بسم الله الرحمن الرحیم

یادگیری ماشین

مدرس: محمدرضا محمدی

زمستان ۱۴۰۱

آموزش مدل‌ها

Training Models

Logistic Regression

- با این تابع می‌توان احتمال تعلق $\mathbf{x}$ به کلاس ۱ را محاسبه کرد

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x})$$

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

- توجه شود که $\sigma(t) < 0.5$ در $t < 0$ اتفاق می‌افتد
  - $\boldsymbol{\theta}^{\mathrm{T}}\mathrm{x} = 0$ معادل با احتمال 0.5 است و مرز تصمیم را مشخص می‌کند

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

- از تابع هزینه Cross Entropy استفاده می‌کنیم

$$c(\boldsymbol{\theta}) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1 - \hat{p}) & \text{if } y = 0 \end{cases}$$

Logistic Regression

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\Sigma_{i=1}^{m}\left[y^{(i)}log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)log\left(1 - \hat{p}^{(i)}\right)\right]$$

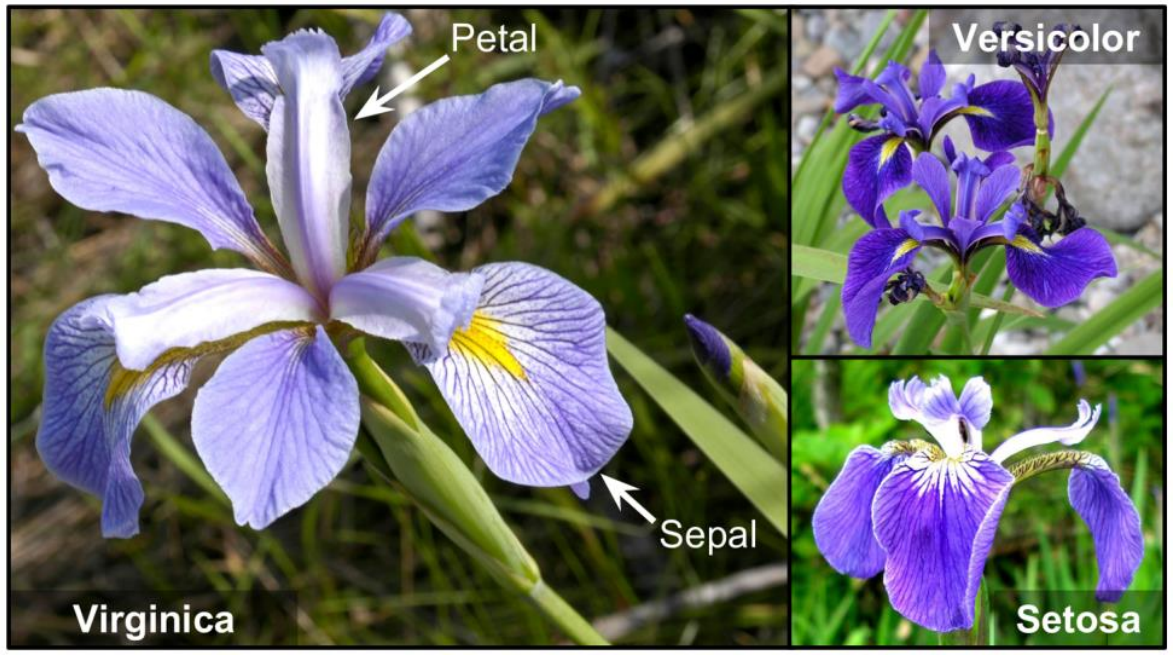
- پاسخ بسته‌ای برای یافتن مینیمم این تابع محاسبه نشده است
- اما این تابع محدب است، بنابراین با استفاده از GD می‌توان نقطه بهینه سراسری آن را بدست آورد

$$\frac{\partial}{\partial\theta_j}\mathrm{J}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma\left(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

- برای هر نمونه، خطای پیش‌بینی را محاسبه می‌کند و آن را در مقدار ویژگی jام ضرب می‌کند

