

支持向量回归（SVR）：原理、公式、应用与实战

2025 年 9 月 6 日

目录

1 引言

支持向量回归（SVR）将大间隔思想扩展到回归任务，通过 ε -不敏感损失与正则项控制模型复杂度。结合核函数，SVR 能在保持稀疏性的同时刻画非线性关系，对噪声具有一定鲁棒性。

2 原理与公式

2.1 模型与 ε -不敏感损失

给定样本 (\mathbf{x}_i, y_i) ，设 $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ 。SVR 通过

$$\min_{\mathbf{w}, b, \xi, \xi^*} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (1)$$

并满足 $|y_i - f(\mathbf{x}_i)| \leq \varepsilon + \xi_i$ 、 $|f(\mathbf{x}_i) - y_i| \leq \varepsilon + \xi_i^*$ 、 $\xi_i, \xi_i^* \geq 0$ 。其中 C 控制违例惩罚， ε 为“无惩罚”管宽。

2.2 对偶形式与核技巧

可得对偶问题：

$$\max_{\alpha, \alpha^*} -\frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^\top \mathbf{K} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \varepsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \quad (2)$$

约束 $\sum_i (\alpha_i - \alpha_i^*) = 0$ 、 $0 \leq \alpha_i, \alpha_i^* \leq C$ 。其中核矩阵 $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ 。预测为

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b, \quad (3)$$

非零 $\alpha_i - \alpha_i^*$ 对应支持向量。

2.3 超参数与预处理

- **标准化**：特征标准化（零均值、单位方差）通常必要；截距不做正则；居中有助稳定。
- **RBF 核**： $k(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$ 。关键超参： C （惩罚强度）、 ε （管宽）、 γ （核宽）。 C 越大拟合越“硬”； ε 越大越忽略小误差； γ 越大模型越“弯”。

3 应用场景与要点

- **非线性回归**：用 RBF 核捕捉平滑的非线性趋势；
- **鲁棒性**： ε -管降低对小噪声的敏感性；适当调小 C 提升鲁棒性；
- **模型选择**：用交叉验证在对数尺度搜索 C, ε, γ ；
- **可解释性**：支持向量反映对模型影响较大的样本，解具有稀疏性。

4 Python 实战

本脚本生成非线性合成数据，拟合 RBF-SVR，标注支持向量，并对超参数 C, ε, γ 的影响进行对比。图片保存到 figures/。

Listing 1: *gensvrfigures.py*

```

1 import os
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.svm import SVR
6
7 np.random.seed(7)
8
9 # 1) 非线性数据: y = sin(1.5x) + 0.5x + 噪声
10 n = 200
11 X = np.linspace(-3, 3, n).reshape(-1, 1)
12 y = np.sin(1.5*X[:, 0]) + 0.5*X[:, 0] + np.random.normal(0, 0.2, size=n)
13
14 # 标准化 X (常见做法); y 保持原尺度
15 scaler = StandardScaler().fit(X)
16 Xs = scaler.transform(X)
17

```

```
18 # 2) 训练 RBF-SVR
19 svr = SVR(kernel='rbf', C=10.0, epsilon=0.1, gamma='scale')
20 svr.fit(Xs, y)
21
22 # 在致密网格上预测
23 xx = np.linspace(X.min(), X.max(), 400).reshape(-1, 1)
24 xg = scaler.transform(xx)
25 yy = svr.predict(xg)
26
27 # 3) 绘制拟合与支持向量
28 fig, ax = plt.subplots(figsize=(7, 4.5))
29 ax.scatter(X[:, 0], y, s=15, alpha=0.6, label='data')
30 ax.plot(xx[:, 0], yy, color='crimson', lw=2.0, label='SVR (RBF)')
31 ax.scatter(X[svr.support_, 0], y[svr.support_], s=35, facecolors='none',
32           edgecolors='k',
33           label='support vectors')
34 ax.set_xlabel('x'); ax.set_ylabel('y'); ax.set_title('SVR (RBF) 拟合与
35           支持向量')
36 ax.legend(loc='best', fontsize=8)
37
38 fig_dir = os.path.join('0_Machine Learning', '0_Supervised Learning', '2
39           _SVR', 'figures')
40 os.makedirs(fig_dir, exist_ok=True)
41 plt.tight_layout(); plt.savefig(os.path.join(fig_dir, 'svr_rbf_fit.png')
42           ), dpi=160)
43
44 # 4) 超参数影响: 分别改变 C/epsilon/gamma
45 fig, axes = plt.subplots(1, 3, figsize=(12, 3.6), sharey=True)
46
47 # (a) 改变 C
48 for C in [0.3, 1.0, 10.0]:
49     m = SVR(kernel='rbf', C=C, epsilon=0.1, gamma='scale').fit(Xs, y)
50     axes[0].plot(xx[:, 0], m.predict(xg), label=f'C={C}')
51 axes[0].scatter(X[:, 0], y, s=8, alpha=0.3, color='gray')
52 axes[0].set_title('C 的影响'); axes[0].set_xlabel('x'); axes[0].
53     set_ylabel('y')
54 axes[0].legend(fontsize=8)
55
56 # (b) 改变 epsilon
57 for e in [0.05, 0.2, 0.5]:
58     m = SVR(kernel='rbf', C=10.0, epsilon=e, gamma='scale').fit(Xs, y)
59     axes[1].plot(xx[:, 0], m.predict(xg), label=f'eps={e}')
60 axes[1].scatter(X[:, 0], y, s=8, alpha=0.3, color='gray')
```

```

56 axes[1].set_title('epsilon 的影响'); axes[1].set_xlabel('x')
57 axes[1].legend(fontsize=8)
58
59 # (c) 改变 gamma (核宽)
60 for g in [0.3, 1.0, 3.0]:
61     m = SVR(kernel='rbf', C=10.0, epsilon=0.1, gamma=g).fit(Xs, y)
62     axes[2].plot(xx[:, 0], m.predict(xg), label=f'gamma={g}')
63 axes[2].scatter(X[:, 0], y, s=8, alpha=0.3, color='gray')
64 axes[2].set_title('gamma 的影响'); axes[2].set_xlabel('x')
65 axes[2].legend(fontsize=8)
66
67 plt.tight_layout(); plt.savefig(os.path.join(fig_dir, '
    svr_params_effect.png'), dpi=160)
68 print('saved to', os.path.join(fig_dir, 'svr_rbf_fit.png'), 'and
    svr_params_effect.png')

