Regularización

Modelos Lineales de Regularización

Introducción

Son estrategias diseñadas para reducir el error en el entrenamiento, que usualmente producidas por sobreajuste.

Ejemplo: reducir el grado de un polinomio.

Ridge Regression

- Es una versión regularizada de la regresión lineal.
- Término de regularización:

$$\alpha \sum_{i=1}^{n} \theta_i^2$$

- α modela qué tanto queremos regularizar el modelo. Ej: α=0.
- ¿Si α es grande? $\rightarrow$ Línea que pasa por la media de los datos.

Ridge Regression cost function:

$$\mathrm{J}(\theta) = \mathrm{MSR}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

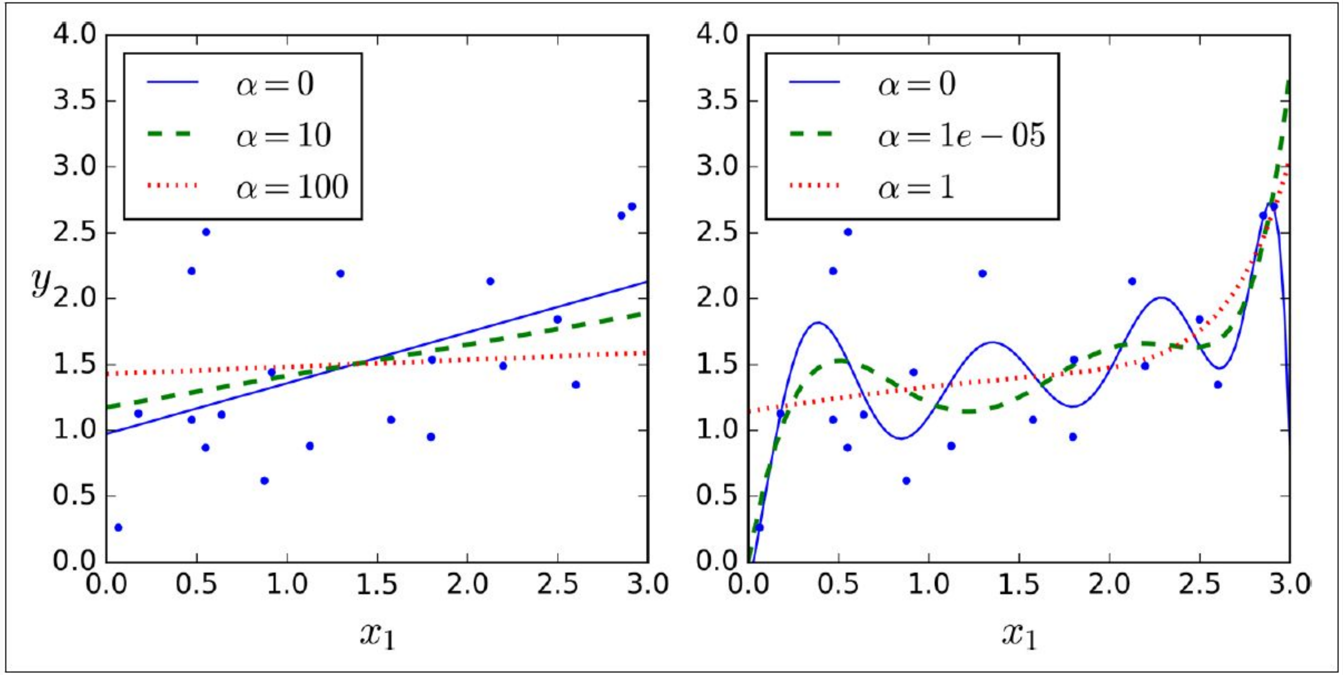
Obs: θ_0 no se regulariza.

- Sensible a la escala de las entradas, por lo tanto, es importante escalar los datos (StadarScaler)

Se emplea el modelo simple de Ridge Regression.

→ Se obtienen predicciones lineales.