مرزهای تصمیم

- برای شبیه‌سازی از مجموعه‌داده iris استفاده می‌کنیم
  - این مجموعه داده شامل ۴ ویژگی (عرض و ارتفاع کاسبرگ و گلبرگ) از ۱۵۰ گل زنبق از ۳ کلاس مختلف است




```
>>> from sklearn.datasets import load_iris
>>> iris = load_iris(as_frame=True)
>>> list(iris)
>>> iris.data.head(3)
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
>>> iris.target.head(3)  # note that the instances are not shuffled
0    0
1    0
2    0
```

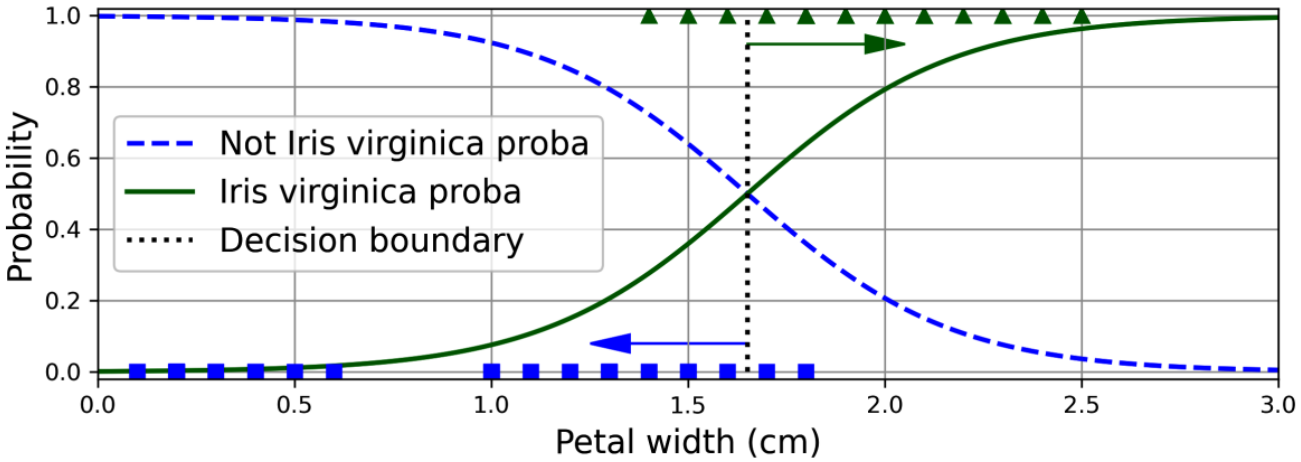
مرزهای تصمیم

- فرض کنید می‌خواهیم یک دسته‌بند برای تشخیص گل‌های virginica تنها با استفاده از عرض گلبرگ آموزش بدهیم

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

X = iris.data[["petal width (cm)"]].values
y = iris.target_names[iris.target] == 'virginica'
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

