Machine Learning 1

Course Overview and Recap

Tin Hadzi Veljkovic

October 14, 2025

A map for the course

Two lenses

- **Probabilistic view**: write down $p(\cdot)$, pick a likelihood, optionally a prior; infer via **ML/MAP/Bayesian** to reason about *uncertainty*.
- **Algorithmic view**: set up a clear objective; optimize it (often geometry/margins); probabilities optional.

Modeling and tools

- Distributions: Gaussian, Bernoulli, categorical
- Optimization: GD/SGD, constraints (Lagrange & KKT)
- Linear algebra: eigendecomp/SVD; kernels

Probability essentials

- Random variables, joint/conditional probability, Bayes' theorem:

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{p(\mathcal{D})}, \quad p(\mathcal{D}) = \int p(\mathcal{D} \mid \theta) p(\theta)\, d\theta.$$

- Expectations and second moments:

$$\mathbb{E}[X], \quad \mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2, \quad \mathrm{Cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

- Gaussian distribution: workhorse for noise models and conjugacy.

MLE, MAP, and the Bayesian spectrum

MLE:

$$\hat{\theta}_{\text{MLE}} =_{\theta} \log p(\mathcal{D} \mid \theta).$$

MAP:

$$\hat{\theta}_{\text{MAP}} =_{\theta} \log p(\mathcal{D} \mid \theta) + \log p(\theta).$$

Fully Bayesian: keep *posterior* $p(\theta \mid \mathcal{D})$ and average:

$$p(y \mid x, \mathcal{D}) = \int p(y \mid x, \theta)\, p(\theta \mid \mathcal{D})\, d\theta.$$

- MAP $\equiv$ regularization in optimization.
- Priors $\rightarrow$ inductive bias.

From probability to models

From probability to models: common patterns)

Recipe

1. **Choose a likelihood** $p(\text{data} \mid \theta)$ that encodes how data is generated (e.g., $\mathcal{N}(t \mid y(x, \theta), \beta^{-1})$ for real-valued targets; Bernoulli/Categorical for labels).
2. Optionally choose a **prior** $p(\theta)$ to encode inductive bias.
3. Fit **by ML** ($\hat{\theta}_{\text{ML}} = \arg\max_\theta p(D \mid \theta)$), **MAP** ($\hat{\theta}_{\text{MAP}} = \arg\max_\theta p(\theta \mid D)$), or go **Bayesian** and *average* via $p(\theta \mid D)$.
4. Derive a **predictive** $p(\tilde{y} \mid \tilde{x}, D)$ - uncertainties naturally come up.

Unifying perspective

- **Linear regression**: MLE $\Rightarrow$ least squares; Gaussian prior $\Rightarrow$ ridge (MAP).
- **Logistic regression**: Bernoulli likelihood $\Rightarrow$ *cross-entropy* objective (MLE).
- **K-means** $\approx$ *hard-assignment limit* of spherical GMM + EM.

Linear regression & model selection

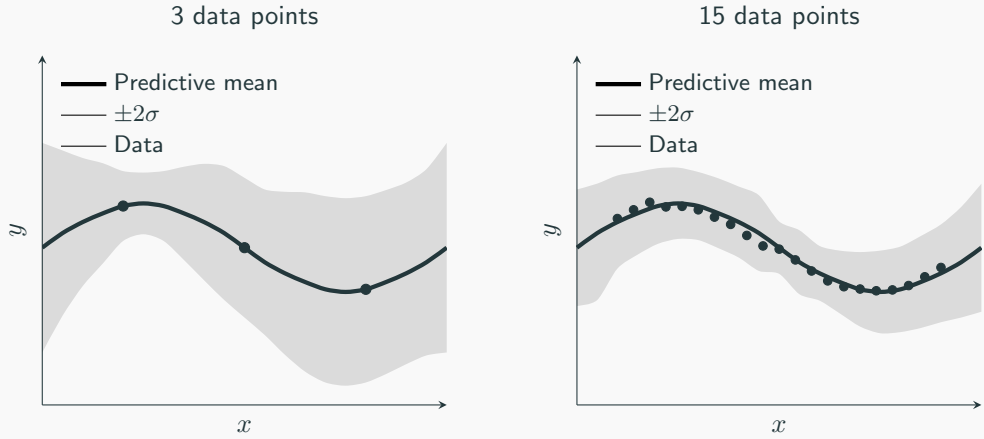
Linear regression: basis functions & regularization

