Supervised Learning II

Fundamentals of Machine Learning

Xufeng Zhang

Inria
xufeng.zhang@inria.fr

December 11, 2025

Outline

1. Supervised Learning Overview

2. Generalized Linear Models

3. Random Forests

4. Support Vector Machines

5. Putting It All Together

Supervised Learning: Big Picture

- **Goal:** Learn a mapping from inputs x to outputs y from labeled data.
- **Input:** Training set $\{(x_i, y_i)\}_{i=1}^{n}$.
- **Output:** A prediction function $\hat{f}(x)$.
- **Tasks:**
  - Regression: y is continuous.
  - Classification: y is categorical (binary or multi-class).
- **This lecture:**
  - Generalized Linear Models (GLMs)
  - Random Forests
  - Support Vector Machines (SVMs)

Basic Supervised Learning Pipeline

1. Problem definition and data understanding.
2. Data preprocessing:
   - Cleaning, handling missing values.
   - Feature engineering and scaling (if needed).
3. Model choice: GLM, Random Forest, SVM, etc.
4. Training and hyperparameter tuning.
5. Model evaluation (train/validation/test).
6. Deployment and monitoring.

Types of Models in This Lecture

- **Generalized Linear Models**
  - Logistic regression (binary response).
  - Poisson regression (count data).
- **Random Forests**
  - Ensembles of decision trees.
  - Strong non-linear models, robust to noise.
- **Support Vector Machines**
  - Maximum-margin classifiers.
  - Kernel trick for non-linear decision boundaries.

Bias–Variance View

- Simple models (e.g., linear regression, GLMs):
    - Higher bias, lower variance.
    - Easier to interpret.
- Complex models (e.g., RF, SVM with complex kernels):
    - Lower bias, higher variance if not regularized.
    - Often better predictive accuracy.
- Need to **balance** bias and variance via:
    - Model choice.
    - Regularization.
    - Ensemble methods.

From Linear Models to GLMs

- Ordinary linear regression assumes:
  - Continuous response $y \in \mathbb{R}$.
  - Gaussian noise with constant variance.
  - Linear relationship: $y = \beta_0 + x^\top \beta + \varepsilon$.
- Many real-world responses are:
  - Binary (0/1).
  - Counts (0, 1, 2, ...).
  - Positive, skewed quantities.
- **Generalized Linear Models** relax these assumptions.

Core Components of a GLM

A GLM has three key components:

1. **Random component**:
   - Distribution of Y from the exponential family
   - e.g. normal, binomial, Poisson, gamma.
2. **Systematic component**:

$$\eta = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

3. **Link function**:

$$g(\mu) = \eta, \quad \mu = \mathbb{E}[Y \mid X]$$

Exponential Family Distributions

- Many common distributions can be written as:

$$f_Y(y; \theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right)$$

- Examples:
  - Normal (for continuous y).
  - Binomial / Bernoulli (for binary y).
  - Poisson (for counts).
  - Gamma (for positive continuous variables).
- GLMs tie mean and variance through this family.

Link Functions

- **Link function** $g(\cdot)$ connects mean μ and linear predictor η:

$$g(\mu) = \eta$$

- Canonical link: arises naturally from the exponential family form.
- Examples:

Identity: $g(\mu) = \mu$ (linear regression).
Logit: $g(\mu) = \log \frac{\mu}{1-\mu}$ (logistic regression).
Log: $g(\mu) = \log \mu$ (Poisson regression).

Types of GLMs

- **Linear regression**
  - Normal errors, identity link.
- **Logistic regression**
  - Binomial errors, logit link.
- **Poisson regression**
  - Poisson errors, log link.
- **Gamma regression**
  - Gamma errors, log or inverse link.

Logistic Regression: Problem Setup

- Binary outcome:

$$y_i \in \{0, 1\}$$

- We model the probability of the event:

$$p_i = \mathbb{P}(y_i = 1 \mid x_i)$$

- Examples:
  - Customer churn (churn / no churn).
  - Disease status (sick / healthy).
  - Credit default (default / no default).

Logistic Regression: Model

- Use the **logit link**:

$$\operatorname{logit}(p_i) = \log \frac{p_i}{1 - p_i} = \beta_0 + x_i^\top \beta.$$

- Equivalently:

$$p_i = \frac{1}{1 + \exp\big(- (\beta_0 + x_i^\top \beta)\big)}.$$

- Interpretation:
    - Coefficients act on the log-odds.
    - $\exp(\beta_j)$ is the multiplicative change in odds when x_j increases by 1.