log_reg = LogisticRegression(random_state=42)
log_reg.fit(X_train, y_train)
```

```
X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
y_proba = log_reg.predict_proba(X_new)
decision_boundary = X_new[y_proba[:, 1] >= 0.5][0, 0]
```

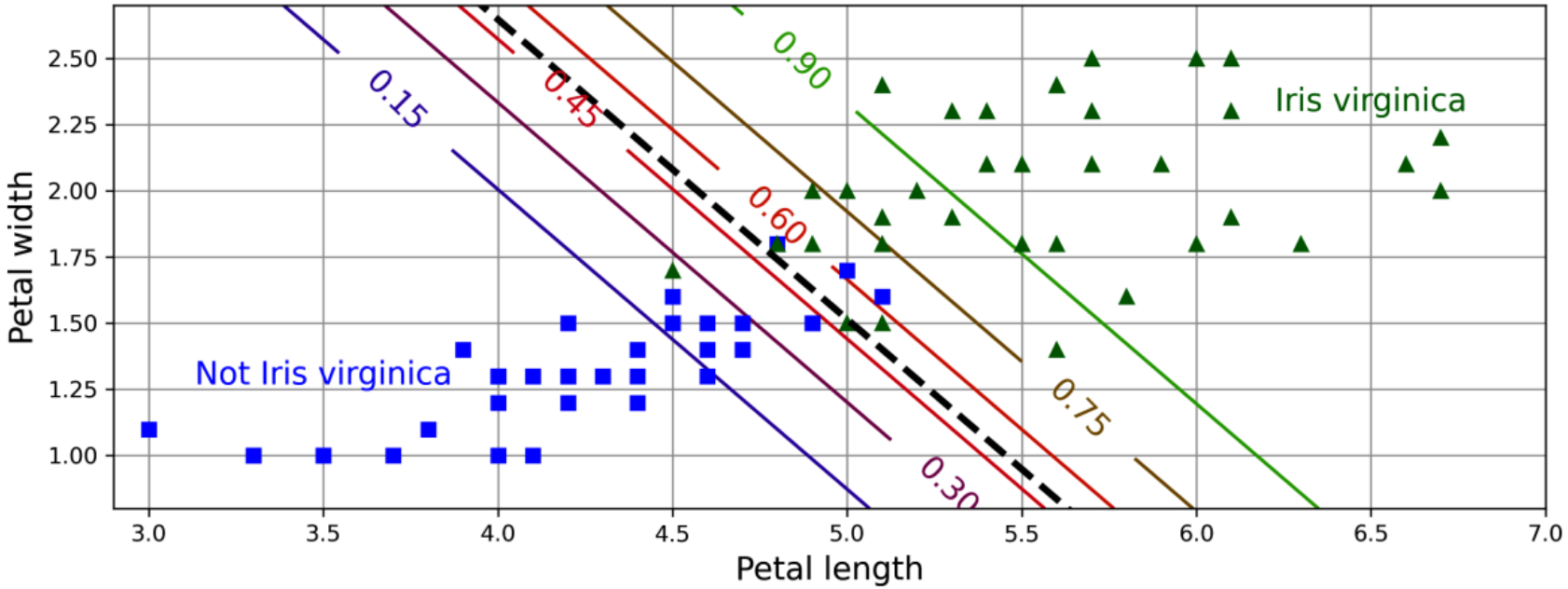
```
>>> decision_boundary
1.6516516516516517

