Data Mining & Machine Learning

CS37300
Purdue University

Sep 20, 2023

Today's topics

- Support Vector Machine (SVM)
- Non-separable data
- Soft-margin SVM
- Noise Modeling, Logistic Regression

Linear Classifiers

- We have a **training data set** **S** = $\{(x_1,y_1),\ldots,(x_n,y_n)\}$ of pairs (x,y)
- x: feature vector $x \in \mathbb{R}^d$
- y: **binary** labels: $y \in \{-1,1\}$
- A simple representation for classifiers: **linear classifier**
- Learn parameters $\widehat{w} \in \mathbb{R}^d$ and $\widehat{b} \in \mathbb{R}$
- After training, classify any new point as

$$\hat{h}(x) = \mathrm{sign}\left(\hat{w}^\top x + \hat{b}\right) \in \{-1, 1\}$$

Finding the SVM Solution

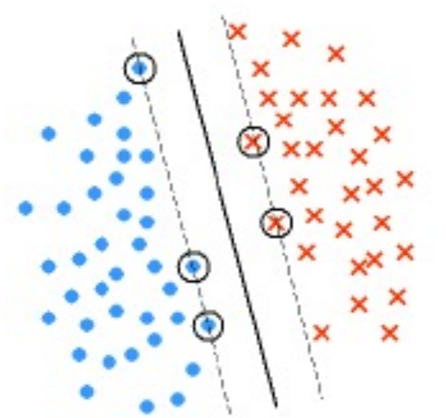
$$(\hat{w}, \hat{b}) = \operatorname*{argmax}_{(w,b):\|w\|=1} \min_{1 \le i \le n} y_i \left(w^\top x_i + b\right)$$

- Can express SVM as a **quadratic program**:

$$\boxed{\begin{array}{l} \text{Minimize} \quad \|w\|^2 \\ \text{subject to} \;\; y_i \left(w^\top x_i + b\right) \ge 1, \;\; \forall i : \;\; 1 \le i \le n \end{array}}$$

- Why is this important? There are standard software packages to solve optimization problems expressed in this form.

- The name "*Support Vector Machine*" stems from the fact that supported by (i.e., is the linear span of) the examples that are at a distance 1 / ||w*|| from the separating hyperplane. These \ are therefore called support vectors.

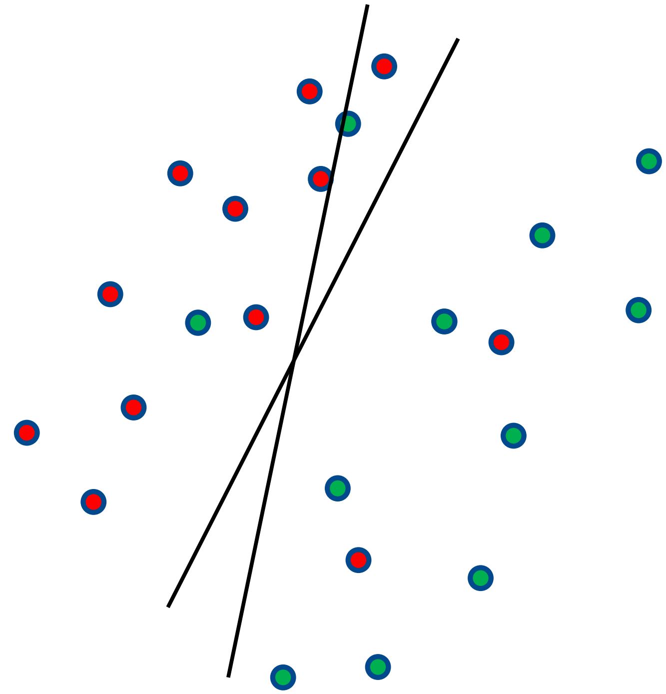


Non-linearly-separable Data

- What if the data aren't linearly separable?

