COGS 118A, Winter 2020

Supervised Machine Learning Algorithms

Lecture 09: Logistic Regression and Support Vector Machine

Zhuowen Tu

Logistic regression classifier continued..

Linear Model

A linear model:

$$
\begin{aligned}
f(\mathbf{x}; \mathbf{w}, b) &= < \mathbf{w}, \mathbf{x} > + b \\
&= \mathbf{w} \cdot \mathbf{x} + b \\
&= \mathbf{w}^T \mathbf{x} + b
\end{aligned}
$$

$$\mathbf{x} = \mathbb{R}^m \qquad \mathbf{w} = \mathbb{R}^m \qquad b \in \mathbb{R}$$

This is a linear function and our job is find the optimal **w** and b to best fit the prediction in learning.

Main motivation

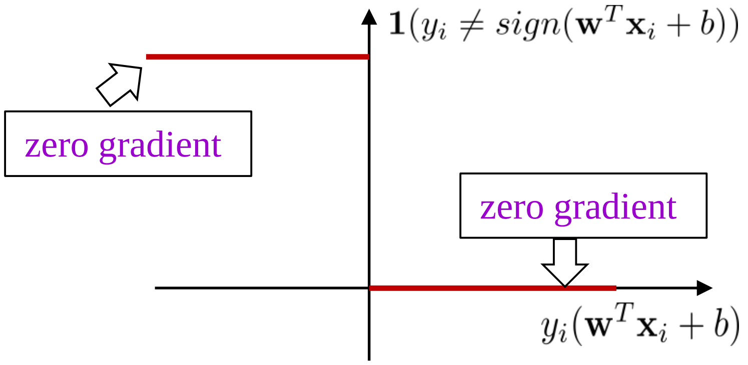
Hard->Half-hard->Soft

Error

Standard loss (error) function

Standard 0/1 loss (gradient 0 nearly everywhere, no gradient feedback):

Training: Minimize $\mathcal{L}(\mathbf{w}, b) = \sum_i \mathbf{1}(y_i \neq sign(\mathbf{w}^T \mathbf{x}_i + b))$



It is the most directly loss, but is also the hardest to minimize.

Zero gradient everywhere!

Half-hard loss (error) function

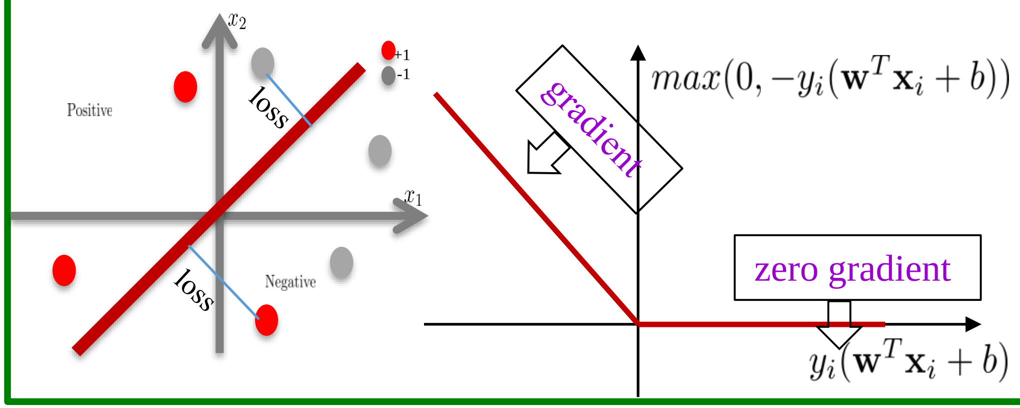
Main motivation

Half->Half-hard->Soft

Error

Loss implicitly used in the perceptron algorithm: with gradient feedback when the target (ground-truth label) and the output (classification) are different).

Training: Minimize $\mathcal{L}(\mathbf{w}, b) = \sum_i max(0, -y_i(\mathbf{w}^T\mathbf{x}_i + b))$



Zero loss for correct classification (no gradient).

A loss based on the distance to the decision boundary for misclassification (with gradient).

Used in the perceptron training.

Soft loss (error) function

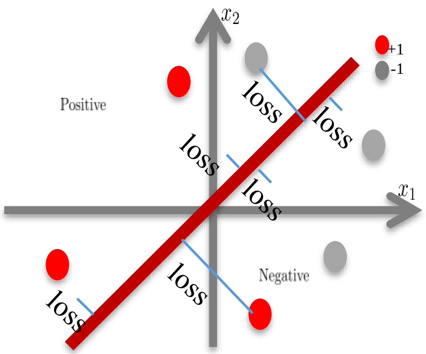
Main motivation

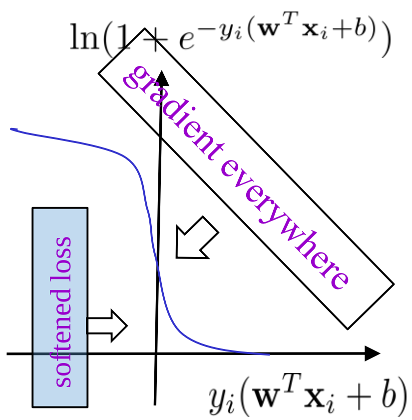
Half->Half-hard->Soft

Error

Loss used in logistic regression.

Training: minimize $\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} \ln(1 + e^{-y_i(\mathbf{w}^T \mathbf{x}_i + b)})$





Every data point receives a loss (gradient everywhere).

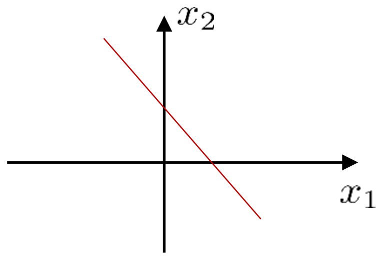
A loss based on the distance to the decision boundary for wrong classification (has a gradient).

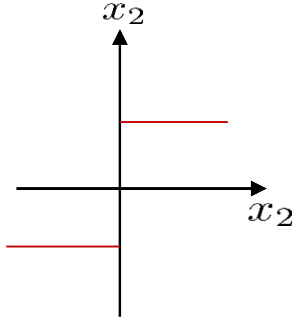
Used in logistic regression classifier.

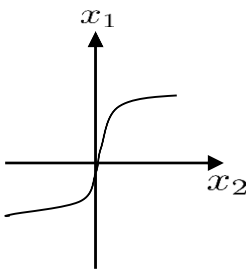
Decision boundary for a logistic regression classifier?

$$\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$f(\mathbf{x}; \mathbf{w}, b) = \begin{cases} +1 & if \ \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} \geq 0.5 \\ -1 & otherwise \end{cases}.$$

A. 

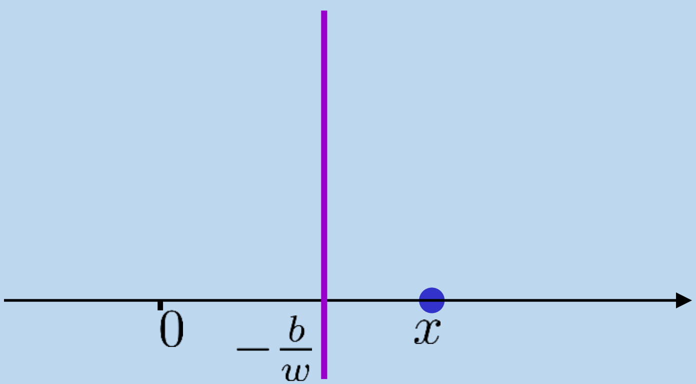
B. 

C. 