1. Los datos son expandidos: PolinomyalFeatures(degree = 10)
2. Los datos se escalan: StandarScaled.
3. Se emplea el modelo Ridge Regression.
   → Se obtienen regresiones polinomiales.



$$\hat{\theta} = \left(X^T \cdot X + \alpha A\right)^{-1} \cdot X \cdot y$$

Ridge Regression closed-form solution

Donde A es la matriz identidad excepto por cero en la primer entrada

(the bias term).

Implementar Ridge Regression

- Con Scikit-Learn (empleando closed-form solution semejante a la ecuación anterior)

```
>>> from sklearn.linear_model import Ridge
>>> ridge_reg = Ridge(alpha=1, solver="cholesky")
>>> ridge_reg.fit(X, y)
>>> ridge_reg.predict([[1.5]])
array([[ 1.55071465]])
```

- Con Stochastic Gradient Descent

```
>>> sgd_reg = SGDRegressor(penalty="l2")
>>> sgd_reg.fit(X, y.ravel())
>>> sgd_reg.predict([[1.5]])
array([[ 1.13500145]])
```

Norma L2

Lasso Regression

Least Absolute Shrinkage and Selection Operator Regression

- Otra versión regularizada de la regresión lineal.
- Utiliza L1 en vez de L2
- Lasso Regression cost function:

$$J(\theta) = \mathrm{MSE}(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

Observaciones:

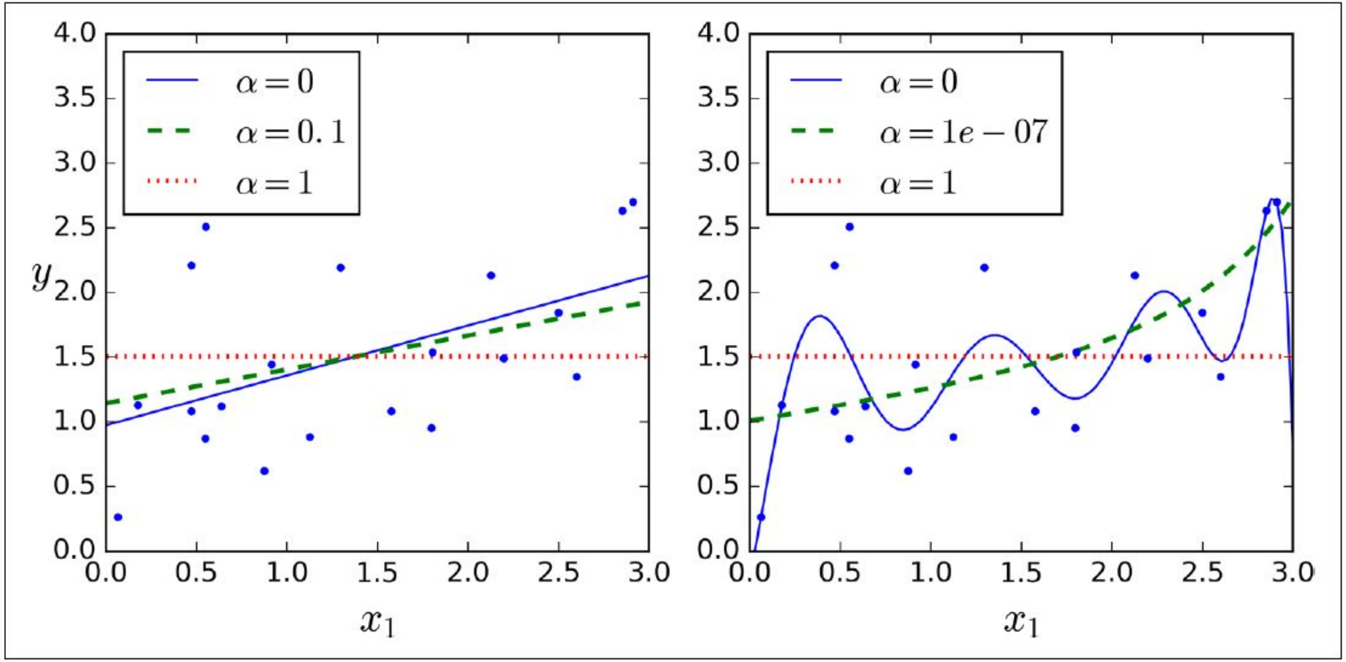
- Tiene a eliminar los pesos de las características menos importantes
- i.e. Automáticamente deja pocos pesos distintos a cero (sparse model)

Se emplea el modelo simple de Lasso Regression.