- We want to find

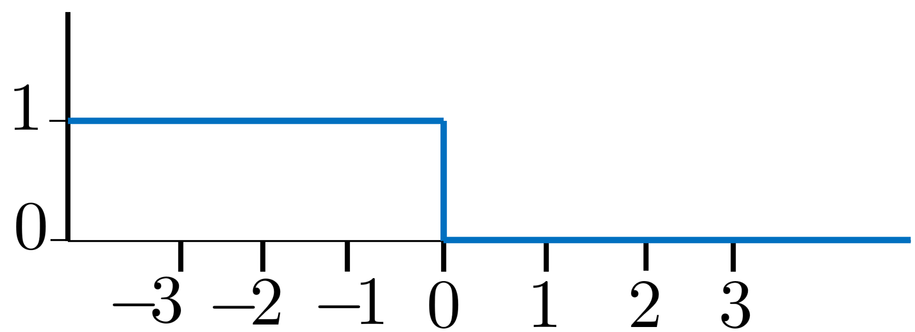
$$(\hat{w}, \hat{b}) = \operatorname*{argmin}_{w,b} \sum_{i=1}^{n} \mathbb{I}[y_i(w^\top x_i + b) \leq 0]$$



- This is generally NP-Hard to minimize (computationally intractable)

- The function is **non-convex**

$$\text{the “0-1” loss}: \quad \ell(z) = \mathbb{I}[z \leq 0]$$

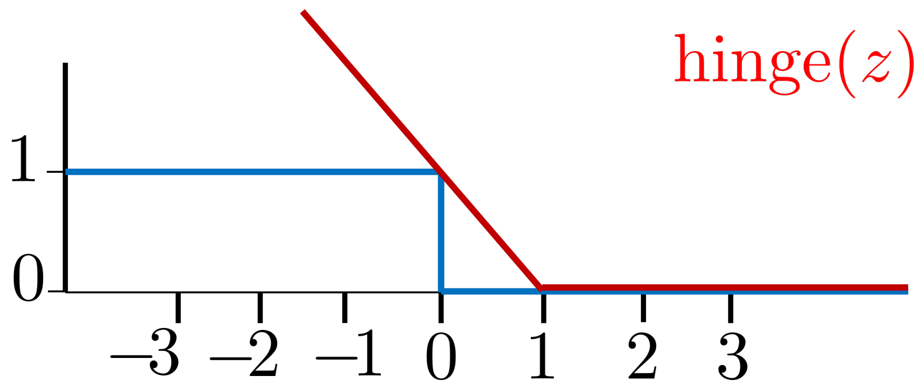


Convex Surrogate Loss

- Relax the objective to make it convex

the "0-1" loss :

$\ell(z) = \mathbb{I}[z \leq 0]$



the hinge loss :

$\text{hinge}(z) = \max\{1 - z, 0\}$

Soft-margin SVM: $$\text{Minimize}\,(1/2)\|w\|^2 + C\sum_{i=1}^{n} \max\{1 - y_i\left(w^\top x_i + b\right), 0\}$$

Equivalently:

(quadratic program)

$$\boxed{\begin{aligned} &\text{Minimize} && (1/2)\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to} && y_i\left(w^\top x_i + b\right) \geq 1 - \xi_i,\ \forall i:\ 1 \leq i \leq n \\ &&& \xi_i \geq 0,\ \forall i : 1 \leq i \leq n \end{aligned}}$$

Later, we'll discuss the "dual" form, when we cover kernel methods

Soft-margin SVM

$$\begin{array}{ll} \text{Minimize} & (1/2)\|w\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} & y_i \left(w^\top x_i + b \right) \geq 1 - \xi_i, \;\; \forall i : \;\; 1 \leq i \leq n \\ & \xi_i \geq 0, \;\; \forall i : 1 \leq i \leq n \end{array}$$

- The C is hyperparameter for us to set
- It defines a trade-off between the loss on the data and the norm ||w||
- For separable data, taking C → ∞ equivalent to “hard margin” SVM from earlier
- The term $||w||^2$ is called a **regularizer**
- Other losses and other regularizers could be used instead, corresponding to other learning algorithms.

Another Approach: **Modeling** the Noise

Logistic Regression

Modeling the Noise: Conditional Probability

- Another approach is to model non-separability as the result of noisy labels
- Examples:
  - X = person’s health history, $Y = 1$ if they will develop heart disease.
    Given a person’s health history x,
    whether they will develop heart disease is not deterministic.
    It has some conditional probability $P(Y=1|X=x)$.
  - X = weather data, $Y = 1$ if it will snow the next day.
    Given the current weather data x,
    whether it will snow tomorrow has some conditional probability $P(Y=1|X=x)$

Modeling the Noise: Conditional Probability

- For learning a linear classifier w:
- Intuitively, we want a model where $P_w(Y=1|X=x)$ is larger for larger $w^\top x + b$
  - Which of the following models makes the most sense? Why?