>>> log_reg.predict([[1.7], [1.5]])
array([ True, False])
```



مرزهای تصمیم

- حال همان مسئله را با استفاده از دو ویژگی بررسی می‌کنیم
  - مرز تصمیم به چه صورتی است؟

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x})$$



Softmax Regression

- مدل logistic regression می‌تواند به سادگی برای دسته‌بندی چندکلاسه تعمیم داده شود
  - بدون نیاز به آموزش و ترکیب چندین دسته‌بند باینری
- برای هر کلاس، یک تابع امتیاز $s_k(\mathbf{x})$ آموزش داده می‌شود و سپس با نرمال‌سازی، یک احتمال برای هر کلاس تخمین زده می‌شود
  - هر کلاس بردار پارامترهای خاص $\boldsymbol{\theta}^{(k)}$ خودش را دارد
  - مجموع احتمال‌ها برابر با ۱ است

$$s_k(\mathbf{x}) = \left(\boldsymbol{\theta}^{(k)}\right)^{\mathsf{T}}\mathbf{x}$$

$$\widehat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K}\exp\left(s_j(\mathbf{x})\right)}$$

$$\widehat{y} = \underset{k}{\operatorname{argmax}}\ \sigma(\mathbf{s}(\mathbf{x}))_k = \underset{k}{\operatorname{argmax}}\ s_k(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\ \left(\left(\boldsymbol{\theta}^{(k)}\right)^{\mathsf{T}}\mathbf{x}\right)$$

Softmax Regression

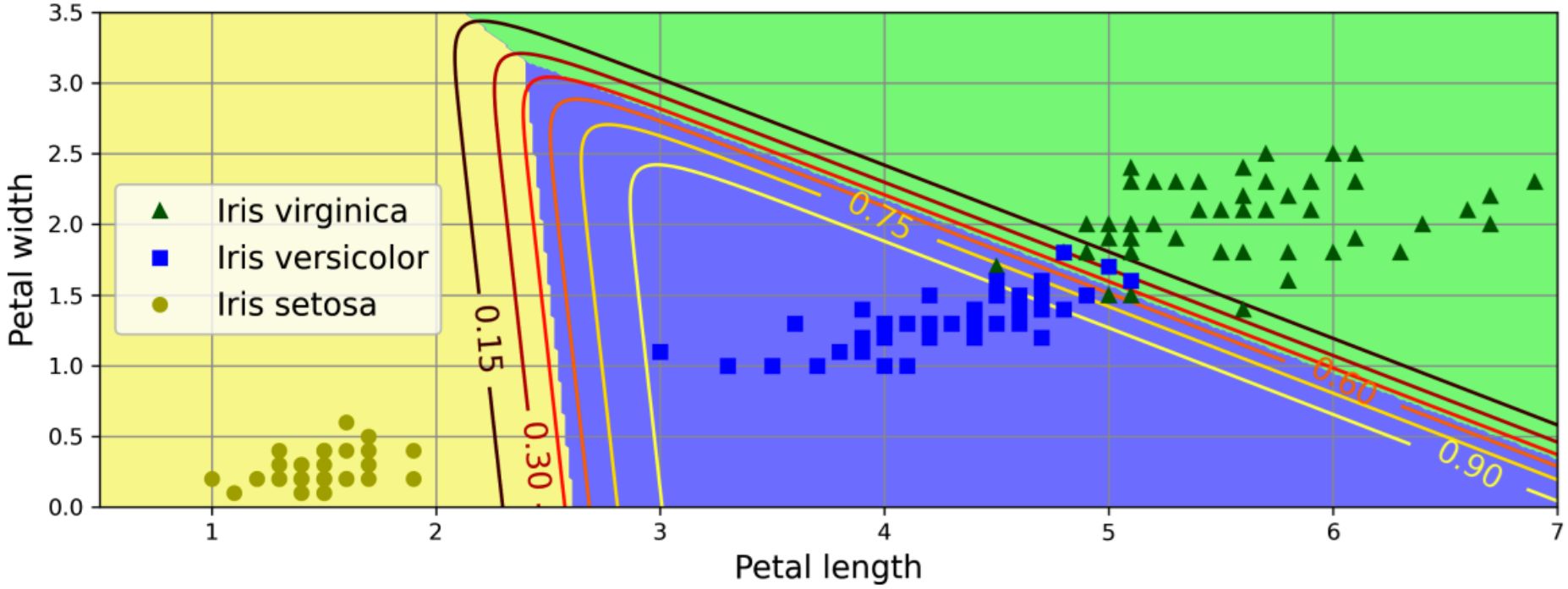
- تابع هزینه مناسب برای این مدل، تابع Cross Entropy است
  - برای حالت دوکلاسه دقیقا مشابه با BCE است

$$J(\mathbf{\Theta}) = -\frac{1}{m}\Sigma_{i=1}^{m}\Sigma_{k=1}^{K} y_k^{(i)}\log\left(\widehat{p}_k^{(i)}\right)$$

- بردار گرادیان برای وزن‌های کلاس k:

$$\nabla_{\boldsymbol{\theta}^{(k)}} J(\mathbf{\Theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(\widehat{p}_k^{(i)} - y_k^{(i)}\right)\mathbf{x}^{(i)}$$

```python
X = iris.data[["petal length (cm)", "petal width (cm)"]].values
y = iris["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

softmax_reg = LogisticRegression(C=30, random_state=42)
softmax_reg.fit(X_train, y_train)

