Machine Learning Homework 3

April 25, 2025

2023141460251 Haoxiang Sun

Question 7: Why is the dual form of SVM important?

7.1 Primal Problem and Lagrangian

Starting from the (soft-margin) primal problem of SVM:

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

$$\text{s.t.} \quad y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) \geq 1 - \xi_i,\ \xi_i \geq 0, \quad i = 1, \dots, n. \tag{2}$$

Introduce Lagrange multipliers $\alpha_i \geq 0$ for (2) and $\mu_i \geq 0$ for the slack constraints. The Lagrangian is

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\big[y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i\big] - \sum_{i=1}^{n}\mu_i\,\xi_i. \tag{3}$$

7.2 KKT Stationarity Conditions

Stationarity with respect to $\mathbf{w}$, b, and ξ_i yields:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i\,\mathbf{x}_i, \tag{4}$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{n}\alpha_i y_i = 0, \tag{5}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \implies \alpha_i + \mu_i = C \implies 0 \leq \alpha_i \leq C. \tag{6}$$

7.3 Dual Quadratic Program

Substituting equations (4)–(6) back into (3) gives the dual quadratic program:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j\, y_i y_j\,\langle\mathbf{x}_i, \mathbf{x}_j\rangle, \tag{7}$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C,\ i = 1, \dots, n. \tag{8}$$

The decision function becomes:

$$f(\mathbf{x}) = \operatorname{sign}\Big(\sum_{i=1}^{n} \alpha_i\, y_i\, K(\mathbf{x}_i, \mathbf{x}) + b\Big), \tag{9}$$

where in the linear case $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

7.4 Key Advantages

1. **Kernel Trick.** Replace the inner product with any positive-definite kernel:

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle \to K(\mathbf{x}_i, \mathbf{x}_j).$$

2. **Sparsity.** Only support vectors (those with $\alpha_i > 0$) contribute:

$$\mathbf{w} = \sum_{i:\alpha_i>0} \alpha_i\, y_i\, \mathbf{x}_i.$$

3. **Dimensionality Benefit.** Solving the dual in $\mathbb{R}^n$ can be more efficient than the primal in $\mathbb{R}^d$ when $n < d$.

4. **Convexity.** The dual is a convex QP with simple box and equality constraints, guaranteeing a unique global optimum.

5. **Easy Extensions.** Variants (e.g. ℓ_1-SVM, ν-SVM) often require only modifying the dual.

Question 8: How to derive the dual form from the primal form of SVM?

8.1 Primal Problem

The soft-margin SVM primal problem (same as (1)–(2)):

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

$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{2}$$

8.2 Lagrangian Formation

Introduce multipliers $\alpha_i, \mu_i \geq 0$ to form the Lagrangian:

$$L = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i\big[y_i(w^\top x_i + b) - 1 + \xi_i\big] - \sum_i \mu_i\, \xi_i. \tag{3}$$

8.3 KKT Stationarity

Applying stationarity yields:

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_i \alpha_i y_i x_i, \tag{4}$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_i \alpha_i y_i = 0, \tag{5}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \implies 0 \le \alpha_i \le C. \tag{6}$$

8.4 Dual Derivation

Substituting (4)–(6) into (3) gives:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \langle x_i, x_j \rangle, \tag{7}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C. \tag{8}$$

8.5 Derivation Steps

1. Write the primal problem with constraints.
2. Formulate the Lagrangian with multipliers.
3. Apply KKT conditions to relate primal to dual variables.
4. Substitute back and simplify to obtain the dual objective.
5. Identify dual constraints.

Question 9: Limitations of Kernel Methods

9.1 Scalability and Complexity

Kernels require storing the Gram matrix $K \in \mathbb{R}^{n \times n}$, costing $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ time (naïve). Prediction takes $\mathcal{O}(n_{SV} \cdot d)$ per instance.

9.2 Kernel Choice and Hyperparameter Tuning

Performance depends on selecting a valid PSD kernel (e.g. RBF) and tuning its parameters (e.g. bandwidth σ). Bad choices lead to under- or overfitting.

9.3 Lack of Feature Learning

Kernels fix a similarity measure and cannot learn hierarchical or task-specific representations as deep networks do.

9.4 Interpretability

The decision function

$$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b$$

is a weighted sum of kernel evaluations, making it difficult to interpret in the original feature space.

9.5 Memory Usage

Storing the full kernel matrix is often prohibitive; approximation methods (e.g. Nyström, random features) trade off accuracy.

9.6 Mercer's Condition

Ensuring a function K satisfies Mercer's condition

$$\iint f(x) K(x, y) f(y)\, dx\, dy \geq 0$$

for all f is non-trivial when designing custom kernels.

9.7 Sensitivity to Noise

Complex kernels can overfit noise or outliers unless regularization parameter C is carefully chosen.

Question 10: Relation between Perceptron and SVM

10.1 Perceptron as Unregularized Hinge Loss

The batch Perceptron minimizes the unregularized hinge loss:

$$L_{perc}(w) = \sum_{i=1}^{n} \max(0, -y_i(w^T x_i + b)).$$

Updates are applied whenever an example is misclassified:

$$w \leftarrow w + y_i x_i.$$

10.2 SVM as Regularized Hinge Loss

The soft-margin SVM solves:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)). \tag{1}$$

Its dual has identical form with box constraints $0 \leq \alpha_i \leq C$ and margin maximization.

10.3 Key Similarities and Differences

- Both have dual $w = \sum_i \alpha_i y_i x_i$ and can be kernelized.
- Perceptron has no regularizer and finds some separator if data are separable; SVM maximizes the margin.
- Perceptron converges only under separability, SVM always has a unique global optimum.
- SVM yields sparse solution (support vectors), Perceptron may accumulate many mistake vectors.
- Perceptron uses online updates, SVM solves a batch convex QP.

Question 11: Non-kernel-based Methods for Handling Nonlinear Separability

11.1 Explicit Feature Mappings

Design a nonlinear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^D$ so that data become linearly separable, e.g. polynomial, Fourier, or spline features.

11.2 Neural Networks / Deep Learning

Multi-layer perceptrons learn both feature representations and classifiers jointly via stacked nonlinear layers.

11.3 Tree-Based Models

Decision trees and ensemble methods (Random Forest, Gradient Boosting) partition the input space into regions with simple predictors.

11.4 Metric Learning

Learn a Mahalanobis distance $d_M(x, y) = \sqrt{(x-y)^T M (x-y)}$ to improve class separability under nearest-neighbor rules.

11.5 Manifold Learning and Embedding

Techniques like Isomap or UMAP embed high-dimensional data into low-dimensional manifolds where linear separators may suffice.

11.6 Hybrid / Deep Kernel Learning

Jointly learn a feature extractor $g_\theta(x)$ and a kernel function $\kappa(g_\theta(x), g_\theta(x'))$ for enhanced flexibility.

11.7 Ensemble of Linear Models

Boosting methods (e.g. AdaBoost) aggregate multiple linear classifiers:

$$F(x) = \mathrm{sign}\big(\sum_{t=1}^{T} \alpha_t w_t^T x + b_t\big).$$