Logistic Regression: Decision Rule and Loss

- **Decision rule** for binary classification:

$$\hat{y}_i = \begin{cases} 1, & p_i \geq 0.5, \\ 0, & p_i < 0.5. \end{cases}$$

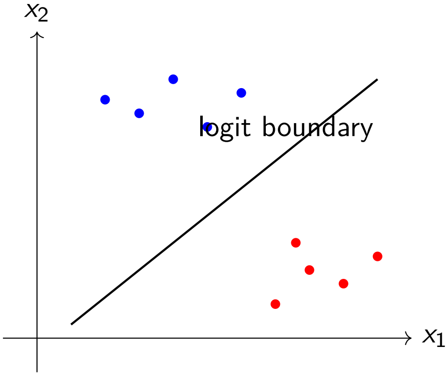
  (Threshold can be adjusted.)

- **Loss function:** negative log-likelihood (cross-entropy)

$$\ell(\beta) = -\sum_{i=1}^{n} \big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \big].$$

- Parameters are usually estimated via gradient-based optimization.

Logistic Regression: Simple Decision Boundary



Logistic Regression: Applications

- Medicine:
  - Predict probability of disease given patient features.
- Marketing:
  - Predict probability of a customer responding to a campaign.
- Finance:
  - Credit scoring and default prediction.
- Advantages:
  - Probabilistic output.
  - Coefficients interpretable as log-odds.

Poisson Regression: Count Data

- Outcome y_i is a non-negative **count**:

$$y_i \in \{0, 1, 2, \dots\}, \quad y_i \sim \text{Poisson}(\mu_i).$$

- Mean equals variance:

$$\mathbb{E}[y_i] = \mu_i, \quad \text{Var}(y_i) = \mu_i.$$

- Examples:
  - Number of calls per hour at a call center.
  - Number of accidents at an intersection per month.
  - Number of website visits per day.

Poisson Regression: Model

- Use the **log link**:

$$\log \mu_i = \beta_0 + x_i^\top \beta$$

so that

$$\mu_i = \exp(\beta_0 + x_i^\top \beta).$$

- Interpretation:
    - $\exp(\beta_j)$ is the multiplicative change in expected count per unit change in x_j.
- Often includes an **offset** term for exposure (e.g., time at risk).

Poisson Regression: Assumptions and Extensions

- Assumptions:
    - Counts arise from Poisson process.
    - Mean equals variance (no overdispersion).
- If data are **overdispersed** (variance $>$ mean):
    - Use quasi-Poisson or negative binomial regression.
- Model checking:
    - Residual plots.
    - Compare observed and fitted counts.

GLM: Advantages

- **Flexibility** in handling different response types.
- **Interpretability**:
  - Coefficients have clear statistical meaning.
- **Scalability** to reasonably large datasets.
- **Statistical inference**:
  - Hypothesis tests, confidence intervals.
- Fits naturally into regression framework.

GLM: Limitations

- Requires **correct choice of distribution** and link:
  - Mis-specification can bias results.
- **Linearity** in predictors (after link) may be too restrictive.
- Sensitive to **outliers** and influential points.
- Requires reasonable sample size for asymptotic theory to hold.
- Can underfit complex non-linear relationships.

GLM in Practice: Workflow

1. Choose response distribution (binomial, Poisson, etc.).
2. Select link function (often canonical).
3. Specify linear predictor and interactions.
4. Fit model (e.g., maximum likelihood).
5. Check diagnostics and goodness-of-fit.
6. Refine model (add/remove predictors, transform variables).

GLM Diagnostics

- Residual analysis:
  - Deviance residuals.
  - Pearson residuals.
- Goodness-of-fit measures:
  - Deviance, AIC, BIC.
- Influence diagnostics:
  - Cook’s distance, leverage.
- Predictive checks:
  - Cross-validation or test-set performance.

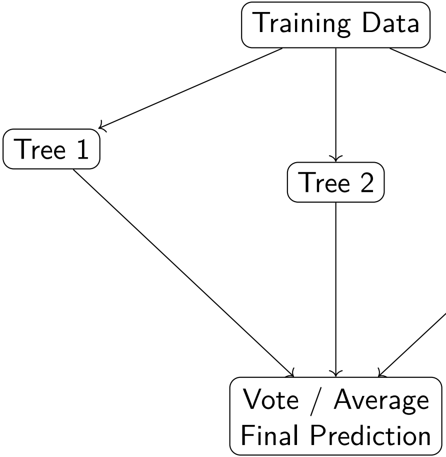
Decision Trees: Quick Recap

- Tree recursively splits feature space into regions.
- At each node:
    - Choose feature and split threshold.
    - Optimize impurity measure (Gini, entropy, MSE).
- Prediction:
    - Classification: majority class in leaf.
    - Regression: average response in leaf.
- Simple and interpretable, but high variance.