>>> softmax_reg.predict([[5, 2]])
array([2])
>>> softmax_reg.predict_proba([[5, 2]]).round(2)
array([[0.  , 0.04, 0.96]])
```



مدل‌های خطی

- در این فصل، روش‌های مختلفی برای آموزش مدل‌های خطی (هم برای رگرسیون و هم برای دسته‌بندی) بررسی شد
- برای رگرسیون خطی پاسخ بسته محاسبه شد $\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \ \mathbf{X}^\top \ \mathbf{y} = \mathbf{X}^\dagger \mathbf{y}$
- استفاده از گرادیان کاهشی (و نسخه‌های stochastic آن) بررسی شد $\boldsymbol{\theta}^{(\text{next step})} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \text{MSE}(\boldsymbol{\theta})$
- با استفاده از طراحی ویژگی‌های غیرخطی، می‌توان به توابع غیرخطی با استفاده از مدل‌های خطی رسید
- افزودن عبارت‌های برای جریمه کردن مقادیر نامناسب برای وزن‌ها بررسی شد $J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \frac{\alpha}{m} \Sigma_{i=1}^{n} \theta_i^2$
- رگرسیون‌های Logistic و Softmax برای دسته‌بندی خطی معرفی شد $\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^\top \mathbf{x})$

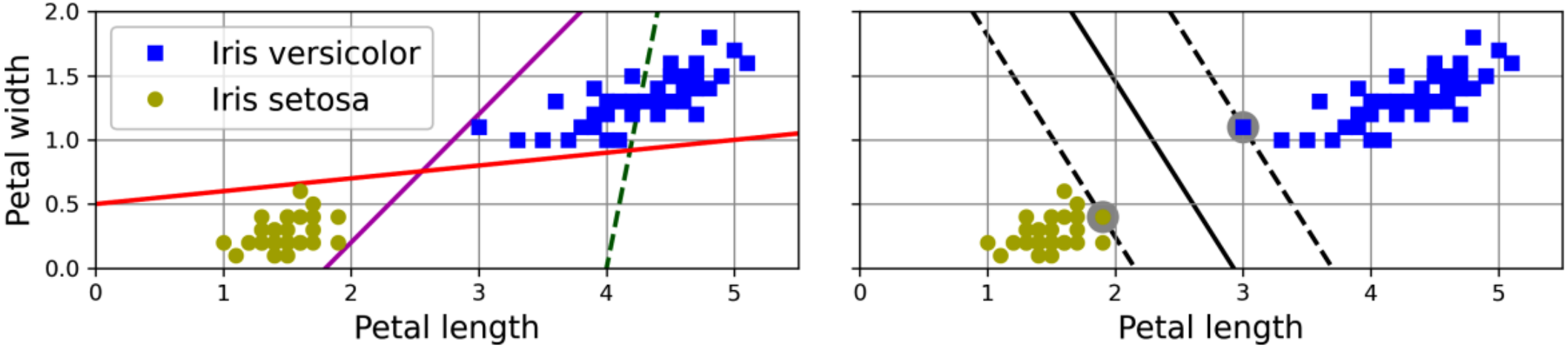
$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\Sigma_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)}$$

ماشین‌های بردار پشتیبان

Support Vector Machines

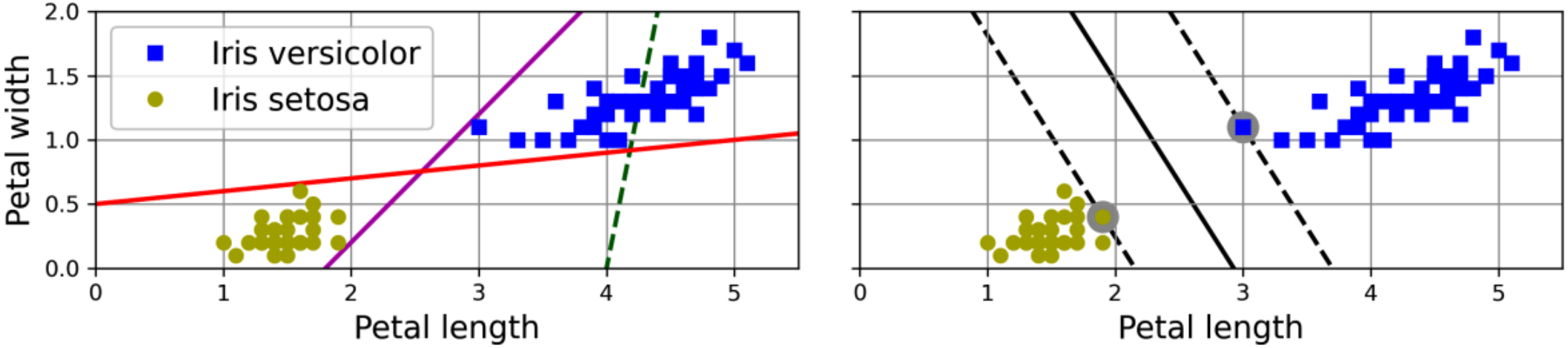
SVM خطی برای دسته‌بندی

- کدام مرز تصمیم مناسب‌تر است؟
  - داده‌های این مسئله به صورت خطی جداپذیر است (linearly separable) و هر دو خط قرمز و بنفش روی داده‌های آموزشی دقت کامل دارند
  - ایده اساسی SVM این است که مرز حاصل، حداکثر فاصله را با نزدیک‌ترین نمونه‌های آموزشی داشته باشد
    - دسته‌بندی با حاشیه بزرگ (large margin) نامیده می‌شود



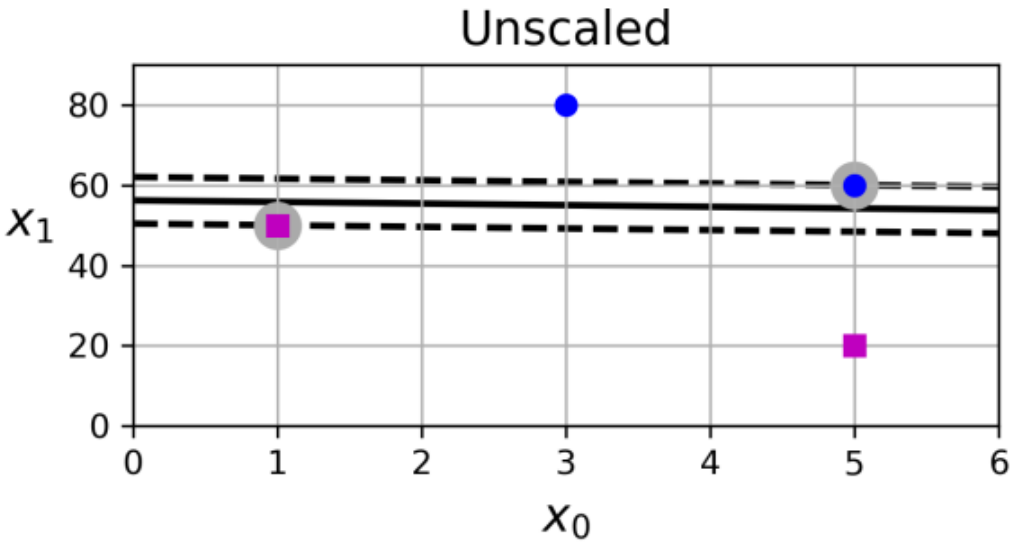
SVM خطی برای دسته‌بندی

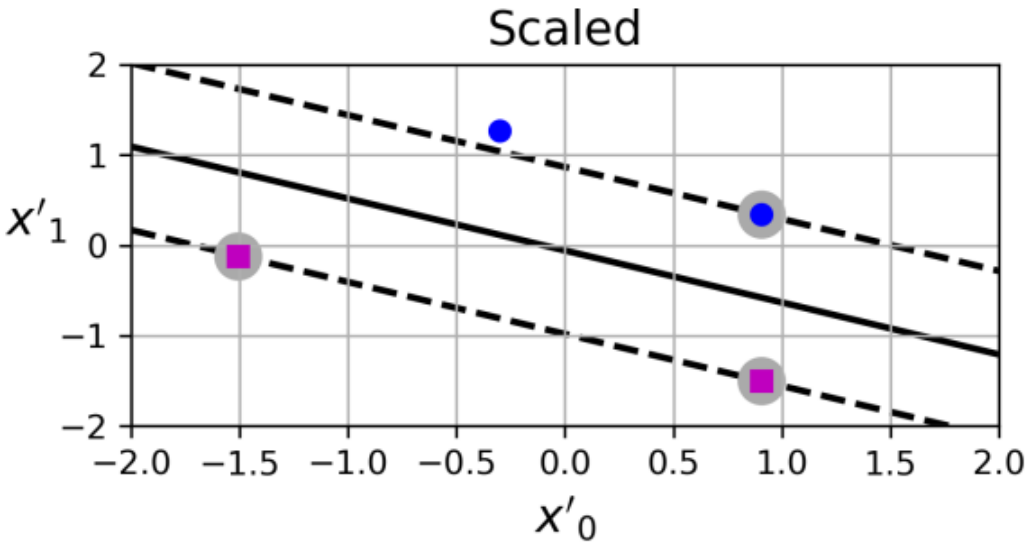
- به نمونه‌هایی که کمترین فاصله تا مرز را دارند، بردار پشتیبان (support vector) گفته می‌شود
  - افزودن نمونه‌هایی به مجموعه داده که خارج از این حاشیه باشند ("off the street")، تاثیری در مرز تصمیم ندارند



SVM خطی برای دسته‌بندی

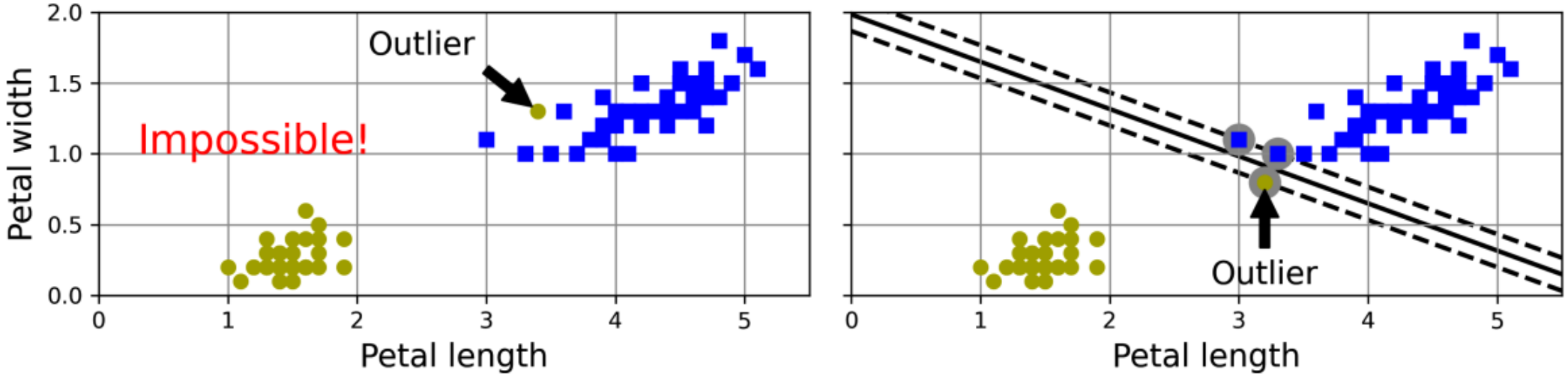
- آیا مقیاس ویژگی‌ها در تعیین بردارهای پشتیبان موثر است؟