```

5 运行效果

图 ?? 与图 ?? 展示了 SVR 拟合与支持向量，以及 C, ϵ, γ 超参数变化对拟合曲线的影响。

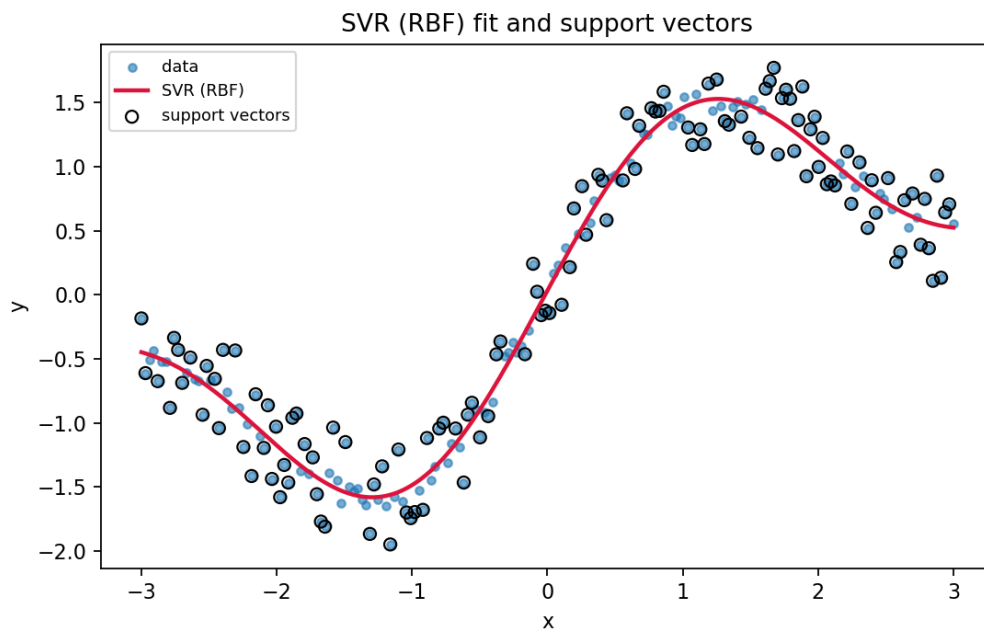


图 1: SVR (RBF) 拟合与支持向量（合成数据）

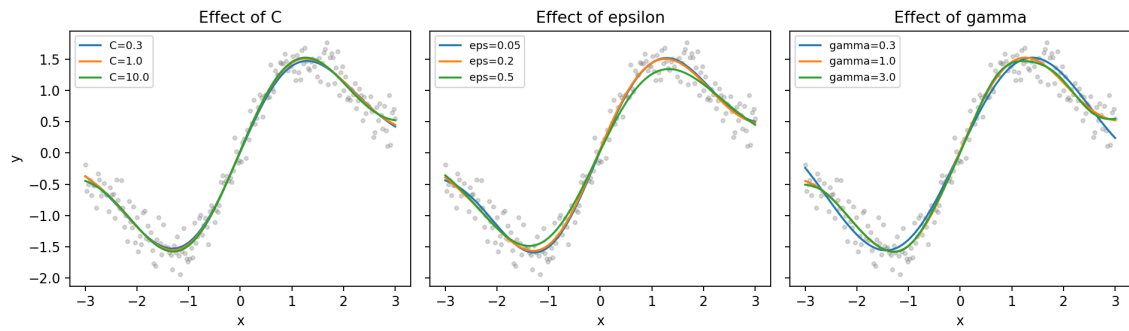


图 2: 超参数影响: 改变 C 、 ϵ 、 γ

6 小结

SVR 将大间隔与 ϵ -不敏感损失结合, 并借助核函数处理非线性回归。实践中应标准化特征, 并用交叉验证在对数尺度调参 C, ϵ, γ , 以获得稳健表现。