D. None of above.

Logistic regression classifier

$x, w, b \in \mathbb{R}$

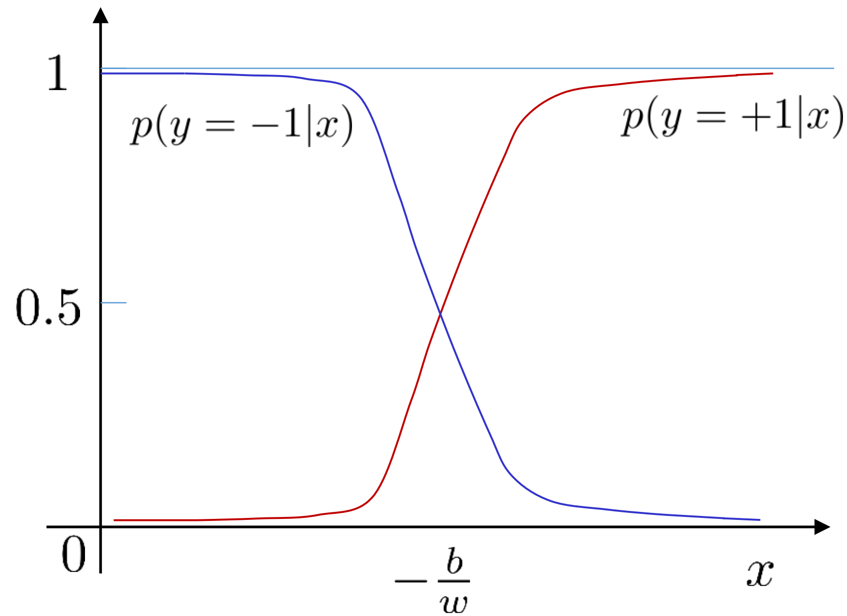


$$w \times x + b \overset{?}{\geq} 0$$

$$w \times (\frac{b}{w} + x) \overset{?}{\geq} 0$$

Let's look at the simplest case where x is a scalar:

Probability of being positive

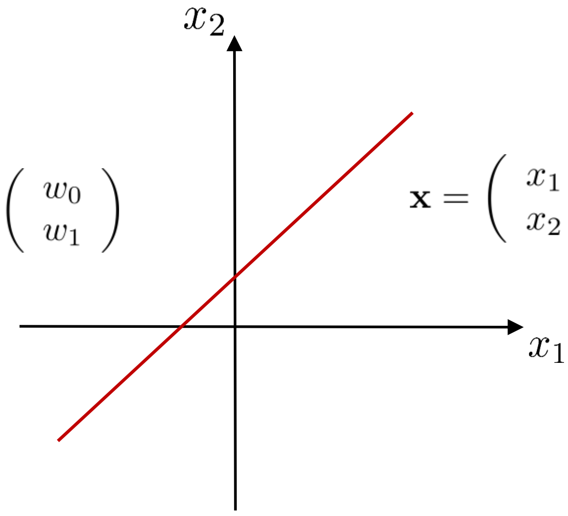


We have: $f(x; w, b) = \begin{cases} +1 & if\ w \times x + b \geq 0 \\ -1 & otherwise \end{cases}$.

$$p(y=+1|x) = \frac{e^{w \times x + b}}{1 + e^{w \times x + b}}$$

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

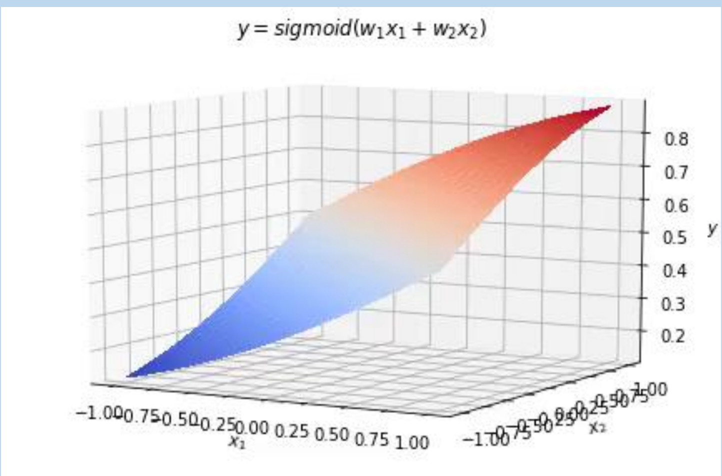
Logistic regression classifier (2D case)



$$\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix}$$

b

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



We have: $f(\mathbf{x}; \mathbf{w}, b) = \begin{cases} +1 & if \;\; \mathbf{w}^T\mathbf{x} + b \geq 0 \\ -1 & otherwise \end{cases}.$

sigmoid function: $\sigma(v) = \frac{1}{1+e^{(-v)}}$.

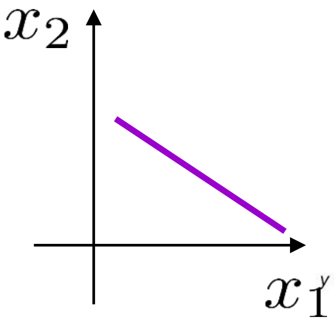
$$p(y = +1|\mathbf{x}) = \sigma\big(\mathbf{w}^T\mathbf{x} + b\big)$$

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

Logistic regression function

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

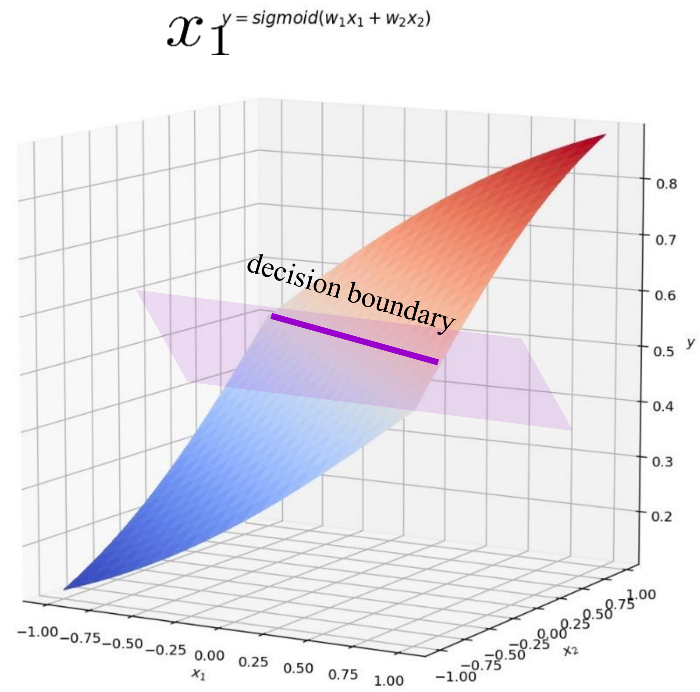
$$p(y=-1|\mathbf{x}) = \frac{1}{1+e^{(\mathbf{w}^T\mathbf{x}+b)}}$$