Random Forest: Intuition

- Build many decision trees and **aggregate** them.
- Two sources of randomness:
    1. Bootstrap sampling of training data for each tree.
    2. Random subset of features considered at each split.
- Predictions:
    - Classification: majority vote across trees.
    - Regression: average prediction across trees.

Random Forest: Simple Diagram



Training Data

Tree 1

Tree 2

Tree B

Vote / Average
Final Prediction

Random Forest: Training Procedure

1. For $b = 1, \ldots, B$ (number of trees):
   - Sample a bootstrap dataset from training data.
   - Grow a decision tree:
     - At each split, randomly choose m features.
     - Choose best split among those m features.
2. For prediction:
   - Aggregate predictions from all trees.

Random Forest: Hyperparameters

- Number of trees B.
- Number of features m considered at each split.
- Maximum depth of each tree.
- Minimum samples per leaf.
- Criterion: Gini, entropy (classification) or MSE (regression).

Random Forest: Variable Importance

- Measure how much each feature contributes to prediction.
- Common approaches:
  - Mean decrease in impurity.
  - Permutation importance:
    - Shuffle a feature across samples.
    - Measure increase in prediction error.
- Useful for feature selection and model interpretation.

Random Forest: Advantages

- High predictive accuracy for many tasks.
- Handles nonlinear relationships and interactions automatically.
- Robust to outliers and noise.
- Works with mixed feature types (numeric and categorical).
- Built-in estimate of generalization error via out-of-bag (OOB) samples.

Random Forest: Drawbacks

- Reduced interpretability compared to a single tree.
- Large models:
  - Higher memory usage.
  - Slower prediction for very large forests.
- May perform worse than simple linear models on very high-dimensional sparse data.
- Cannot extrapolate beyond the range of training responses.

Random Forest: Practical Tips

- Use enough trees to stabilize OOB error.
- Tune:
    - Number of features *m*.
    - Maximum depth and minimum samples per leaf.
- For highly imbalanced data:
    - Use class weights or balanced subsampling.
- Use feature importance to reduce dimensionality if needed.

SVM: Motivation

- For linearly separable data, many separating hyperplanes exist.
- **Support Vector Machine** chooses the one with the **maximum margin**.
- Intuition:
  - Larger margin $\Rightarrow$ better generalization.
- Works very well in high-dimensional spaces.

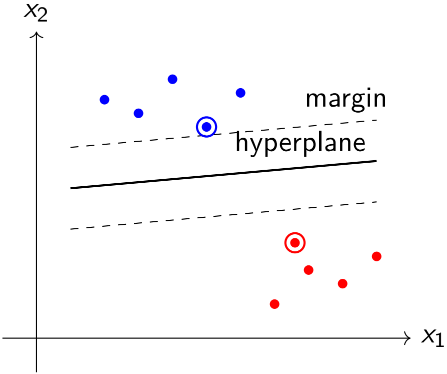
Key Terms in SVM

- **Hyperplane**:

$$w^\top x + b = 0$$

- **Margin**: distance between hyperplane and closest data points.
- **Support vectors**: data points lying on the margin boundaries.

SVM Geometry (2D Illustration)



Linear SVM: Hard-Margin Formulation

- Assume data are linearly separable.
- Labels $y_i \in \{-1, +1\}$.
- Optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to

$$y_i(w^\top x_i + b) \geq 1, \quad i = 1, \ldots, n.$$

- Maximizes margin $2/\|w\|$.

Soft-Margin SVM

- Real data are often not perfectly separable.
- Introduce **slack variables** $\xi_i \geq 0$:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i.$$

- Optimization:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

  where $C > 0$ controls trade-off between margin and misclassification.

Effect of Parameter C

- Large C:
  - High penalty for misclassification.
  - Smaller margin, can overfit.
- Small C:
  - Allows more misclassifications.
  - Larger margin, can underfit.
- Tune C via cross-validation.

