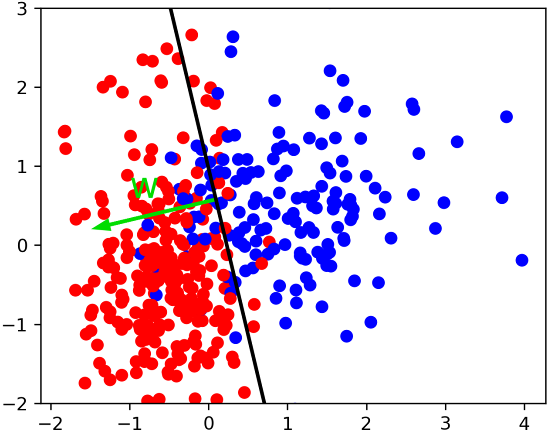
Logistic Regression

Presented by Yasin Ceran

Table of Contents

Linear Binary Classification

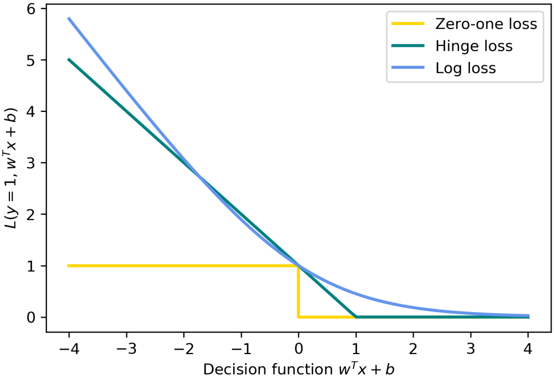


$$\hat{y} = \text{sign}(w^T \mathbf{x} + b) = \text{sign}\left(\sum_i w_i x_i + b\right)$$

Picking A Loss

$$\hat{y} = \text{sign}(w^T\mathbf{x} + b)$$

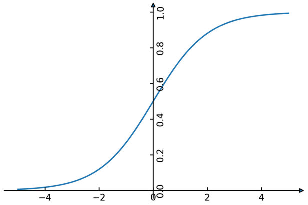
$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \sum_{i=1}^{n} 1_{y_i \neq \text{sign}(w^T\mathbf{x} + b)}$$



Logistic Regression

$$\log\left(\frac{p(y=1|x)}{p(y=-1|x)}\right) = w^T\mathbf{x} + b$$

$$p(y|\mathbf{x}) = \sigma(w^T x + b) = \frac{1}{1+e^{-w^T\mathbf{x}-b}}$$



$$\hat{y} = \mathrm{sign}(w^T\mathbf{x} + b)$$

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

$$C(\mathbf{w}) = \begin{cases} -log(\hat{p}), & \text{if } y = 1 \\ -log(1-\hat{p}), & \text{if } y = -1 \end{cases}$$

$$J(\mathbf{w}) = -\frac{1}{2n}\sum_{i=1}^{n}\left[(y^{(i)})\log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)})\right]$$

$$J(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}\log(\exp(-y_i(w^T\mathbf{x}_i + b)) + 1)$$

Penalized Logistic Regression

- $\min_{w\in^{p}, b\in\mathbb{R}} C \sum_{i=1}^{n} \log(\exp(-y_i(w^T\mathbf{x}_i + b)) + 1) + ||w||_2^2$
- $\min_{w\in^{p}, b\in\mathbb{R}} C \sum_{i=1}^{n} \log(\exp(-y_i(w^T\mathbf{x}_i + b)) + 1) + ||w||_1$
- C is inverse to alpha (or $alpha/n_samples$)
- Small C (a lot of regularization) limits the influence of individual points!

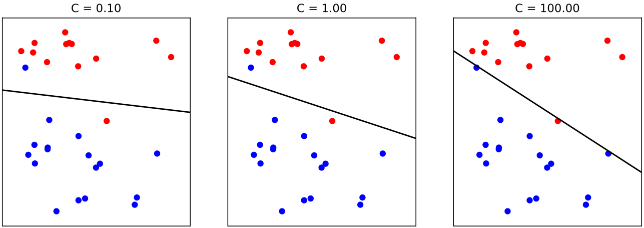


Table of Contents

MultiClass Classification

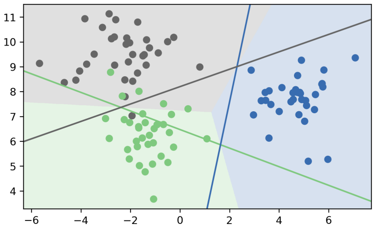
Reduction to Binary Clasification

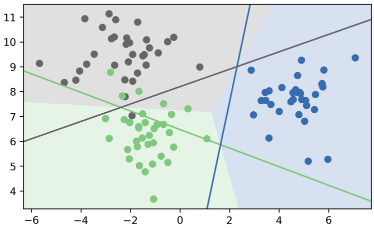
One Vs Rest

- For 4 classes:
  1v2,3,4, 2v1,3,4,
  3v1,2,4, 4v1,2,3
- In general:
  n binary classifiers
  each on all data

One vs One

- 1v2, 1v3, 1v4, 2v3, 2v4, 3v4
  n * (n-1) / 2 binary classifiers
  each on a fraction of the data
- "Vote for highest positives"
- Return most commonly predicted class.





In Scikit Learn

- OvO: only SVC
- OvR: default for all linear models except for logistic regression
- `LogisticRegression(multi_class=`auto')`
- `clf.decision_function` $= w^T x + b$
- `logreg.predict_proba`

MultiClass in Practice

OvR and multinomial LogReg produce one coef per class:

```
from sklearn.datasets import load_iris
iris = load_iris()
X, y = iris.data, iris.target
print(X.shape)
print(np.bincount(y))
```

```
(150, 4)
[50 50 50]
```

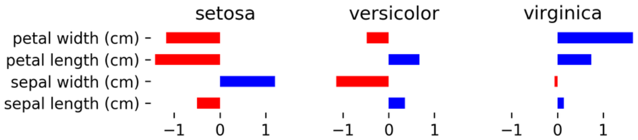
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC

logreg = LogisticRegression(multi_class="multinomial", solver="lbfgs").fit(X, y)
linearsvm = LinearSVC().fit(X, y)
print(logreg.coef_.shape)
print(linearsvm.coef_.shape)
```

```
(3, 4)
(3, 4)
```

MultiClass in Practice

```
logreg.coef_
```

```
array([[-0.42339232,  0.96169329, -2.51946669, -1.0860205 ],
       [ 0.53411332, -0.31794321, -0.20537377, -0.93961515],
       [-0.11072101, -0.64375008,  2.72484045,  2.02563566]])
```



(after centering data, without intercept)

Summary

- Logistic Regression and Linear SVM differ from each other by their loss functions