- Idea 1: make $P_w(Y = 1|X = x) = w^\top x + b$
- Idea 2: make $P_w(Y = 1|X = x) = e^{w^\top x+b}$
- Idea 3: make $P_w(Y = 1|X = x) = \dfrac{1}{1 + e^{-(w^\top x+b)}}$

Modeling the Noise: Conditional Probability

- For learning a linear classifier w:
- Intuitively, we want a model where P(Y=1|X=x) is larger for larger $w^\top x + b$
- Idea 1: make $P_w(Y=1|X=x) = w^\top x + b$
- But $w^\top x$ is unbounded, and can be negative. (needed between 0 and 1)
- Idea 2: make $\log(P_w(Y=1|X=x)) = w^\top x + b$ $\left(\text{same as } P_w(Y=1|X=x) = e^{w^\top x + b}\right)$
- But $P_w(Y=1|X=x) \leq 1$, so $\log(P_w(Y=1|X=x))$ is unbounded only on the negative side
- Idea 3 (logistic transform): make

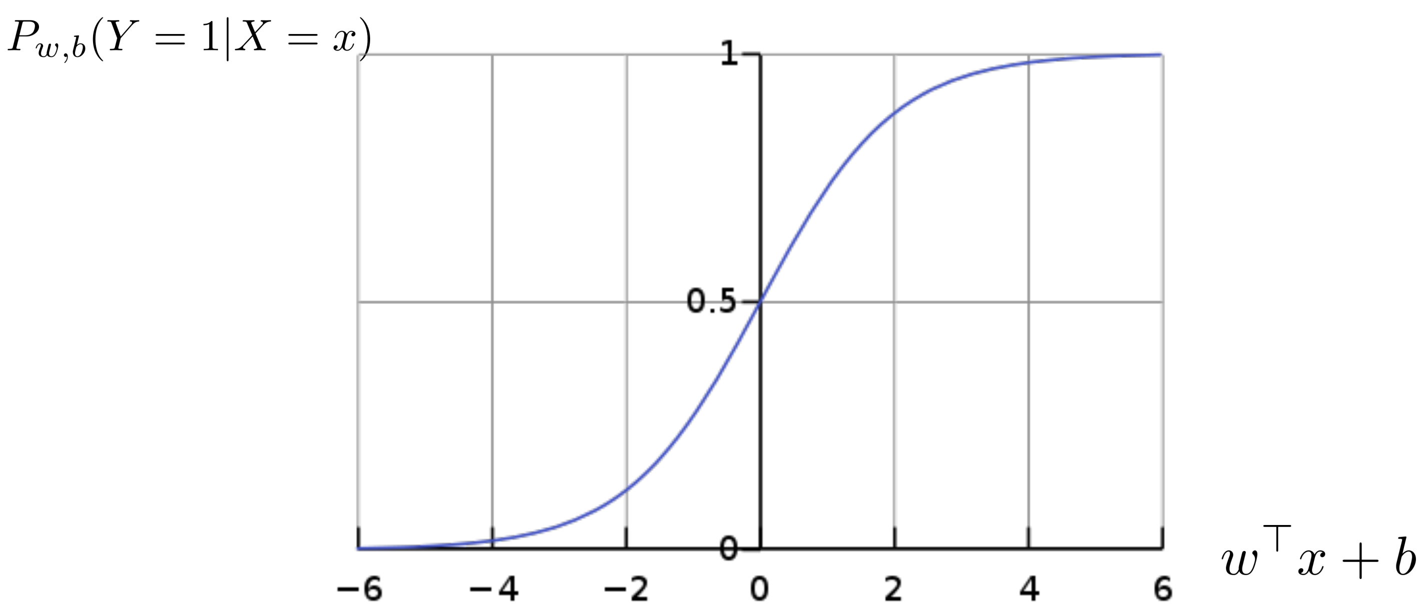
$$\log\left(\frac{P_w(Y=1|X=x)}{P_w(Y=-1|X=x)}\right) = w^\top x + b$$

- Solving this for P(Y=1|X=x): $$P_w(Y=1|X=x) = \frac{1}{1+e^{-(w^\top x + b)}}$$

Modeling the Noise: Conditional Probability

- Logistic Model:

$$P_{w,b}(Y = 1|X = x) = \frac{1}{1 + e^{-(w^\top x + b)}}$$



- Nice properties:
- Close to 1 for very large $w^\top x + b$
- Close to 0 for very small $w^\top x + b$
- Equal 1/2 when $w^\top x + b = 0$ (i.e, for x on the linear classifier's boundary)

