-optional parts. Ratings are not prerequisites or grading weights. A superscript asterisk marks an exercise based on supplementary lecture material; it is not a difficulty rating and does not make the exercise optional when assigned.

Exercise 1: A common target still requires coverage.

◎ Separate preservation of the pointwise target from identification of that target by the training law.

Let $\mathcal{X} = \{a, b\}$, $\mathcal{Y} = \{0, 1\}$, and let the loss be zero–one loss. Suppose the training and deployment laws share the deterministic conditional kernel

$$K(1 \mid a) = 0, \qquad K(1 \mid b) = 1.$$

The model class contains every function from $\mathcal{X}$ to $\mathcal{Y}$.

(a) Prove directly from the two conditional risks that $f^\star(a) = 0$, $f^\star(b) = 1$ minimizes population risk under every input marginal on $\mathcal{X}$.

(b) Let $\mu_{\mathrm{tr}}(a) = 1$ and $\mu_{\mathrm{dep}}(b) = 1$. Construct a predictor with zero training risk and deployment risk one. Explain why the training risk cannot distinguish this predictor from $f^\star$.

(c) Now assume $d\mu_{\mathrm{dep}}/d\mu_{\mathrm{tr}} \le \rho^{-1}$ for some $\rho \in (0, 1]$. Prove the excess-risk bound in Proposition 7.2.4 for this finite input space. Identify the step that fails for the laws in part (b).

Hints For (c), write deployment excess risk as a sum over $x \in \{a, b\}$ and insert the density ratio into each summand.

Exercise 2: Learning noiseless parity from linear equations.

◎ Reduce noiseless parity recovery to linear equations over $\mathbb{F}_2$ and prove a high-probability recovery guarantee.

Let $S \subseteq [d]$ be unknown, and let $(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})$ be independent with law P_S as defined in Section 7.4.

You may use the following standard linear-algebra fact without proof. Given $A \in \mathbb{F}_2^{n \times d}$ and $b \in \mathbb{F}_2^n$, there is an algorithm that determines in a polynomial number of arithmetic operations whether $Az = b$ has a unique solution $z \in \mathbb{F}_2^d$ and returns that solution when it is unique. If z is one solution, then it is unique if and only if

$$Aa \neq 0 \qquad \text{for every } a \in \mathbb{F}_2^d \setminus \{0\}.$$

The usual linear-system theorem over $\mathbb{R}$ remains valid with $\mathbb{F}_2$ in place of $\mathbb{R}$ because both are fields. For background, see Sections 2–4 of David Vogan's notes on Gaussian elimination.

Construct an algorithm that uses a polynomial number of arithmetic operations and whose output $\widehat{S}$ satisfies

$$\mathbb{P}(\widehat{S} \neq S) < 2^{d-n}.$$

Then prove that, for every $\delta \in (0,1)$, the algorithm recovers S with probability at least $1-\delta$ whenever

$$n \geq d + \left\lceil \log_2 \frac{1}{\delta} \right\rceil.$$

HINTS Write

$$X_j^{(i)} = (-1)^{U_{ij}}, \qquad Y^{(i)} = (-1)^{V_i}, \qquad s_j = \mathbb{I}\{j \in S\}.$$

With arithmetic over $\mathbb{F}_2$, the examples give $\mathbf{v} = \mathbf{U}\mathbf{s}$. Apply the stated linear-system procedure. To bound its failure probability, first fix a nonzero $a \in \mathbb{F}_2^d$ and calculate $\mathbb{P}(\mathbf{U}a = 0)$. Then use a union bound over the $2^d - 1$ possible choices of a.

Exercise 3: Representing parity with one ReLU layer.

Verify that a width-$(|S|+1)$ ReLU network represents a fixed parity exactly.