→ Se obtienen predicciones lineales.

1. Los datos son expandidos: PolinomyalFeatures(degree = 10)
2. Los datos se escalan: StandarScaled.
3. Se emplea el modelo Lasso Regression.

→ Se obtienen regresiones polinomiales.



Lasso cost function no es diferenciable en $\theta_i = 0$

En ese caso el Gradiente Descendente funciona bien si se emplea el vector subgradiente **g(·)**

$$g(\theta, J) = \nabla_\theta MSE(\theta) + \alpha \begin{pmatrix} \text{sign}(\theta_1) \\ \text{sign}(\theta_2) \\ \vdots \\ \text{sign}(\theta_1) \end{pmatrix} \quad \text{donde} \quad \theta_i = \begin{cases} -1 & \text{si } \theta_i < 0 \\ 0 & \text{si } \theta_i = 0 \\ 1 & \text{si } \theta_i > 0 \end{cases}$$

Lasso Regression subgradient vector

Implementar Lasso Regression

- Con Scikit-Learn

```
>>> from sklearn.linear_model import Lasso
>>> lasso_reg = Lasso(alpha=0.1)
>>> lasso_reg.fit(X, y)
>>> lasso_reg.predict([[1.5]])
```

- Con Stochastic Gradient Descent

```
>>> sgd_reg = SGDRegressor(penalty="l1")
>>> sgd_reg.fit(X, y.ravel())
>>> sgd_reg.predict([[1.5]])
```

Norma L1

Elastic Net

- Punto medio entre Ridge Regression y Lasso Regression.
- El término de regularización es una mezcla de ambas.
- Elastic Net cost function:

$$J(\theta) = \mathrm{MSE}(\theta) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \theta_i^2$$

¿Cuál de las tres usar?

- Por lo general es preferible tener un poco de regularización (evitar Linear Regression)
- Es bueno usar Ridge por default
- Si se sospecha que sólo pocas características son útiles, entonces usar Lasso o Elastic Net
- En general es preferible Elastic Net sobre Lasso
  - Lasso puede ser errática si hay correlación entre las características

Implementar Elastic Net

- Con Scikit-Learn

```
>>> from sklearn.linear_model import ElasticNet
>>> elastic_net = ElasticNet(alpha=0.1, l1_ratio=0.5)
>>> elastic_net.fit(X, y)
>>> elastic_net.predict([[1.5]])
```