  - بله! استانداردسازی ویژگی‌ها مهم است





حاشیه نرم (Soft Margin)

- در حالت حاشیه سخت (hard)، حساست نسبت به داده‌های پرت بسیار زیاد است و حتی برای مسائلی که به صورت خطی جداپذیر نیستند قابل استفاده نیست
- در حاشیه نرم، کمی منعطف‌تر رفتار می‌شود و سعی می‌شود حاشیه تا حد امکان زیاد باشد و نمونه‌هایی که از حاشیه عبور می‌کنند، جریمه ایجاد کنند

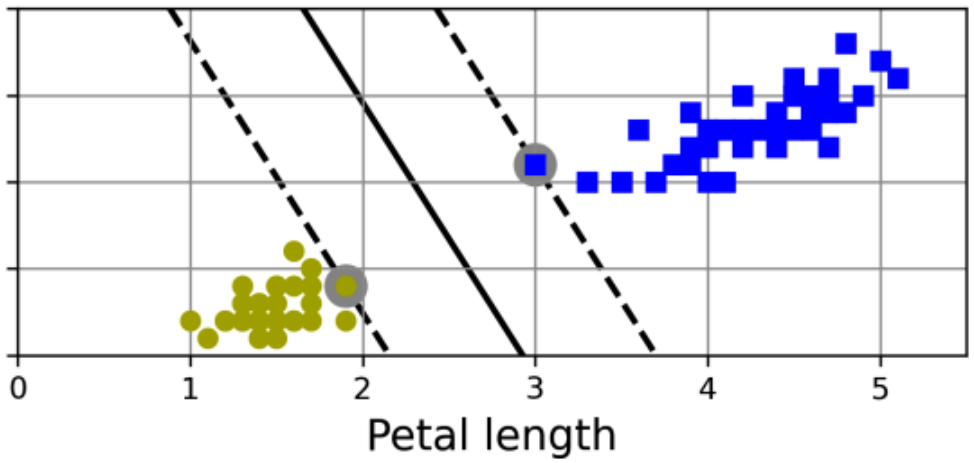


SVM خطی برای دسته‌بندی

$$\boldsymbol{\theta}^\top \mathbf{x} = \theta_0 x_0 + \cdots + \theta_n x_n = \mathbf{w}^\top \mathbf{x} + b = w_1 x_1 + \cdots + w_n x_n + b$$

$$\hat{y} = \begin{cases} 0 & \text{if } \boldsymbol{\theta}^\top \mathbf{x} < 0.5 \\ 1 & \text{if } \boldsymbol{\theta}^\top \mathbf{x} \geq 0.5 \end{cases}$$

- چگونه وزن‌های مربوط به خط دارای بیشترین حاشیه را بدست بیاوریم؟
  - حاشیه را جایی تعیین می‌کنیم که $\boldsymbol{\theta}^T \mathbf{x} = \pm 1$ شود
  - بیشترین حاشیه زمانی بدست می‌آید که $\mathbf{w}$ کمترین مقدار را داشته باشد
  - در حاشیه سخت، هیچکدام از نمونه‌ها نباید از حاشیه عبور کنند