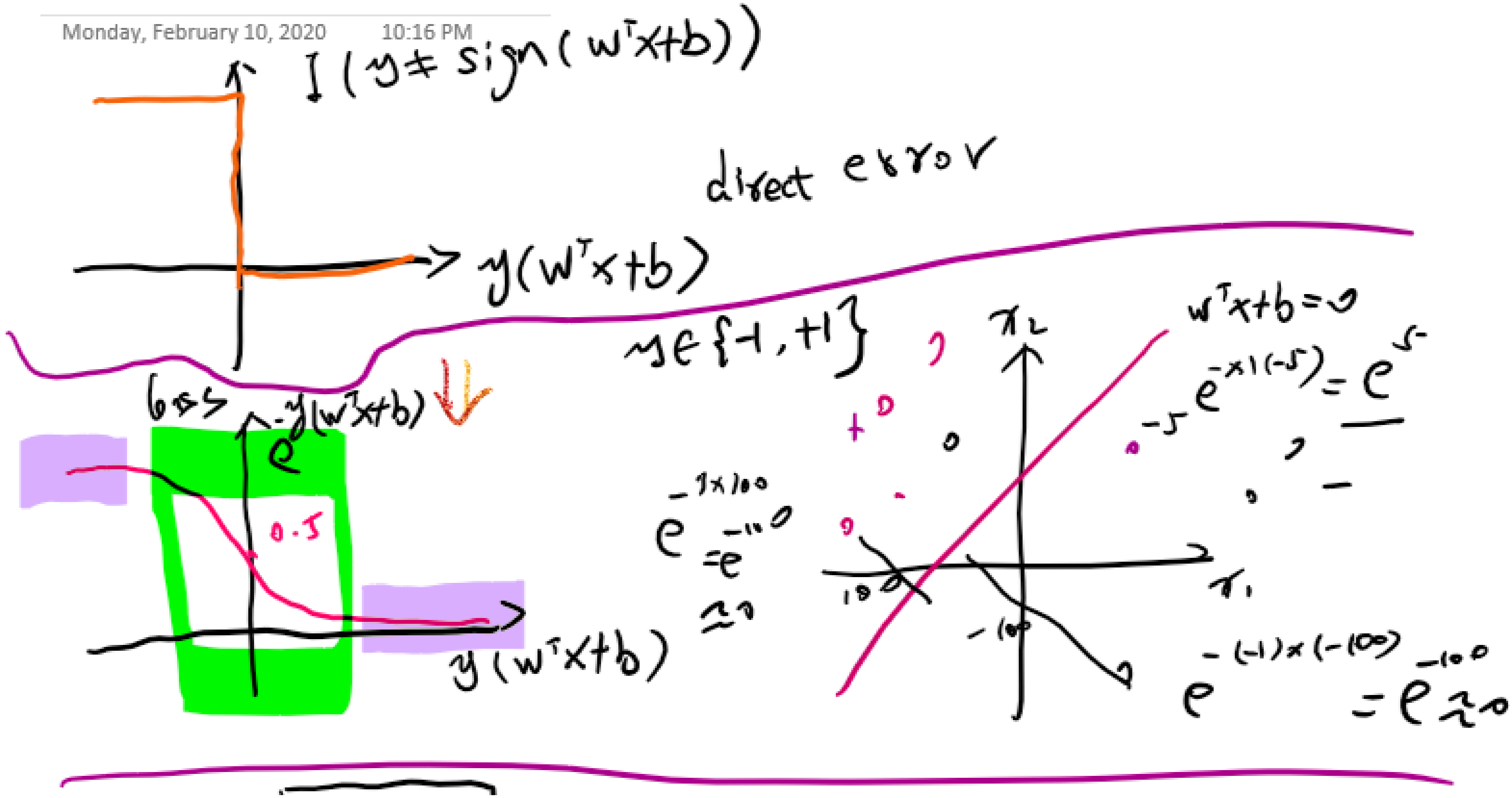
$$\mathbf{x}, \mathbf{w} \in \mathbb{R}^m$$

$$b \in \mathbb{R}$$

$$y \in \{-1, +1\}$$



Monday, February 10, 2020 10:16 PM



$\boxed{w^T x + b}$

$$\boxed{\begin{aligned} p(y=+1 \mid x) &= \frac{1}{1+e^{-(w^T x+b)}} \\ p(y=-1 \mid x) &= \frac{1}{1+e^{+(w^T x+b)}} \end{aligned}} \Rightarrow p(y \mid x) = \frac{1}{1+e^{-y(w^T x+b)}}$$

$$\frac{1}{1+e^{(w^T x+b)}} = \frac{1 \times e^{-(w^T x+b)}}{(1+e^{w^T x+b}) \times e^{-(w^T x+b)}} = \frac{e^{-(w^T x+b)}}{e^{-(w^T x+b)}+1}$$

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

$\Downarrow$

Logit of logistic regression

$$\ln\left(\frac{p(y=+1 \mid x)}{p(y=-1 \mid x)}\right) = \ln\left(\frac{\frac{1}{1+e^{-(w^T x+b)}}}{\frac{1}{1+e^{w^T x+b}}}\right) = \ln\left(\frac{1+e^{w^T x+b}}{1+e^{-(w^T x+b)}}\right)$$

$$= \ln\left(\frac{(1+e^{w^T x+b})\, e^{(w^T x+b)}}{(1+e^{-(w^T x+b)})\, e^{(w^T x+b)}}\right)$$

$\boxed{w^T x + b}$

logit of logistic regression

$$\ln\left(\frac{P(y=+1|x)}{P(y=-1|x)}\right) = \ln\left(\frac{\frac{1}{1+e^{-(w^Tx+b)}}}{\frac{1}{1+e^{w^Tx+b}}}\right) = \ln\left(\frac{1+e^{w^Tx+b}}{1+e^{-(w^Tx+b)}}\right)$$

$$= \ln\left(\frac{(1+e^{w^Tx+b})\,e^{(w^Tx+b)}}{(1+e^{-(w^Tx+b)})\,e^{(w^Tx+b)}}\right)$$

$$= \ln\left(\frac{\cancel{(1+e^{w^Tx+b})}\,e^{(w^Tx+b)}}{\cancel{e^{w^Tx+b}+1}}\right)$$

$$= \ln\left(e^{w^Tx+b}\right) = w^Tx+b$$

$$\boxed{w^Tx+b}$$

$$D = \{(x_i, y_i),\ i = 1 \dots n\}$$

minimize:

$$\boxed{\sum_{i=1}^{n} -\ln p(y_i \mid x_i)}$$

encouraging fitting ground-truth labels.

$$L(w,b) = \sum_{i=1}^{n} -\ln\Big[\underbrace{\frac{1}{1+e^{-y_i(w^T x_i + b)}}}_{=f(w,b)}\Big]$$

$$\frac{\partial L(w,b)}{\partial w} = \sum_{i=1}^{n} - \frac{\dfrac{\boxed{-e^{-y_i(w^T x_i+b)} \cdot (-y_i x_i)}}{\left(1+e^{-y_i(w^T x+b)}\right)^2}}{\dfrac{1}{1+e^{-y_i(w^T x+b)}}}$$

$$\frac{\partial \ln(f(w,b))}{\partial w} = \frac{\frac{\partial f(w,b)}{\partial w}}{f(w,b)}$$

$$\frac{\partial \frac{1}{g(w,b)}}{\partial w} = -\frac{\frac{\partial g(w,b)}{\partial w}}{(g(w,b))^2}$$

$$g(w,b) = 1 + e^{-y_i(w^T x_i + b)}$$

$$\frac{\partial g(w,b)}{\partial w} = \boxed{e^{-y_i(w^T x_i+b)} \cdot (-y_i x_i)}$$

$$\Downarrow$$

$$= \sum_{i=1}^{n} \cancel{-}\ \frac{-e^{-y_i(w^T x_i+b)}}{1+e^{-y_i(w^T x+b)}} \cdot (\cancel{-}\, y_i x_i)$$

$$= \sum_{i=1}^{n} \cancel{\times} \frac{-e^{-y_i(w^T x_i+b)}}{1+e^{-y_i(w^T x+b)}} \times (\cancel{\times} y_i x_i)$$

$$= \sum_{i=1}^{n} -\frac{1+e^{-y_i(w^T x_i+b)} - 1}{1+e^{-y_i(w^T x+b)}} \times (y_i x_i)$$

$$= \sum_{i=1}^{n} -\left[1 - \frac{1}{1+e^{-y_i(w^T x_i+b)}}\right] \times (y_i x_i)$$

$$\boxed{= \sum_{i=1}^{n} -\left(1 - p(y_i|x_i)\right) \times y_i x_i}$$

bad model $p(y_i|x_i) = 0$

$(1-0) \times y_i \times x_i$

good model $p(y_i|x_i) = 1$

$(1-1) \times y_i \times x_i$

Logistic regression function

Below is the main mathematical convenience of the logistic regression function!