l1_ratio corresponde a la tasa de mezcla r

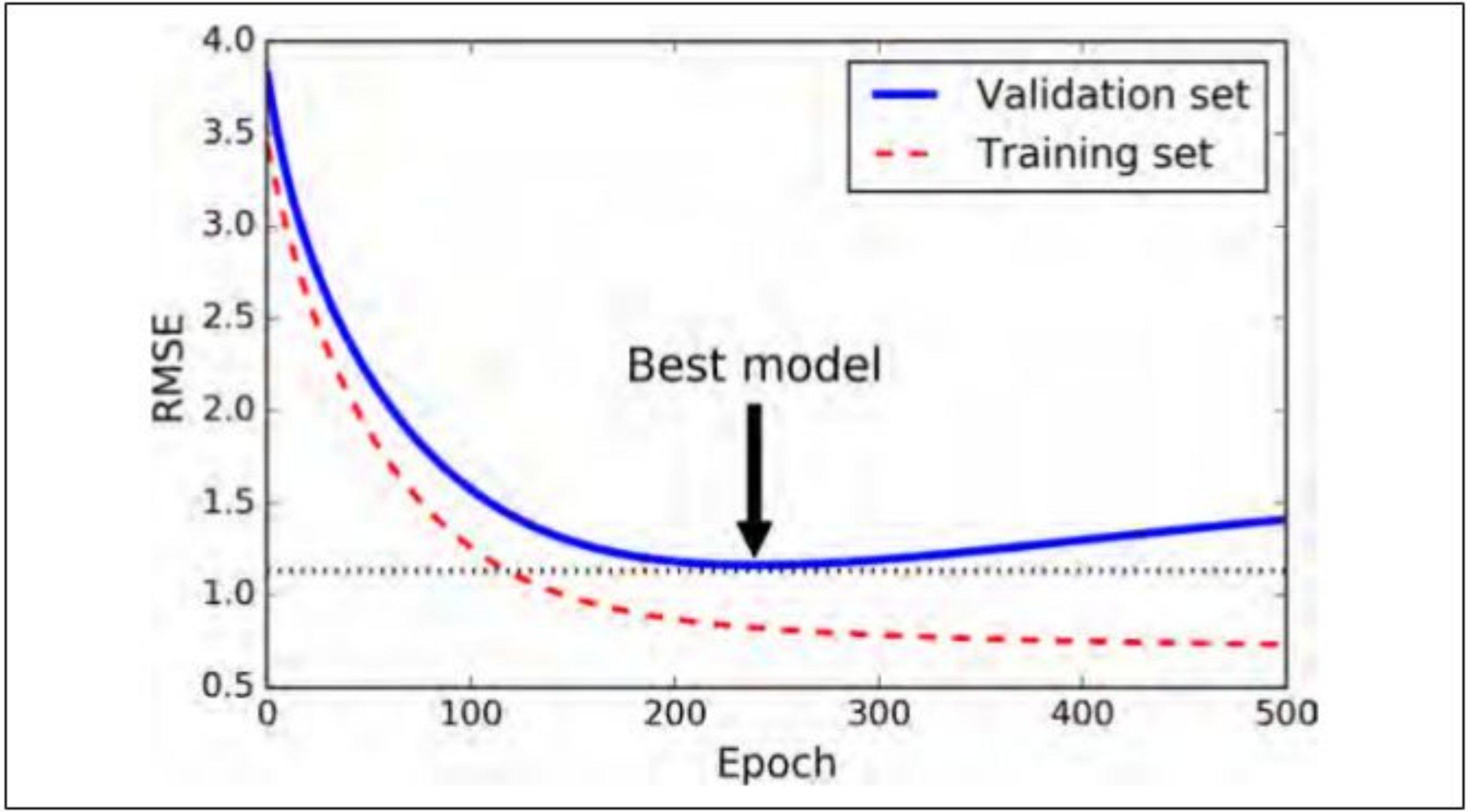
Early Stopping

- Una forma distinta de regularizar algoritmos de aprendizaje iterativos
- Detener el entrenamiento cuando el error de validación alcanza un mínimo.
  - En BGD el RMSE en el cjto de entrenamiento decrece
  - También decrece el RMSE del cjto de validación… por un tiempo
  - Tomar el mínimo evitando el sobre ajuste

Observaciones:

- Con Stochastic y mini-batch GD las curvas no son suaves y puede ser difícil se ha alcanzado o no el mínimo:

Tip: Parar cuando se ha estado por encima del mínimo durante un tiempo (estar seguros que el modelo ya no mejora). Luego retroceder en el algoritmo y quedarnos con el mínimo hasta ese punto.



Early stopping regularization

Logistic Regression

Logit Regression

- Estimar la prob. de que que una instancia pertenezca a determinada clase: ¿Este correo es spam?
- Si la prob. estimada es > 0.5
  → pertenece (etiqueta 1)
  si no
  → no pertenece (etiqueta 0)
- Por lo tanto es un Clasificador binario

Logistic Regresion Model (vectorized form):