Model $y(x) = \mathbf{w}^\top \phi(x)$ with basis $\phi(x)$ (polynomial, Gaussian, splines, etc.).

$$\underbrace{\mathbf{w}_{\text{ML}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_i (t_i - \mathbf{w}^\top \phi(x_i))^2}_{\text{Gaussian MLE}} \qquad \underbrace{\mathbf{w}_{\text{MAP}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_i (t_i - \mathbf{w}^\top \phi(x_i))^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2}_{\text{Ridge (Gaussian prior)}}$$

- ℓ_2 (ridge) shrinks weights; ℓ_1 (lasso) encourages sparsity - both can be viewed as constrained optimization problems, with the penalty interpreted as a Lagrange multiplier.
- Bayesian linear regression yields *posterior* over $\mathbf{w}$ and a predictive variance that shrinks near data.

Bayesian linear regression: predictive mean & uncertainty



Bias–variance decomposition (intuition & math)

Setup (squared loss) We imagine repeatedly drawing different datasets D from the same underlying data distribution $p(x, t)$, and fitting a model each time to get predictors $\hat{y}_D(x)$. The total expected error of such a learning procedure can be decomposed as:

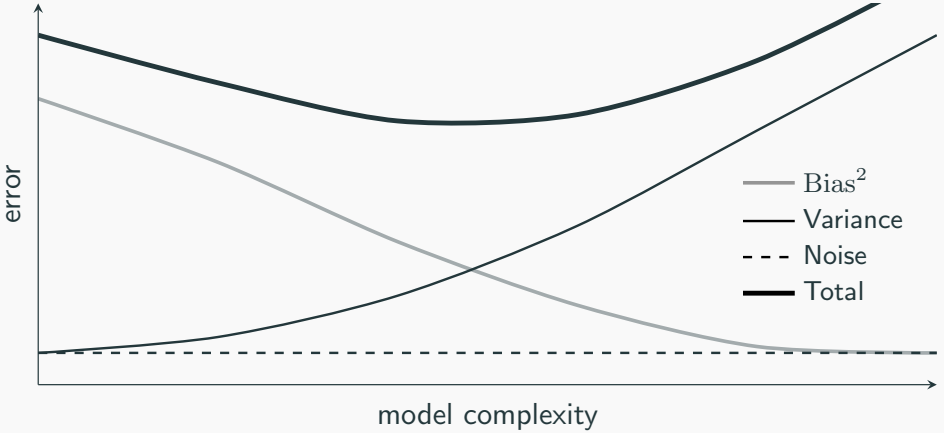
$$\underbrace{\mathbb{E}_{x,t}\big[(\hat{y}_D(x) - t)^2\big]}_{\text{expected prediction error}} = \underbrace{\mathbb{E}_x\big[(\mathbb{E}_D[\hat{y}_D(x)] - f(x))^2\big]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_x[\mathrm{Var}_D(\hat{y}_D(x))]}_{\text{Variance}} + \underbrace{\mathbb{E}_x[\sigma^2(x)]}_{\text{Irreducible noise}},$$

where $f(x) = \mathbb{E}[t \mid x]$ and $\sigma^2(x) = \mathrm{Var}[t \mid x]$.

Interpretation

- The outer expectation averages over test points x; the inner one over different datasets D.
- **Bias** measures how far the mean prediction $\mathbb{E}_D[\hat{y}_D(x)]$ is from the true function $f(x)$.
- **Variance** measures how much the predictions fluctuate across different training sets.
- The **noise** term captures randomness inherent in the data and cannot be reduced by modeling.

Bias–variance: typical curves



Decision theory & classification

Decision theory: three families you met

$p(x, y)$

1) Generative probabilistic (*model* $p(x \mid y)$ and $p(y)$): e.g., LDA/QDA, Naive Bayes. $p(y \mid x) \propto p(x \mid y)p(y)$; can sample, impute, handle missing features.

2) Discriminative probabilistic (*model* $p(y \mid x)$ directly): e.g., **logistic regression**. MLE $\Rightarrow$ **cross-entropy** loss. Probabilities calibrated.