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

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

$$p(y=+1|\mathbf{x}) + p(y=-1|\mathbf{x}) = 1$$

$$\frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} + \frac{1}{1+e^{(\mathbf{w}^T\mathbf{x}+b)}}$$

$$= \frac{e^{(\mathbf{w}^T\mathbf{x}+b)}}{e^{(\mathbf{w}^T\mathbf{x}+b)}+1} + \frac{1}{1+e^{(\mathbf{w}^T\mathbf{x}+b)}} = \frac{e^{(\mathbf{w}^T\mathbf{x}+b)}+1}{1+e^{(\mathbf{w}^T\mathbf{x}+b)}} = 1$$

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

A general form, independent of the value of y!

Training a logistic regression classifier

$$S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$$

$$\mathbf{x}_i \in \mathbb{R}^m, i = 1..n \qquad y_i \in \{-1, +1\}, i = 1..n$$

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

Model parameters:

$\mathbf{w} \in \mathbb{R}^m$

$b \in \mathbb{R}$

Training a logistic regression classifier

$$(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}$$

$$= \arg\max_{(\mathbf{w},b)} \ln\left(\prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$$

$$= \arg\min_{(\mathbf{w},b)} -\ln\left(\prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$$

Question: which is the correct answer for the optimal solution?

Answer A: $(\mathbf{w}, b)^* = \arg\min_{(\mathbf{w},b)} \sum_{i=1}^{n} -\ln\left(\frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$

Answer B: $(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \sum_{i=1}^{n} -\ln\left(\frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$

Answer C: $(\mathbf{w}, b)^* = \arg\min_{(\mathbf{w},b)} -\ln\left(\sum_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$

Training a logistic regression classifier

Intuition: find the best parameters $(\mathbf{w}, b)^*$ to maximize the probabilities of fitting the ground-truth label y_i for each $\mathbf{x}_i$.

Math: $(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}$

$$(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}$$

$$= \arg\max_{(\mathbf{w},b)} \ln(\prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T x_i+b)}})$$

$$= \arg\min_{(\mathbf{w},b)} \sum_{i=1}^{n} -\ln(\frac{1}{1+e^{-y_i\times(\mathbf{w}^T\mathbf{x}_i+b)}})$$

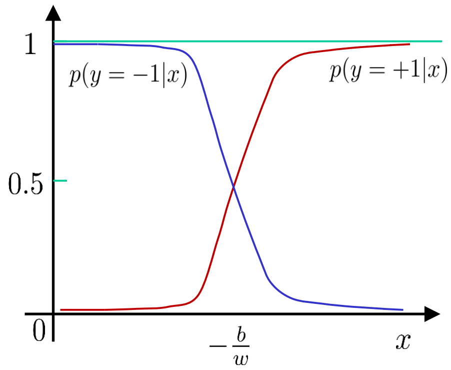
Training a logistic regression classifier

$S_{training} = \{(-1.1, -1), (3.2, +1), (2.5, -1), (5.0, +1), (4.3, +1)\}$

Train a logistic regression classifier $f(\mathbf{x}) = \begin{cases} +1 & if\ \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} \geq 0.5 \\ -1 & otherwise \end{cases}$:

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

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



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

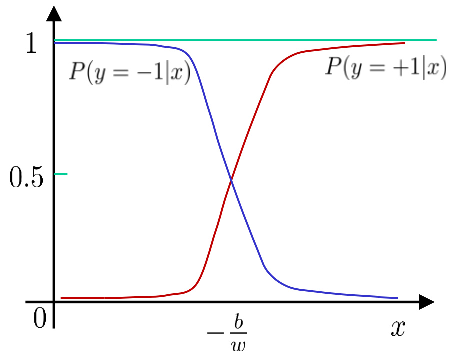
Intuition: find the best parameters $(\mathbf{w}, b)^*$ to maximize the probabilities of fitting the ground-truth label y_i for each $\mathbf{x}_i$.

Math: $(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \prod_{i=1}^{n} \frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}$

Training a logistic regression classifier

$$S_{training} = \{(-1.1, -1), (3.2, +1), (2.5, -1), (5.0, +1), (4.3, +1)\} \quad y_i \in \{-1, +1\}, i = 1..n$$

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



$$(\mathbf{w}, b)^* = \arg\max_{(\mathbf{w},b)} \prod_{i=1}^{n} [p(y_i|\mathbf{x}_i)]$$

$$(w,b)^* = \arg\min_{(w,b)} -\sum_{i=1}^{n} \ln\left(\frac{1}{1+e^{-y_i\times(w\times x_i+b)}}\right) = \arg\min_{(w,b)} \sum_{i=1}^{n} \ln(1+e^{-y_i\times(w\times x_i+b)})$$

$$(w,b)^* = \arg\min_{(w,b)} [\ln(1+e^{(-1.1w+b)}) + \ln(1+e^{-(3.2w+b)}) +$$

$$\ln(1+e^{(2.5w+b)}) + \ln(1+e^{-(5.0w+b)}) + \ln(1+e^{-(4.3w+b)})]$$

Training a logistic regression classifier

$$\mathbf{x}_i \in \mathbb{R}^m, i = 1..n \qquad y_i \in \{-1, +1\}, i = 1..n$$

Model parameters: $\mathbf{w} \in \mathbb{R}^m$ and $b \in \mathbb{R}$

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

Intuition: find the best parameters $(\mathbf{w}, b)^*$ to maximize the probabilities of fitting the ground-truth label y_i for each x_i.

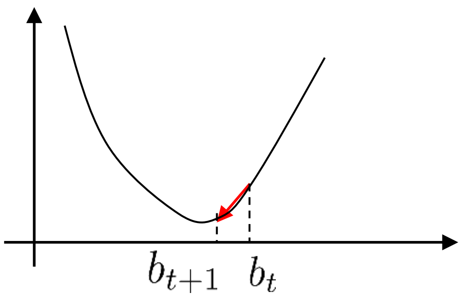
Math: $(\mathbf{w}, b)^* = \arg\min_{(\mathbf{w},b)} \sum_{i=1}^{n} -\ln\left(\frac{1}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}\right)$

$$= \arg\min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b)$$

Multivariate input

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

Train a logistic regression classifier $f(x) = \begin{cases} +1 & if \ \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} \geq 0.5 \\ -1 & otherwise \end{cases}$:



$$(\mathbf{w}, b)^* = \arg\min_{(\mathbf{w},b)} \mathcal{L}(\mathbf{w}, b)$$

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} \ln(1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)})$$

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b) = \sum_i \frac{-y_i\mathbf{x}_i e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}} = \sum_i -y_i\mathbf{x}_i(1 - p(y_i|\mathbf{x}_i))$$

$$\nabla_b\mathcal{L}(\mathbf{w}, b) = \sum_i \frac{-y_i e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}} = \sum_i -y_i(1 - p(y_i|\mathbf{x}_i))$$

Derivation of the derivative for the logistic regression classifier

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} \ln(1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i + b)})$$

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

$$\frac{\partial \mathcal{L}(\mathbf{w},b)}{\partial \mathbf{w}} = \sum_{i=1}^{n} \frac{\partial \ln(1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)})}{\partial \mathbf{w}}$$

$$= \sum_{i=1}^{n} \frac{\frac{\partial(1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)})}{\partial \mathbf{w}}}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}$$

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

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

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

$$= \sum_i -y_i\mathbf{x}_i(1 - p(y_i|\mathbf{x}_i))$$

Gradient: A Rule of Thumb

For a linear model:

$$\mathbf{w}^T\mathbf{x} + b$$

If you define a general loss function:

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} cost(y_i, \mathbf{w}^T\mathbf{x}_i + b)$$

The gradient to update the w is nearly always in a form of a weighted combination of the input x.