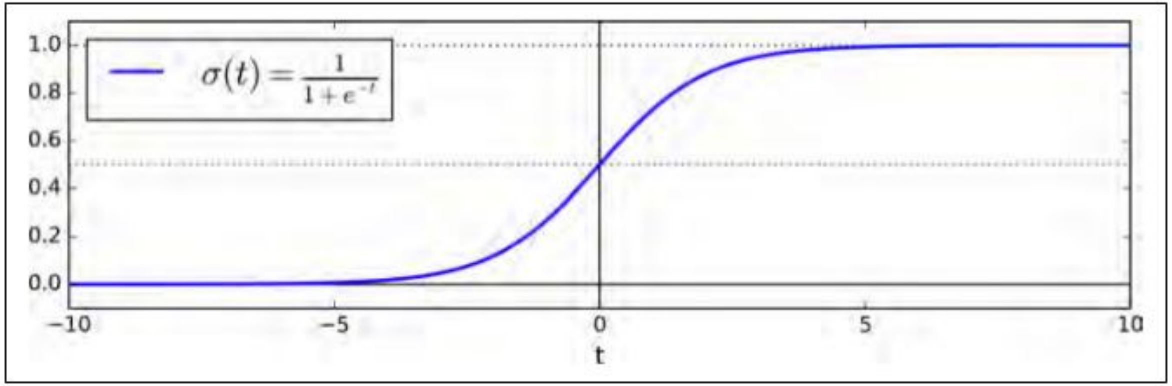
$$\hat{p} = h_\theta(\mathbf{x}) = \sigma(\theta^T \cdot \mathbf{x})$$

Logistic Function:

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

Logistic Regression model prediction

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



Logistic Function

Logistic Regression

Training & Cost Function

Función de costo para una sola instancia

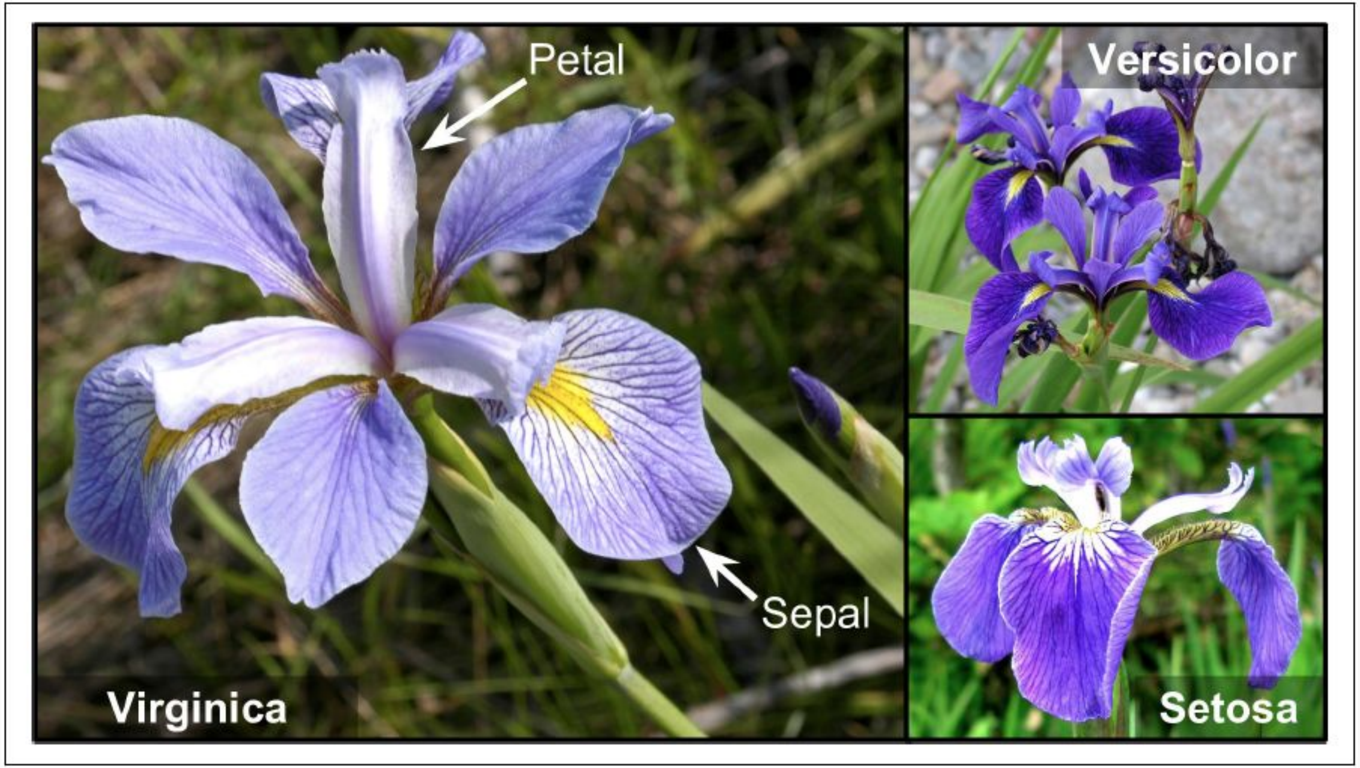
$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{si } y = 1 \\ -\log(1-\hat{p}) & \text{si } y = 0 \end{cases}$$