Fix $S \subseteq [d]$, write $k = |S|$, and let $\mathbf{1}_S \in \{0,1\}^d$ be the indicator vector of S. Consider

$$N(x) = [-\mathbf{1}_S^\top x + k + 1]_+ + \sum_{j=1}^{k} 4j(-1)^j [-\mathbf{1}_S^\top x + k + 1 - 2j]_+.$$

Prove that

$$N(x) = \chi_S(x) \qquad \text{for every } x \in \{-1,1\}^d.$$

Conclude that the network in Proposition 7.4.1 needs at most $k+1$ hidden units.

HINTS Let a be the number of negative coordinates of x indexed by S. First show that $-\mathbf{1}_S^\top x + k + 1 = 2a + 1$ and identify which ReLU terms are active. If $\widetilde{N}(a)$ denotes the resulting value, compare $\widetilde{N}(a)$ with $\widetilde{N}(a-1)$. Pair consecutive terms when evaluating $\sum_{j=1}^{a-1} j(-1)^j$.

Exercise 4: What signed correlation controls.

Relate signed correlation to classification and to the gradients of standard surrogate losses.

Under the fixed-parity law P_S, recall

$$\mathcal{C}(f) = -\mathbb{E}_{P_S}[Y f(X)].$$

(a) Let $h : \{-1,1\}^d \to \{-1,1\}$. Prove

$$\mathcal{C}(h) = 2R_{01}(h) - 1, \qquad R_{01}(h) = P_S(h(X) \neq Y).$$

(b) For $\varepsilon > 0$, set $f_\varepsilon = \varepsilon \chi_S$. Compute $\mathcal{C}(f_\varepsilon)$ and $R_{01}(\mathrm{sign}(f_\varepsilon))$. Explain why a correlation value close to zero does not determine the classification risk of the sign of a real-valued score.

(c) For any score $f : \{-1, 1\}^d \to \mathbb{R}$, derive

$$\mathbb{E}_{P_S}[(f(X) - Y)^2] = 1 + \mathbb{E}_{P_S}[f(X)^2] + 2\mathcal{C}(f).$$

Deduce that $|\mathcal{C}(f)| \le \varepsilon$ implies squared loss at least $1 - 2\varepsilon$. Prove

$$R_{01}(\mathrm{sign}(f)) \le \mathbb{E}_{P_S}[(f(X) - Y)^2], \qquad \mathrm{sign}(0) = 1.$$

For the scores in (b), show that the squared loss tends to 1 as $\varepsilon \downarrow 0$ while classification remains perfect.

(d) Derive

$$\mathbb{E}_{P_S}[\log(1 + e^{-Yf(X)})] = \mathbb{E}_{P_S}\left[\log\left(2\cosh\frac{f(X)}{2}\right)\right] + \frac{1}{2}\mathcal{C}(f).$$

Deduce that $|\mathcal{C}(f)| \le \varepsilon$ implies log loss at least $\log 2 - \varepsilon/2$. If $f = f_\theta$ is differentiable in θ, differentiate the identities in parts (c) and (d) and identify every term in addition to $\nabla_\theta \mathcal{C}(f_\theta)$.

Hints For (a), determine the value of $Yh(X)$ on the correct and incorrect events. For (d), verify the pointwise identity

$$\log(1 + e^{-z}) = \log\left(2\cosh\frac{z}{2}\right) - \frac{z}{2}.$$

Exercise 5: The cost of a compromise under a changed mixture.

Quantify how a restricted model's population optimum changes with the training mixture.

For $\lambda \in [0, 1]$, define the input probability mass function μ_λ by

$$\mu_\lambda(a) = \lambda, \qquad \mu_\lambda(b) = 1 - \lambda.$$

Let $k(\cdot \mid x)$ be the conditional probability mass function on $\{0, 1\}$ specified by

$$k(1 \mid a) = 0.1, \qquad k(1 \mid b) = 0.9, \qquad k(0 \mid x) = 1 - k(1 \mid x).$$

