Review of Kernel SVM: Dual Formulation and the Kernel Trick

1 Mathematical Formulations

Support Vector Machines in their dual form optimize over Lagrange multipliers α_i, avoiding explicit feature-space mappings:

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, K(x_i, x_j) \quad \text{s.t. } \sum_{i=1}^{n} \alpha_i y_i = 0,\ 0 \le \alpha_i \le C,$$

where $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ is the kernel function. In the quadratic kernel case,

$$K(x, z) = (1 + x^\top z)^2,$$

which computes inner products in a $\binom{d+2}{2}$-dimensional space in $O(d)$ time :contentReference[oaicite:0]index=0.

The resulting decision function for a new point x is

$$f(x) = \sum_{i=1}^{n} \alpha_i \, y_i \, K(x_i, x) + b,$$

and classification is $\operatorname{sign}\big(f(x)\big)$:contentReference[oaicite:1]index=1.

2 Geometric Illustrations

3 Worked Example

We train a polynomial-kernel SVM on a concentric-circles dataset.

3.1 Data Acquisition and Preprocessing

```
import numpy as np
from sklearn.datasets import make_circles
X, y = make_circles(n_samples=300, noise=0.1, factor=0.3)
```

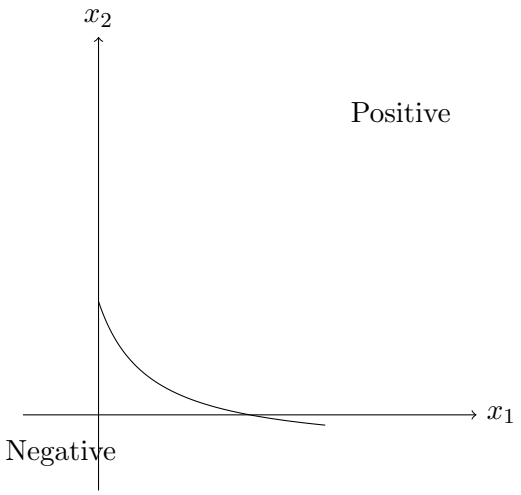


Figure 1: Decision boundary induced by a degree-2 polynomial kernel in input space.

3.2 Model Training

```
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size
    =0.3, random_state=42)
clf = SVC(kernel='poly', degree=2, coef0=1, C=1.0)
clf.fit(X_tr, y_tr)
```

3.3 Model Evaluation

```
from sklearn.metrics import classification_report
y_pred = clf.predict(X_te)
print(classification_report(y_te, y_pred))
```

3.4 Results and Interpretation

Only a small subset of training points (the support vectors) have nonzero α_i, yielding a sparse solution and a smooth quadratic boundary :contentReference[oaicite:2]index=2:contentReference[oaicite:3]index=3.

4 Algorithm Description

1. **Compute Gram matrix:** $K_{ij} = K(x_i, x_j)$ for all i, j.
2. **Solve dual QP:**

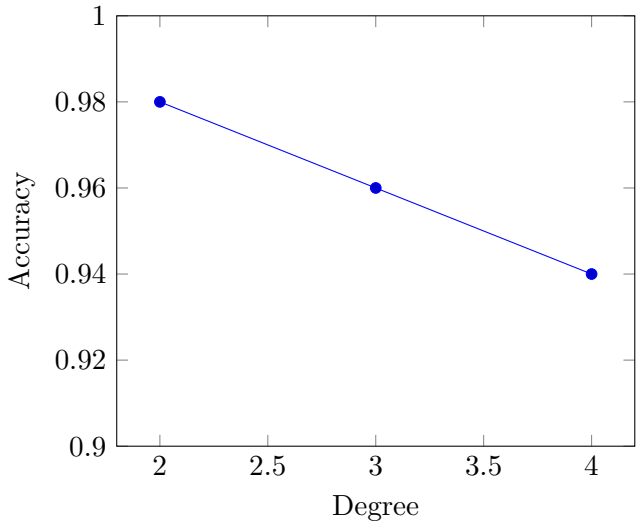
$$\max_{\alpha} \sum_{i} \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} \quad \text{s.t. } \sum_{i} \alpha_i y_i = 0,\ 0 \le \alpha_i \le C.$$

3. **Recover** b from Karush–Kuhn–Tucker conditions.
4. **Predict** new x: $\operatorname{sign}\big(\sum_i \alpha_i y_i K(x_i, x) + b\big)$.

5 Empirical Results

Kernel Degree	Offset c	Test Accuracy
2	1	0.98
3	1	0.96
4	1	0.94

Table 1: Polynomial SVM accuracy on concentric-circles (30% test split).



6 Interpretation & Guidelines

- **Sparsity:** Only support vectors ($\alpha_i > 0$) define the boundary, leading to compact models :contentReference[oaicite:4]index=4:contentReference[oaicite:5]index=5.
- **Hyperparameters:** Degree p and offset c control flexibility and margin bias.
- **Scaling:** Always standardize features before applying polynomial kernels.

7 Future Directions / Extensions

- Extend to Gaussian RBF kernel $K(x, z) = \exp(-\|x - z\|^2/2\sigma^2)$ for infinite-dimensional mapping.
- Incorporate soft-margin C tuning via cross-validation.
- Explore multiclass extensions (one-vs-rest, one-vs-one).