Función de costo sobre todo el conjunto de entrenamiento

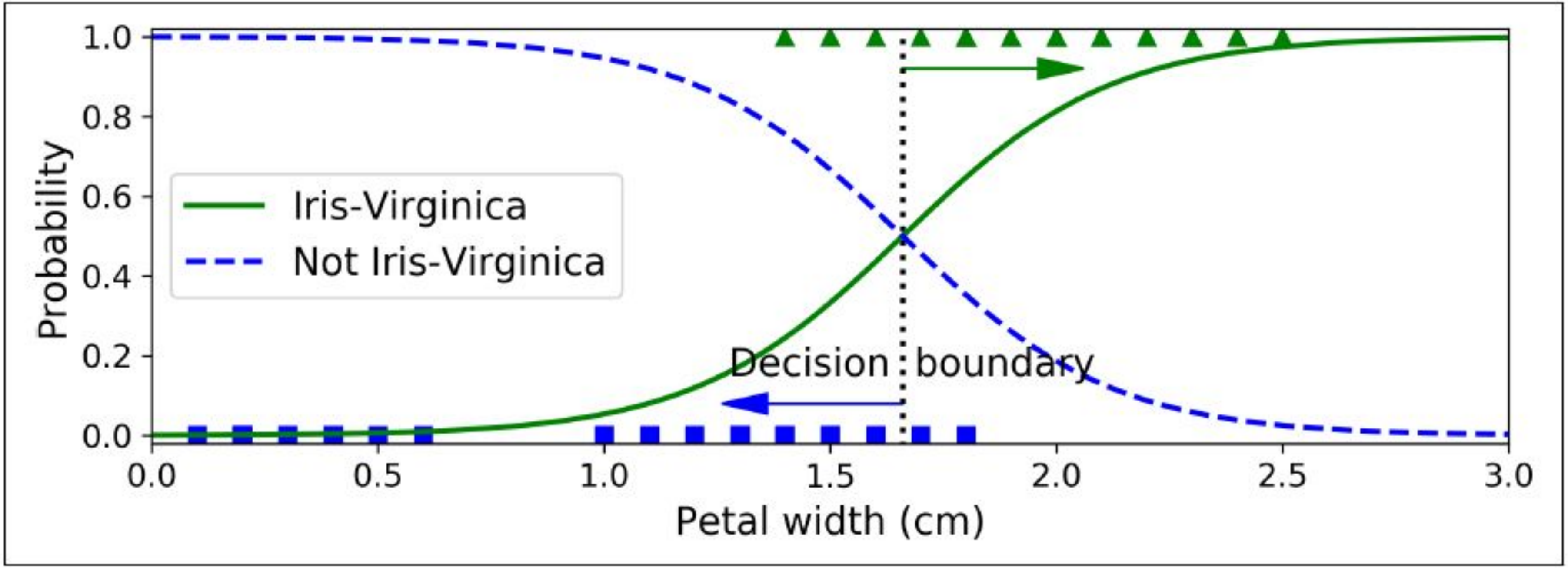
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(\hat{1}-p^{(i)}\right)\right]$$

Logistic Regression cost function (log loss)

- No conocemos una forma cerrada de la ec. para calcular θ el min.
- ¡Es convexa! → GD alcanza el mínimo



Flores de las 3 especies en iris



Probabilidades estimadas y fronteras de decisión