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b) = \sum_i difference(y_i, \mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i$$

The large the difference between the ground-truth label y_i and the prediction $\mathbf{w}^T\mathbf{x} + x$ is, the higher weight it is.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda_t \times \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_t, b_t)$$

Gradient: A Rule of Thumb

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b) = \sum_i difference(y_i, \mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i$$

The larger the difference between the ground-truth label y_i and the prediction $\mathbf{w}^T\mathbf{x} + x$ is, the higher weight, $difference(y_i, \mathbf{w}^T\mathbf{x}_i + b)$, it is.

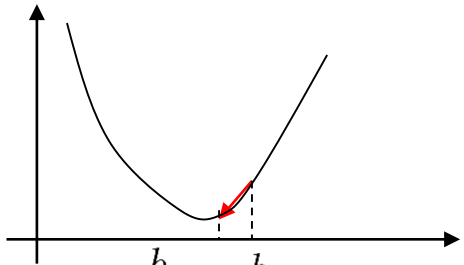
That is:

For any data point x_i, if the current model parameters (**w**,b) makes a good prediction, then x_i makes less contribution to the change (being happy so wanting no change).

If the current model parameter (**w**,b) makes a bad prediction for a data x_i, then this point x_i makes a large contribution to the change (unhappy so change the parameter for me please!).

Multivariate input

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} \ln(1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i + b)})$$



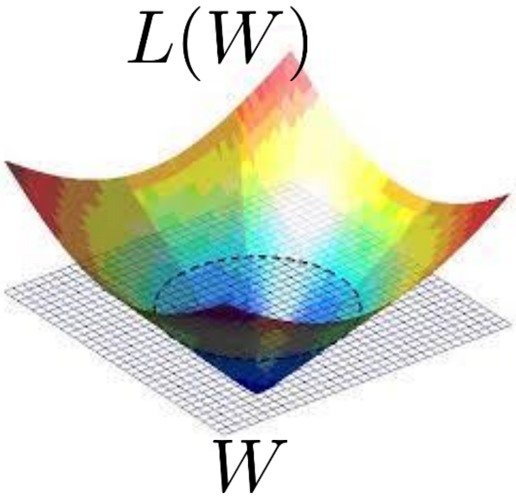
$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b) = \sum_i \frac{-y_i\mathbf{x}_i e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}} = \sum_i -y_i\mathbf{x}_i(1 - p(y_i|\mathbf{x}_i))$$

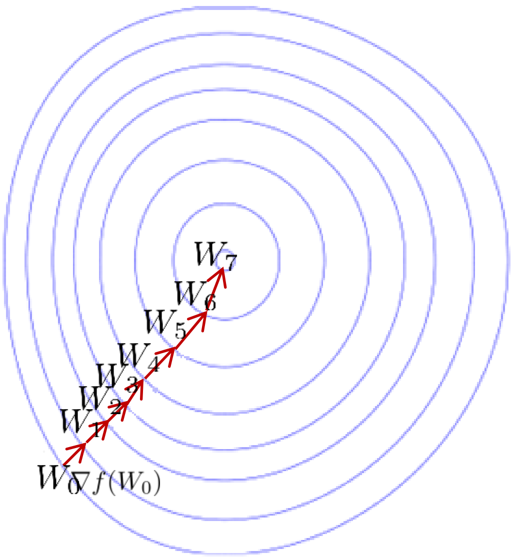
$$\nabla_{b}\mathcal{L}(\mathbf{w}, b) = \sum_i \frac{-y_i e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}}{1+e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)}} = \sum_i -y_i(1 - p(y_i|\mathbf{x}_i))$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda_t \times \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_t, b_t)$$

$$b_{t+1} = b_t - \lambda_t \times \nabla_b\mathcal{L}(\mathbf{w}_t, b_t)$$

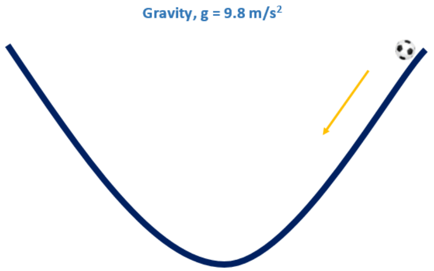
Gradient descent



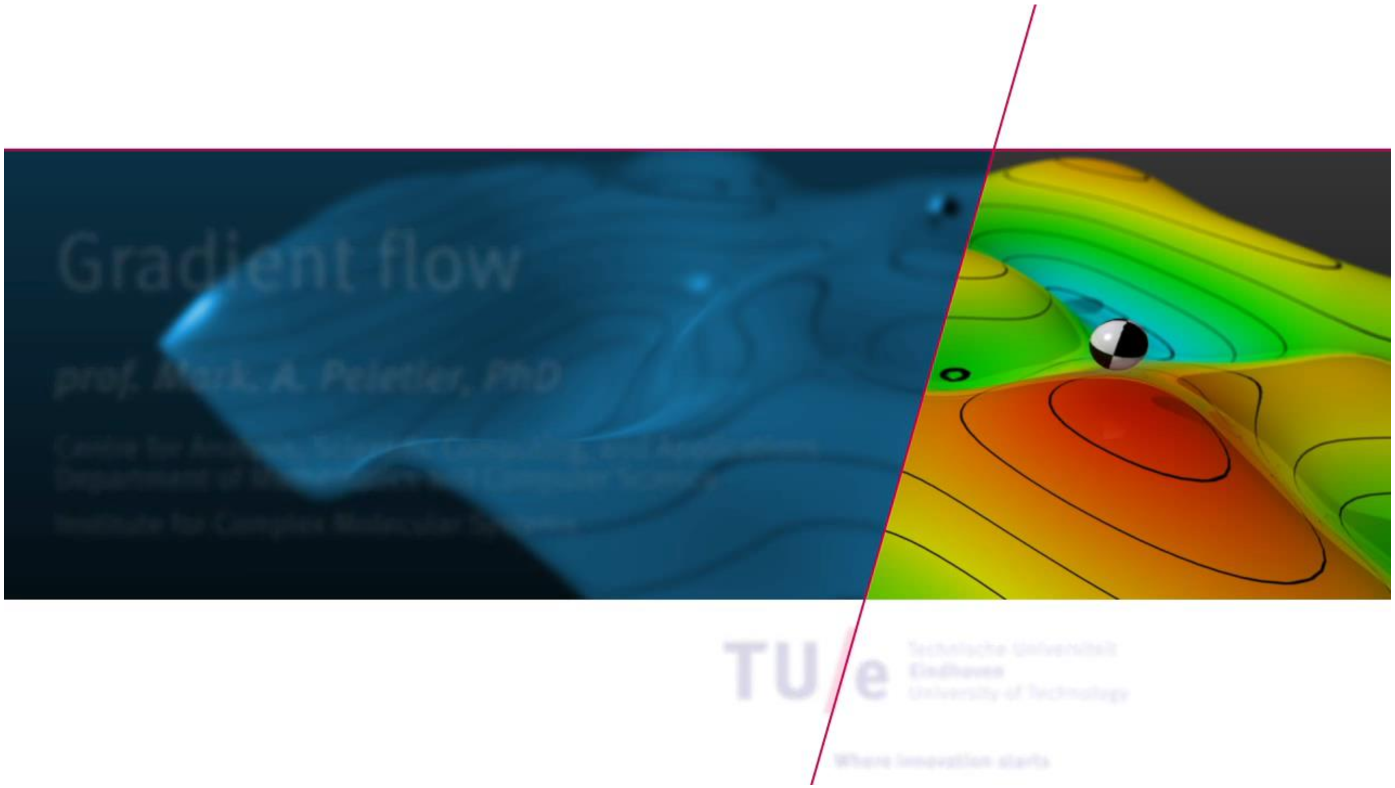


$$W_{t+1} \leftarrow W_t - \lambda_t \nabla L(W_t) \quad \lambda_t : stepsize$$

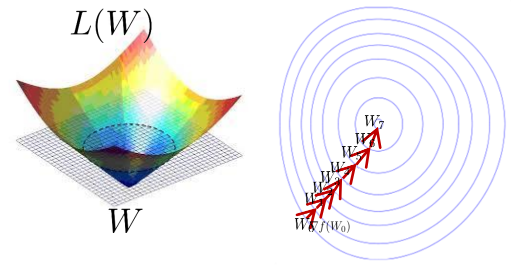
Gradient decent animation



https://www.kaggle.com/abdalimran/intuition-of-gradient-descent-for-machine-learning