Nonlinearly Separable Data

- In many problems, no linear hyperplane can separate classes well.
- Idea: map inputs to a **higher-dimensional feature space**

  $$\phi : \mathbb{R}^d \to \mathcal{H}$$

  where data may become linearly separable.
- Directly computing $\phi(x)$ may be very expensive or infinite-dimensional.

Kernel Trick

- In SVM, data appear only as inner products:

$$\phi(x_i)^\top \phi(x_j).$$

- Use a **kernel function**

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

  without explicitly computing ϕ.
- Allows efficient learning in high- or infinite-dimensional feature spaces.

Common SVM Kernels

- **Linear kernel**

$$K(x, z) = x^\top z$$

- **Polynomial kernel**

$$K(x, z) = (x^\top z + c)^d$$

- **Radial Basis Function (RBF) kernel**

$$K(x, z) = \exp\big(- \gamma \|x - z\|^2 \big)$$

- **Sigmoid kernel**

$$K(x, z) = \tanh(\kappa x^\top z + c)$$

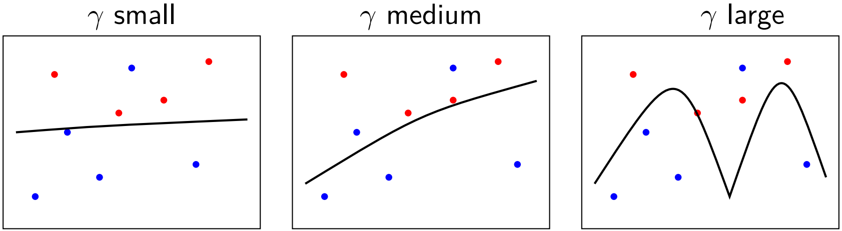
RBF Kernel and γ

- RBF kernel:

$$K(x, z) = \exp\big(-\gamma\|x - z\|^2\big).$$

- Parameter γ controls how far each training point influences the decision boundary:
    - Small γ: wide influence (smoother, more linear boundary).
    - Large γ: narrow influence (more complex, wiggly boundary).
- Tune γ together with C via grid search or cross-validation.

Effect of γ (Qualitative Illustration)



γ small

γ medium

γ large

SVM: Advantages

- Effective in high-dimensional spaces.
- Often works well even when number of features $p > n$.
- Margin-based formulation can yield good generalization.
- Flexible via different kernels (linear, RBF, etc.).

SVM: Drawbacks

- Training can be slow for very large datasets.
- Choice of kernel and hyperparameters (C, γ) is critical.
- Less interpretable than linear models and simple trees.
- Probabilistic outputs require additional calibration.

Real-World Applications of SVM

- Image recognition and handwriting recognition.
- Text classification and spam filtering.
- Bioinformatics (e.g., gene expression classification).
- Fraud detection and anomaly detection.

GLM vs Random Forest vs SVM

	GLM	Random Forest	SVM
Nonlinearity	Low	High	High
Interpretability	High	Medium–Low	Medium
Feature scaling needed	Sometimes	Rarely	Often
Handles high dimension	Moderate	Moderate	Good
Works well on small n	Good	OK	Good

When to Use Which?

- **Use GLM when:**
  - You need clear interpretability and statistical inference.
  - The relationship is roughly linear after transformation.
- **Use Random Forest when:**
  - You care mainly about predictive performance.
  - There are complex interactions and nonlinearities.
  - You have mixed data types and moderate data size.
- **Use SVM when:**
  - You have high-dimensional data.
  - You can afford careful kernel and hyperparameter tuning.

Example Scenario: Credit Risk

- Predict whether a borrower will default on a loan.
- Possible models:
  - Logistic regression (GLM):
    - Clear interpretation of risk factors.
  - Random Forest:
    - Strong predictive performance, feature importance.
  - SVM:
    - Effective when feature space is high-dimensional.
- In practice:
  - Compare multiple models and choose based on performance, interpretability, and regulatory constraints.

Summary

- GLMs generalize linear models to non-Gaussian responses via link functions.
- Random Forests are powerful ensembles of decision trees with built-in averaging.
- SVMs are maximum-margin classifiers that can use kernels for nonlinear decision boundaries.
- Model choice depends on:
  - Data size and dimensionality.
  - Need for interpretability vs pure accuracy.
  - Computational resources and time.

Further Reading

- Textbooks on statistical learning and machine learning.
- Library documentation (e.g., scikit-learn) for practical usage.
- Research papers and case studies in your application domain.

Thank you!