Define the joint law P_λ on $\{a, b\} \times \{0, 1\}$ through its mass function

$$P_\lambda(x, y) = \mu_\lambda(x) k(y \mid x).$$

For each $\lambda \in [0, 1]$, let (X, Y) have joint law P_λ. For $q \in (0, 1)$, let $f_q : \{a, b\} \to (0, 1)$ satisfy $f_q(a) = f_q(b) = q$. Define the constant and context-dependent model classes

$$\mathcal{F}_{\mathrm{const}} = \{f_q : q \in (0, 1)\}, \qquad \mathcal{F}_{\mathrm{ctx}} = \{f : \{a, b\} \to (0, 1)\}.$$

For $f \in \mathcal{F}_{\mathrm{ctx}}$, define

$$L_\lambda^{\mathrm{sq}}(f) = \mathbb{E}[(f(X) - Y)^2], \qquad L_\lambda^{\log}(f) = \mathbb{E}[-Y \log f(X) - (1 - Y)\log(1 - f(X))].$$

(a) Let $q_\lambda^\star = 0.9 - 0.8\lambda$. Prove that $f_{q_\lambda^\star}$ is the unique minimizer of both losses over $\mathcal{F}_{\mathrm{const}}$. Establish, for every $q \in (0, 1)$,

$$L_\lambda^{\mathrm{sq}}(f_q) - L_\lambda^{\mathrm{sq}}(f_{q_\lambda^\star}) = (q - q_\lambda^\star)^2, \qquad L_\lambda^{\log}(f_q) - L_\lambda^{\log}(f_{q_\lambda^\star}) = D(\mathrm{Ber}(q_\lambda^\star), \mathrm{Ber}(q)).$$

(b) Suppose training uses λ and deployment uses $\nu \in [0,1]$. Derive the deployment excess risks of the training minimizer:

$$L^{\mathrm{sq}}_{\nu}(f_{q^\star_\lambda}) - L^{\mathrm{sq}}_{\nu}(f_{q^\star_\nu}) = 0.64(\lambda - \nu)^2,$$

$$L^{\log}_{\nu}(f_{q^\star_\lambda}) - L^{\log}_{\nu}(f_{q^\star_\nu}) = D(\mathrm{Ber}(q^\star_\nu), \mathrm{Ber}(q^\star_\lambda)).$$

Define $f^\star \in \mathcal{F}_{\mathrm{ctx}}$ by $f^\star(a) = 0.1$ and $f^\star(b) = 0.9$. Prove that, for every $\rho \in [0,1]$ and $f \in \mathcal{F}_{\mathrm{ctx}}$,

$$L^{\mathrm{sq}}_{\rho}(f) - L^{\mathrm{sq}}_{\rho}(f^\star) = \rho(f(a) - 0.1)^2 + (1-\rho)(f(b) - 0.9)^2$$

and

$$L^{\log}_{\rho}(f) - L^{\log}_{\rho}(f^\star) = \rho D(\mathrm{Ber}(0.1), \mathrm{Ber}(f(a))) + (1-\rho) D(\mathrm{Ber}(0.9), \mathrm{Ber}(f(b))).$$

Use these identities to prove that $f^\star$ minimizes both risks over $\mathcal{F}_{\mathrm{ctx}}$ for every ρ. Identify the restriction to $\mathcal{F}_{\mathrm{const}}$ as the source of the nonzero deployment excess risks.

Hints For (a), first determine $\mathbb{P}(Y = 1)$. For the $\mathcal{F}_{\mathrm{ctx}}$ identities, consider the risk at each context separately.

References

Edward S. Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Dinesh Jayaraman, Alex Lamb, and John Langford. Learning to achieve goals with belief state transformers. 2025. https://arxiv.org/abs/2410.23506.

Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: Unveiling how chain-of-thought enhances transformer sample efficiency. International Conference on Learning Representations, 2025. https://arxiv.org/abs/2410.05459.

Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative machine teaching. International Conference on Machine Learning, 2017. https://proceedings.mlr.press/v70/liu17b.html.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Itamar Shoshani and Ohad Shamir. Hardness of learning fixed parities with neural networks. 2025. https://arxiv.org/abs/2501.00817.

Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. AAAI Conference on Artificial Intelligence, 2015.