https://www.youtube.com/watch?v=vWFjqgb-ylQ



$$W_{t+1} \leftarrow W_t - \lambda_t \nabla L(W_t) \quad \lambda_t : stepsize$$

The gradient decent algorithm

1. The gradient decent algorithm is one of the most widely used optimization methods in machine learning.
2. It can be applied to both convex and non-convex functions.
3. For non-convex functions, no guarantee to find the globally optimal solution but local optimums are ok in practice.
4. Finding the proper learning rates (not always fixed) is an important research topic for gradient decent.
5. Typically, you can start by using a small fixed learning rate when understanding the algorithm and your problem.

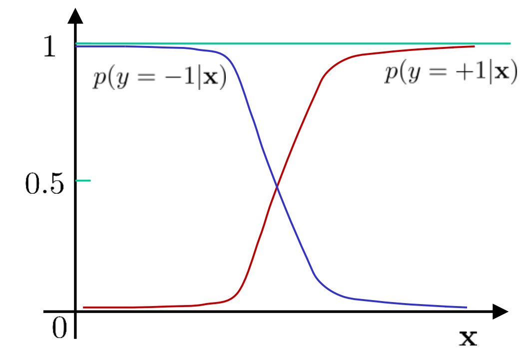
Logistic regression classifier

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

$$\mathbf{x} \in \mathbb{R}^m$$

$$y \in \{-1, +1\}$$

$$f(\mathbf{x}) = \begin{cases} +1 & if\ \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} \geq 0.5 \\ -1 & otherwise \end{cases}$$



Pros:

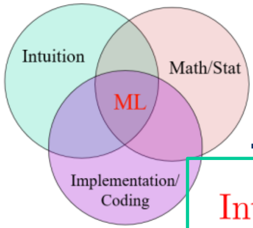
1. It is well-normalized.
2. Easy to turn into probability.
3. Easy to implement.

Cons:

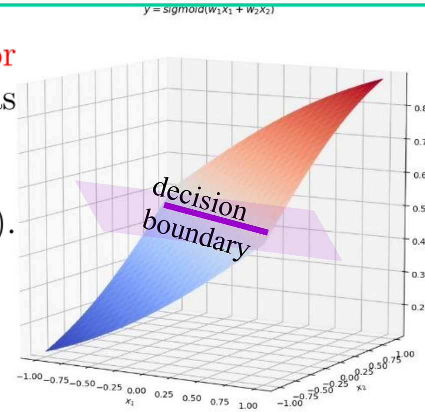
1. Indirect loss function.
2. Dependent on good feature set.
3. Weak on feature selection.

Take home message

- Logistic regression classifier is still a linear classifier but with a probability output.
- It can be trained using a gradient descent algorithm.
- The "regression" refers to fitting the discriminative probabilities: $p(y|\mathbf{x})$
- It has been widely adopted in practice, especially in the modern deep learning era.



Recap: Logistic Regression Classifier



Intuition: Logistic regression classifier nicely turns a hard classification error (0 *or* 1) into a soft measure using the sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$ which has three particularly appealing properties:

- A soft measure that maps any value $v \in (-\infty, \infty)$ to a normalized $\rightarrow (0, 1)$.
- Nice gradient form.
- Convex function for the objective function in training.

Math:

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

Training :

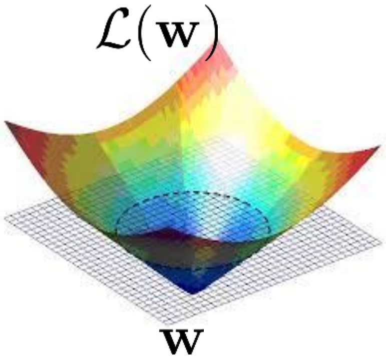
$$(\mathbf{w}, b)^* = \arg\min_{(\mathbf{w},b)} \mathcal{L}(\mathbf{w}, b) = \arg\min_{(\mathbf{w},b)} \sum_{i=1}^{n} \ln(1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i+b)})$$

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b) = \sum_i -y_i \times \mathbf{x}_i(1 - p(y_i|\mathbf{x}_i))$$

$$\nabla_b\mathcal{L}(\mathbf{w}, b) = \sum_i -y_i \times (1 - p(y_i|\mathbf{x}_i))$$

Recap: Logistic Regression Classifier

Implementation:



Gradient Descent Direction

(a) Pick a direction $\nabla \mathcal{L}(\mathbf{w}_t, b_t)$

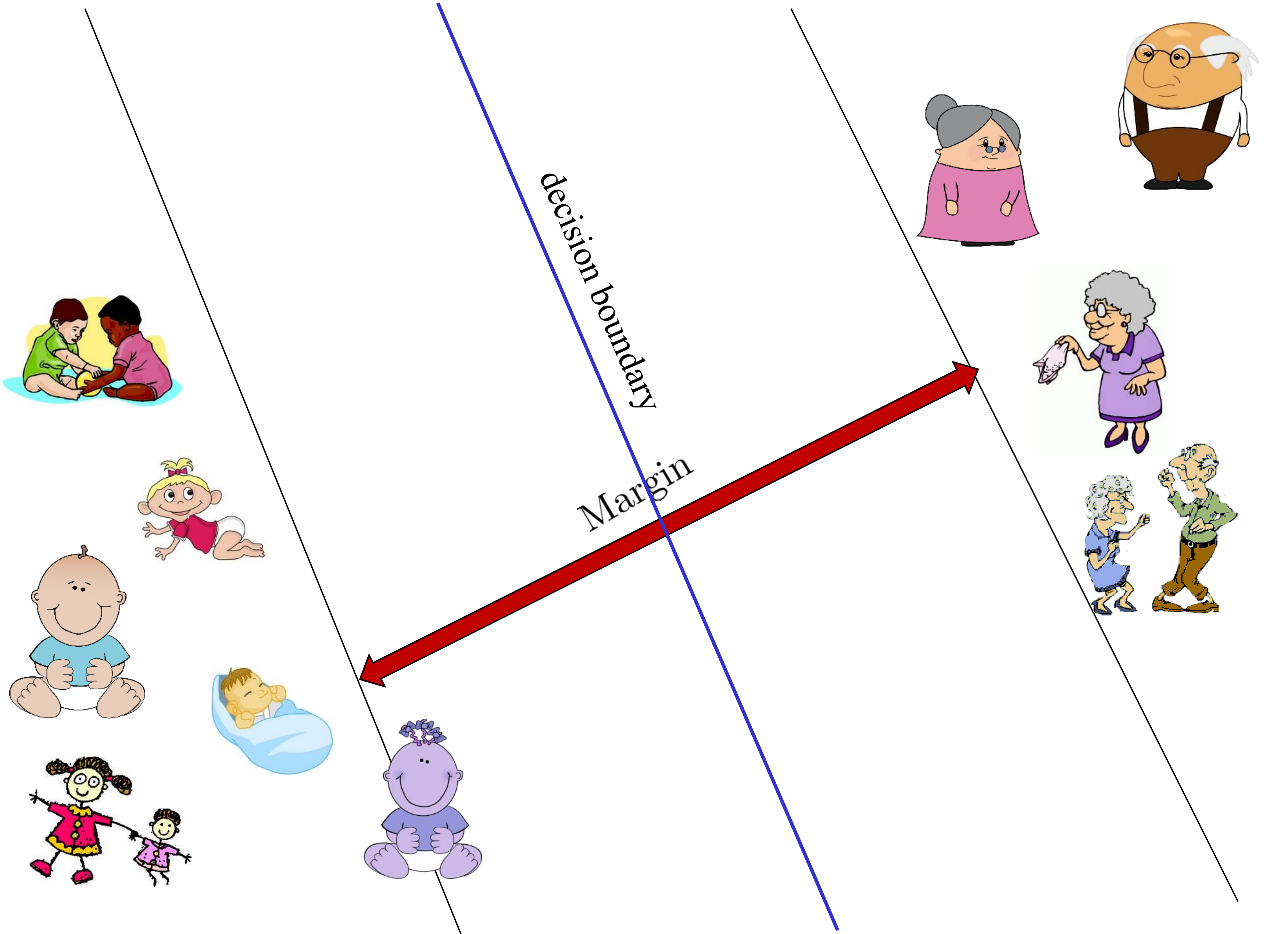
(b) Pick a step size λ_t

(c) $\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda_t \times \nabla \mathcal{L}_{\mathbf{w}_t}(\mathbf{w}_t, b_t)$ such that function decreases;
$b_{t+1} = b_t - \lambda_t \times \nabla \mathcal{L}_{b_t}(\mathbf{w}_t, b_t)$

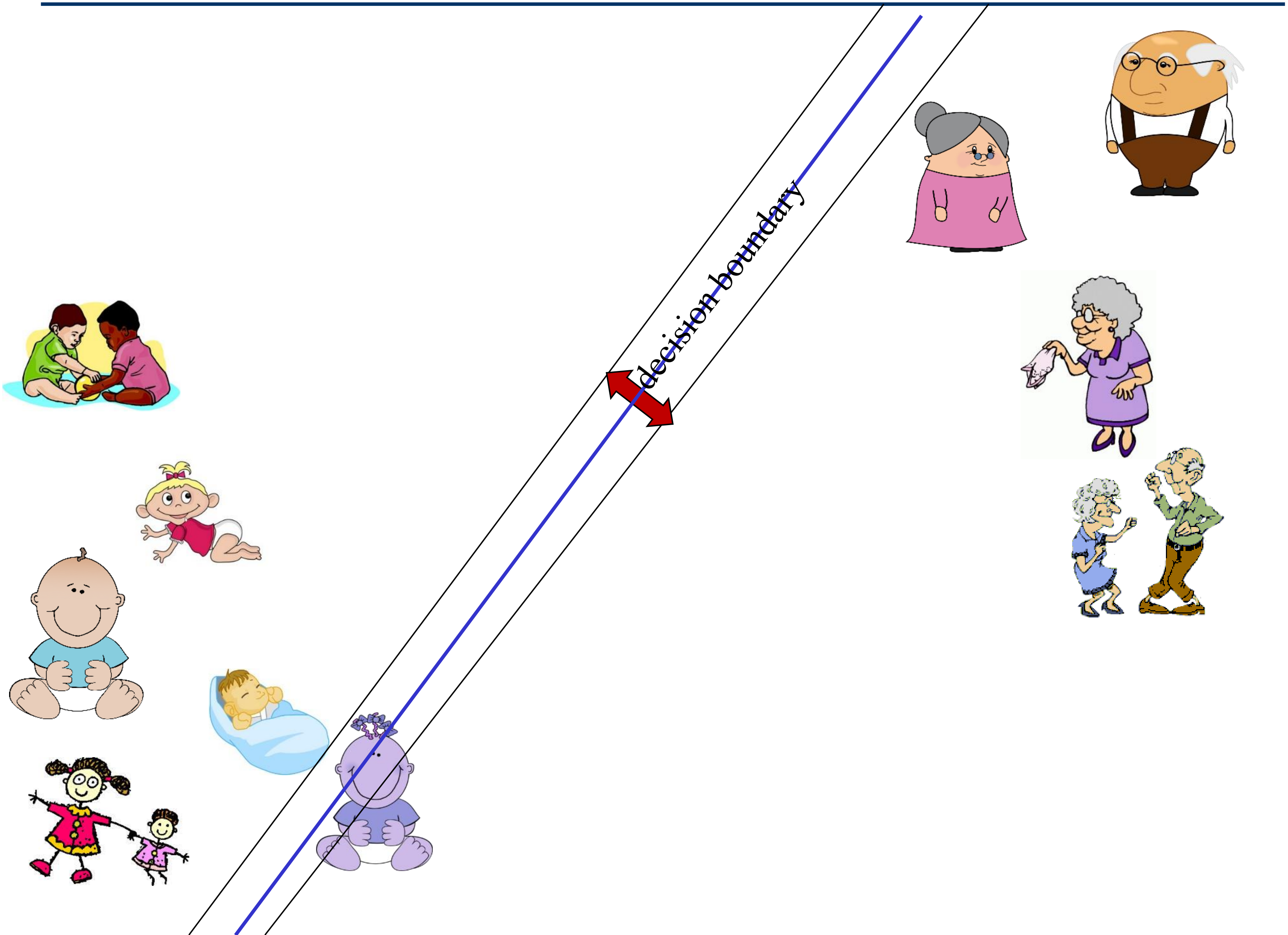
(d) Repeat

Support Vector Machine

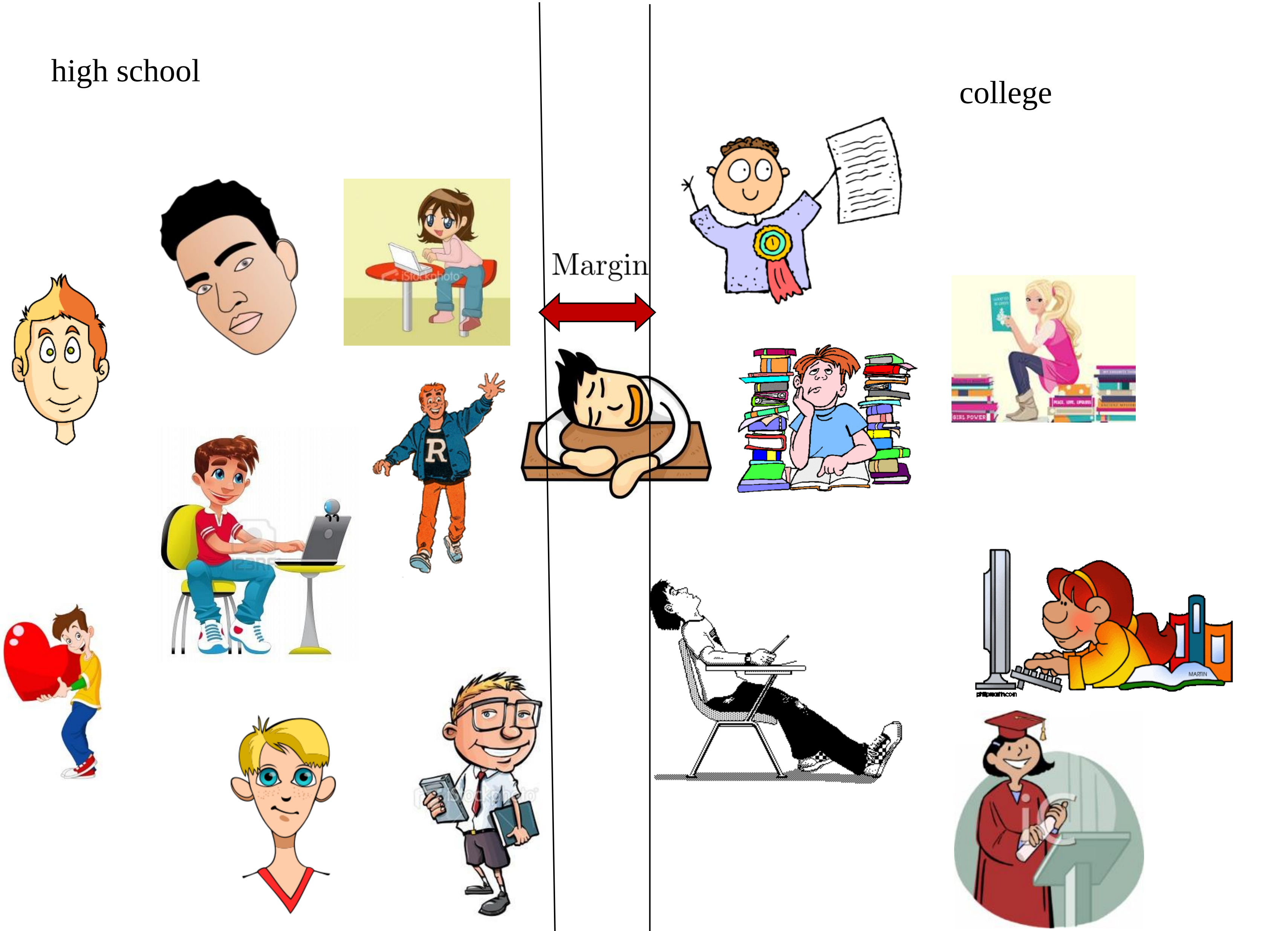
Why large margin?