3) Decision-function (algorithmic) (no explicit probabilities): e.g., **SVM**, perceptron, large-margin methods. Optimize a margin/hinge objective under constraints; care about geometry.

Cross-entropy appears naturally (binary and multiclass)

Binary case (Bernoulli) $\Rightarrow$ Binary Cross-Entropy (BCE). For labels $t_i \in \{0, 1\}$ and predicted probability $\pi(x) \in (0, 1)$,

$$p(t_i \mid x_i) = \pi(x_i)^{t_i} \left[1 - \pi(x_i)\right]^{1-t_i}, \qquad \ell_{\mathsf{BCE}}(\theta) = -\sum_i \Big[t_i \log \pi(x_i) + (1 - t_i) \log(1 - \pi(x_i))\Big].$$

Multiclass case (Categorical) $\Rightarrow$ Cross-Entropy (CE). For one-hot labels t_{ik}, $\sum_k t_{ik} = 1$, and class probabilities $\pi_k(x)$ satisfying $\sum_k \pi_k(x) = 1$,

$$p(t_i \mid x_i) = \prod_k \pi_k(x_i)^{t_{ik}} \quad \Rightarrow \quad \ell_{\mathsf{CE}}(\theta) = -\sum_i \sum_k t_{ik} \log \pi_k(x_i).$$

Interpretation.

- Cross-entropy is the negative log-likelihood for independent categorical observations.
- The softmax parameterization is one convenient way to ensure $\pi_k(x)$ are valid probabilities, but not required.

Neural networks

Neural networks in context

- Neural nets are flexible function approximators built by stacking linear layers and nonlinearities to learn useful representations from data.
- Training uses stochastic gradient descent; backpropagation is just reverse-mode autodiff through the computation graph.
- In this view, neural nets extend linear and logistic regression by replacing fixed features with learned ones.

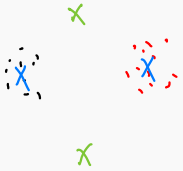
Unsupervised: K-means, GMM, EM

K-means: geometry and intuition

Goal: partition data into K clusters by finding centers $\{\mu_k\}$ that minimize within-cluster variance.

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \, \|x_i - \mu_k\|^2, \quad r_{ik} \in \{0, 1\}, \; \sum_k r_{ik} = 1.$$

Algorithm (alternating minimization):



1. **Assign:** $r_{ik} = 1$ for the nearest μ_k.
2. **Update:** $\mu_k = \frac{1}{N_k} \sum_i r_{ik} x_i$ where $N_k = \sum_i r_{ik}$.

Interpretation:

- Finds a local minimum of J — a *hard-assignment* clustering.
- Equivalent to fitting spherical clusters of equal size (no probabilities involved).
- Sensitive to initialization and scale; multiple restarts are common.

K-means visualized

Gaussian Mixture Models (GMM): definition

Model. $p(x) = \sum_z p(x, z) = \sum_z p(x \mid z) \cdot p(z)$

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \qquad \pi_k \geq 0, \ \sum_{k=1}^{K} \pi_k = 1.$$

Latent variable. Introduce $z_i \in \{1, \ldots, K\}$ indicating the component for x_i:

$$p(x_i, z_i = k) = \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k), \qquad p(x_i) = \sum_k p(x_i, z_i = k).$$

Soft assignments (responsibilities).

$$\gamma_{ik} \ := \ p(z_i = k \mid x_i) = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}.$$

Fitting. Max. likelihood with latent z uses **EM**: alternate computing γ_{ik} and updating (μ_k, Σ_k, π_k).

GMM: constraints and the EM algorithm

Why constraints? Mixture weights must satisfy $\sum_k \pi_k = 1$ and each $\pi_k \geq 0$. Use **Lagrange multipliers** to enforce $\sum_k \pi_k = 1$ in the M-step.