  - مسئله بهینه‌سازی مقید زیر را باید حل کنیم:



$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

$$\text{subject to} \quad t^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) \geq 1 \quad \text{for } i = 1, 2, \cdots, m$$

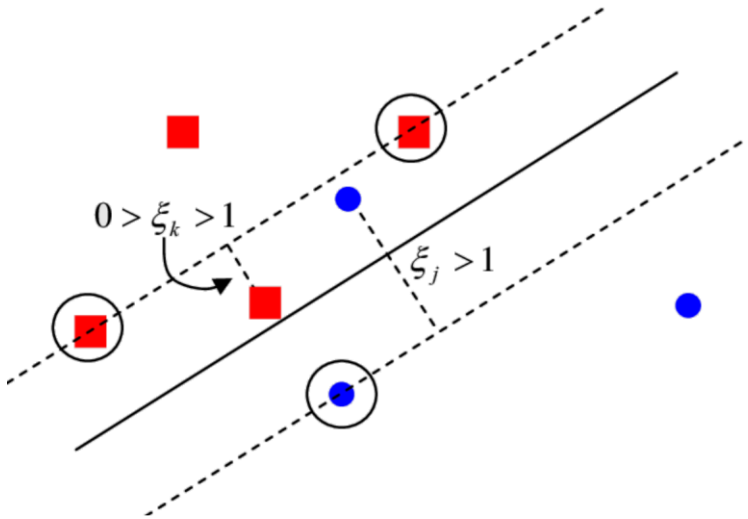
$$t^{(i)} = \begin{cases} -1 & \text{if } y^{(i)} = 0 \\ +1 & \text{if } y^{(i)} = 1 \end{cases}$$

حاشیه نرم

$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$

$$\text{subject to} \quad t^{(i)}\left(\mathbf{w}^\top\mathbf{x}^{(i)} + b\right) \geq 1 \quad \text{for } i = 1, 2, \cdots, m$$

- در حاشیه نرم، اجازه داده می‌شود تا برخی نمونه‌ها بتوانند حاشیه را بشکنند
  - متناسب با شکستن حاشیه، به تابع هزینه افزوده می‌شود

$$\underset{\mathbf{w}, b, \zeta}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{m}\zeta^{(i)}$$

$$\text{subject to} \quad t^{(i)}\left(\mathbf{w}^\top\mathbf{x}^{(i)} + b\right) \geq 1 - \zeta^{(i)} \quad \text{and} \quad \zeta^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$