Modeling the Noise: Conditional Probability

- Logistic Model:

$$P_{w,b}(Y = 1|X = x) = \frac{1}{1 + e^{-(w^\top x + b)}}$$

- Note:

$$\begin{aligned} P_{w,b}(Y = -1|X = x) &= 1 - \frac{1}{1 + e^{-(w^\top x + b)}} \\ &= \frac{1 + e^{-w^\top x}}{1 + e^{-(w^\top x + b)}} - \frac{1}{1 + e^{-(w^\top x + b)}} \\ &= \frac{e^{-(w^\top x + b)}}{1 + e^{-(w^\top x + b)}} \\ &= \frac{1}{1 + e^{(w^\top x + b)}} \end{aligned}$$

- So generally:

$$P_{w,b}(Y = y|X = x) = \frac{1}{1 + e^{-y(w^\top x + b)}}$$

Training: Logistic Regression

- How do we find a good $\hat{w}$?
- **Maximum conditional likelihood** estimation
- For a data set S = {(x_1,y_1),…,(x_n,y_n)}
- Define the conditional likelihood

Models the labels y_i as being conditionally independent given x_i

$$L_{Y|X}(w, b; S) = \prod_{i=1}^{n} P_{w,b}(Y = y_i | X = x_i)$$

- The maximum conditional likelihood estimator is

$$(\hat{w}, \hat{b}) = \underset{w,b}{\operatorname{argmax}}\, L_{Y|X}(w, b; S)$$

- Equivalently (because it's easier to work with), conditional **log-likelihood**:

$$l_{Y|X}(w, b; S) = \ln(L_{Y|X}(w, b; S))$$

- and then

$$(\hat{w}, \hat{b}) = \underset{w,b}{\operatorname{argmax}}\, l_{Y|X}(w, b; S)$$

Training: Logistic Regression

- Explicitly:

$$
\begin{aligned}
l_{Y|X}(w,b;S) &= \ln\left(\prod_{i=1}^{n} P_w(Y=y_i|X=x_i)\right)\\
&= \sum_{i=1}^{n} \ln(P_w(Y=y_i|X=x_i))\\
&= \sum_{i=1}^{n} \ln\left(\frac{1}{1+e^{-y_i(w^\top x_i+b)}}\right)\\
&= -\sum_{i=1}^{n} \ln\left(1+e^{-y_i(w^\top x_i+b)}\right)
\end{aligned}
$$

- Unfortunately, there is no simple expression for the $\widehat{w}$ that maximizes this
- But $-\sum_{i=1}^{n} \ln\left(1+e^{-y_i(w^\top x_i+b)}\right)$ is a concave function, which can be maximized using iterative numerical methods: e.g., gradient ascent or Newton's method.

Training: Logistic Regression

- Explicitly: denote $\sigma(x) = \dfrac{1}{1+e^{-x}}$

- Simple derivative: $\dfrac{\partial \sigma(x)}{\partial x} = \sigma(x)(1-\sigma(x))$

$$\nabla l_{Y|X}(w,b;S) = \nabla \sum_{i=1}^{n} \ln\big(\sigma\big(y_i(w^\top x_i + b)\big)\big)$$

$$= \sum_{i=1}^{n} \frac{1}{\sigma(y_i(w^\top x_i + b))} \sigma\big(y_i(w^\top x_i + b)\big)\left(1 - \sigma\big(y_i(w^\top x_i + b)\big)\right) y_i[x_i, 1]^\top$$

$$= \sum_{i=1}^{n} \left(1 - \sigma\big(y_i(w^\top x_i + b)\big)\right) y_i[x_i, 1]^\top$$

$$\text{gradient ascent: iterate} \quad (w,b) \leftarrow (w,b) + \epsilon \nabla l_{Y|X}(w,b;S)$$

Logistic Regression vs SVM

- For a given data set, which one should we use?
- If you think the data are (nearly) linearly separable, and with large margin, makes sense to use (Soft)-SVM
- If you think the cause of non-separability truly is label noise, makes sense to use logistic regression
- One nice thing about Logistic Regression is that it provides an estimated probability $P(y|x)$ for its predictions, rather than just a -1,1 prediction.
- Doesn't hurt to try them both and find out which is better on a validation set!