EM steps (soft assignments):

E-step: $\gamma_{ik} \leftarrow \dfrac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}$,

M-step:

$$\begin{cases} \mu_k \leftarrow \dfrac{\sum_i \gamma_{ik} x_i}{\sum_i \gamma_{ik}} \\ \Sigma_k \leftarrow \dfrac{\sum_i \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_i \gamma_{ik}} \\ \pi_k \leftarrow \dfrac{1}{N} \sum_i \gamma_{ik} \quad (\sum_k \pi_k = 1) \end{cases}$$

PCA done three ways

PCA I — variance maximization view

Goal: find a direction $\mathbf{u}$ onto which the projected data $y_i = \mathbf{u}^\top (x_i - \bar{x})$ has maximal variance.

$$\mathrm{Var}(y) = \mathbf{u}^\top \Big(\frac{1}{N} \sum_i (x_i - \bar{x})(x_i - \bar{x})^\top \Big) \mathbf{u} = \mathbf{u}^\top \Sigma \mathbf{u}, \quad \Sigma = \frac{1}{N} \sum_i (x_i - \bar{x})(x_i - \bar{x})^\top.$$

Optimization problem:

$$\max_{\mathbf{u}} \ \mathbf{u}^\top \Sigma \mathbf{u} \quad \text{s.t.} \quad \|\mathbf{u}\|^2 = 1.$$

Lagrangian:

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{u}^\top \Sigma \mathbf{u} - \lambda(\mathbf{u}^\top \mathbf{u} - 1).$$

Stationarity condition:

$$\nabla_{\mathbf{u}} \mathcal{L} = 2\Sigma \mathbf{u} - 2\lambda \mathbf{u} = 0 \quad \Rightarrow \quad \Sigma \mathbf{u} = \lambda \mathbf{u}.$$

Thus, $\mathbf{u}$ is an **eigenvector** of Σ, and maximizing variance chooses the eigenvector with the largest eigenvalue λ_1. Projecting to m dimensions uses $\mathbf{U}_m = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$:

PCA II — reconstruction view

Goal: find a low-dimensional subspace that best reconstructs the data in the least-squares sense:

$$\min_{\mathbf{U}_m,\,\{z_i\}} \frac{1}{N}\sum_{i=1}^{N} \|x_i - \bar{x} - \mathbf{U}_m z_i\|_2^2 \quad \text{s.t.} \quad \mathbf{U}_m^\top \mathbf{U}_m = I.$$

Intuition.

- We seek m orthogonal directions (columns of $\mathbf{U}_m$) that define a flat subspace through the mean.
- Each x_i is approximated by its closest point in this subspace, $\hat{x}_i = \bar{x} + \mathbf{U}_m \mathbf{U}_m^\top (x_i - \bar{x})$.
- Plugging this back gives the same objective as in the variance-maximization view.

Result.

- Columns of $\mathbf{U}_m$ are the top m eigenvectors of $\Sigma = \frac{1}{N}\sum_i (x_i - \bar{x})(x_i - \bar{x})^\top$.
- Minimizing reconstruction error $\Leftrightarrow$ maximizing explained variance.

PCA III — probabilistic PCA (pPCA)

Latent-variable generative model:

$$x = \mathbf{W}h + \mu + \epsilon, \qquad h \sim \mathcal{N}(0, I), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

Marginal distribution of x:

$$p(x) = \mathcal{N}\big(x \mid \mu, \mathbf{C}\big), \qquad \mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2 I.$$

Interpretation:

- Latent variable h lives in a low-dimensional space; $\mathbf{W}$ maps it to the data space.
- $\sigma^2 I$ models isotropic Gaussian noise around the low-dimensional manifold.
- MLE yields $\mathbf{W}$ whose column space coincides with the classical PCA subspace.

Kernels and SVMs

Kernels, feature maps, and the Representer idea

A **kernel** $k(x, x') = \phi(x)^\top \phi(x')$ lets us work in high/infinite-dimensional feature spaces implicitly.

Representer theorem (intuition): many regularized objectives yield solutions in the span of training points, $f(x) = \sum_i \alpha_i k(x, x_i)$.

Why it matters: we can (i) separate with complex boundaries, (ii) carry geometry into GPs.

From constrained optimization to the dual ($g_i(\mathrm{w}) \geq 0$)

General form:

$$\min_{\mathbf{w}} f(\mathbf{w}) \quad \text{s.t.} \quad g_i(\mathbf{w}) \geq 0.$$

1. Lagrangian

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\alpha}) = f(\mathbf{w}) - \sum_i \alpha_i g_i(\mathbf{w}), \qquad \alpha_i \geq 0.$$

2. Dual problem

$$\mathcal{D}(\boldsymbol{\alpha}) = \inf_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \boldsymbol{\alpha}), \qquad \max_{\boldsymbol{\alpha} \geq 0} \mathcal{D}(\boldsymbol{\alpha}).$$

The dual provides an equivalent (often simpler) optimization problem.

3. KKT conditions (for convex problems):

$$\begin{cases} \nabla_{\mathbf{w}} \mathcal{L} = 0 & \text{(stationarity)} \\ g_i(\mathbf{w}) \geq 0, \ \alpha_i \geq 0 & \text{(feasibility)} \\ \alpha_i g_i(\mathbf{w}) = 0 & \text{(complementary slackness)} \end{cases}$$

KKT

Soft-margin SVM: primal → dual via Lagrangian

Primal problem:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}\geq 0} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_n \xi_n \quad \text{s.t.} \quad t_n(\mathbf{w}^\top \phi(x_n) + b) - 1 + \xi_n \geq 0. \quad , \ \xi_n \geq 0$$

Lagrangian:

$$\mathcal{L} = \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_n \xi_n - \sum_n \alpha_n [\, t_n(\mathbf{w}^\top \phi(x_n) + b) - 1 + \xi_n \,] - \sum_n \mu_n \xi_n, \quad \alpha_n, \mu_n \geq 0.$$

Stationarity (eliminate primal variables):

$$\nabla_{\mathbf{w}}\mathcal{L} = 0 \Rightarrow \mathbf{w} = \sum_n \alpha_n t_n \phi(x_n), \quad \nabla_b \mathcal{L} = 0 \Rightarrow \sum_n \alpha_n t_n = 0,$$

$$\nabla_{\xi_n}\mathcal{L} = 0 \Rightarrow C - \alpha_n - \mu_n = 0 \ \Rightarrow \ 0 \leq \alpha_n \leq C.$$

Dual problem (substitute back):

$$\max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \tfrac{1}{2}\sum_{n,m} \alpha_n \alpha_m t_n t_m \, k(x_n, x_m) \quad \text{s.t.} \quad 0 \leq \alpha_n \leq C, \ \sum_n \alpha_n t_n = 0.$$

Gaussian Processes (GPs)

How GPs arise & why they're useful

Two roads to GPs:

1. **Bayesian linear regression** with many basis functions $\phi(\cdot)$ and prior $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$:

   $$f(x) = \mathbf{w}^\top \phi(x), \quad m_\star = \tfrac{1}{\sigma^2}\phi(x_\star)^\top S_N \Phi^\top \mathbf{t}, \quad S_N^{-1} = \Sigma_p^{-1} + \tfrac{1}{\sigma^2}\Phi^\top \Phi.$$

   This can be seen as a *weighted sum over training points*:

   $$m_\star = \sum_n k_{\mathsf{eq}}(x_\star, x_n) t_n, \quad k_{\mathsf{eq}}(x_\star, x_n) = \tfrac{1}{\sigma^2}\phi(x_\star)^\top S_N \phi(x_n).$$

   As the number of basis functions $\to \infty$, $f(\cdot)$ has a GP prior with covariance $k(x, x') = \phi(x)^\top \Sigma_p \phi(x')$.

2. **Function view:** a GP is a *distribution over functions*,

   $$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)),$$

   so any finite $(f(x_1), \ldots, f(x_N))$ is jointly Gaussian.

GP regression in one slide

Prior: $f \sim \mathcal{GP}(0, k)$. Observations: $t_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Posterior predictive at $x_\star$:

$$\underbrace{m_\star}_{\mathbb{E}[t_\star|D]} = \mathbf{k}_\star^\top (K + \sigma^2 I)^{-1} \mathbf{t}, \qquad \underbrace{v_\star}_{\mathrm{Var}[t_\star|D]} = k(x_\star, x_\star) - \mathbf{k}_\star^\top (K + \sigma^2 I)^{-1} \mathbf{k}_\star.$$

Here $K_{ij} = k(x_i, x_j)$ and $\mathbf{k}_\star = [k(x_1, x_\star), \ldots, k(x_N, x_\star)]^\top$.

Intuition: *interpolate with uncertainty*; variance shrinks near data, grows away.

The end 💀