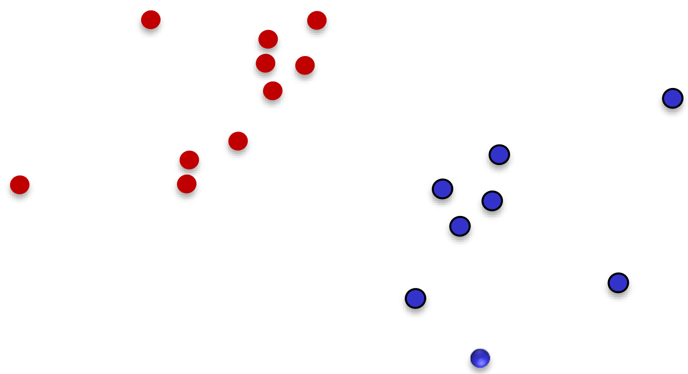
Why large margin?



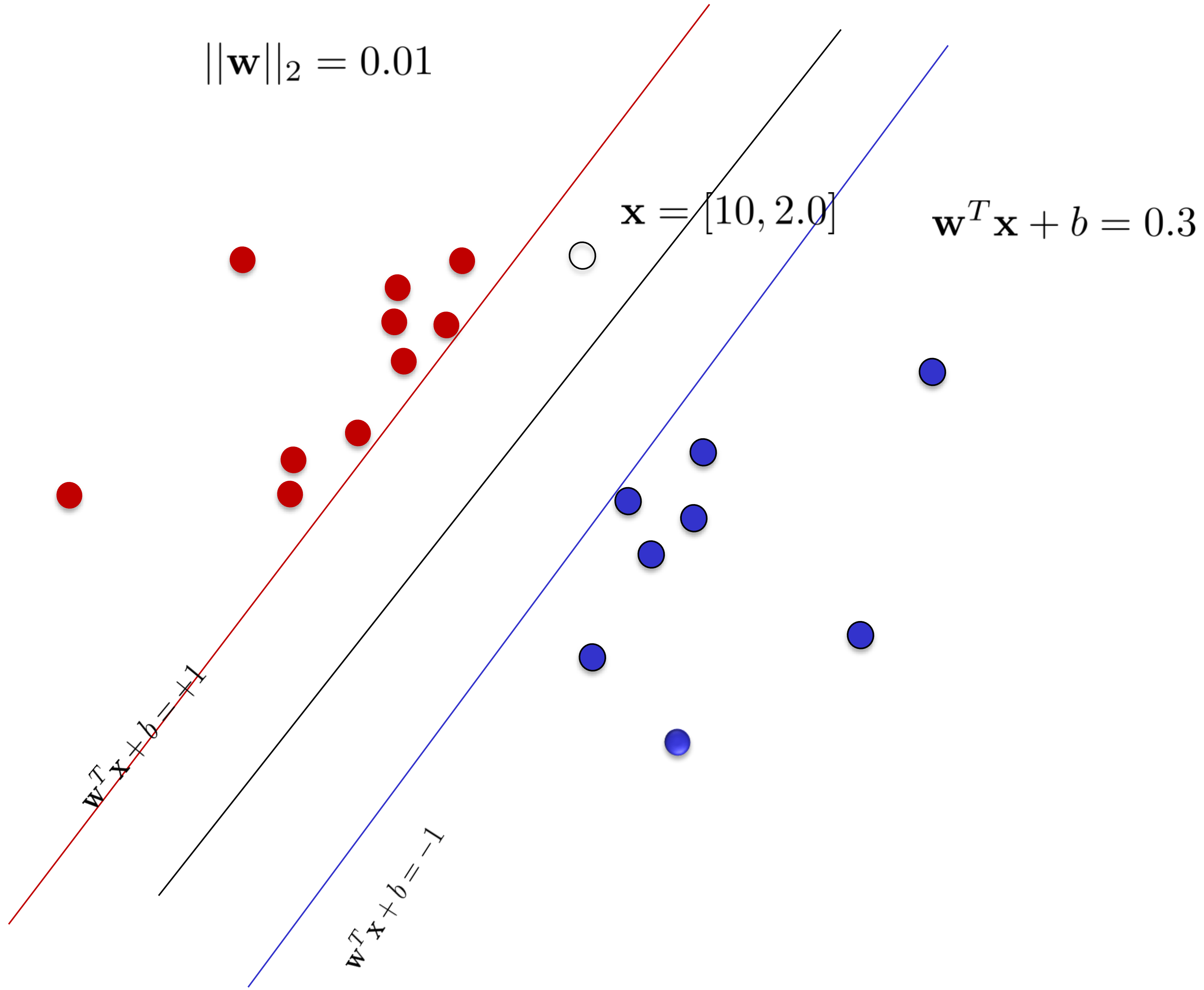
Why large margin



How to understand



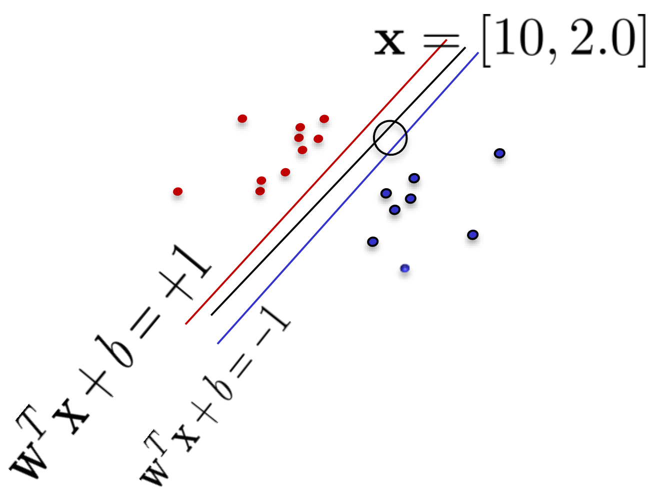
How to understand: a large margin



How to understand: a small margin

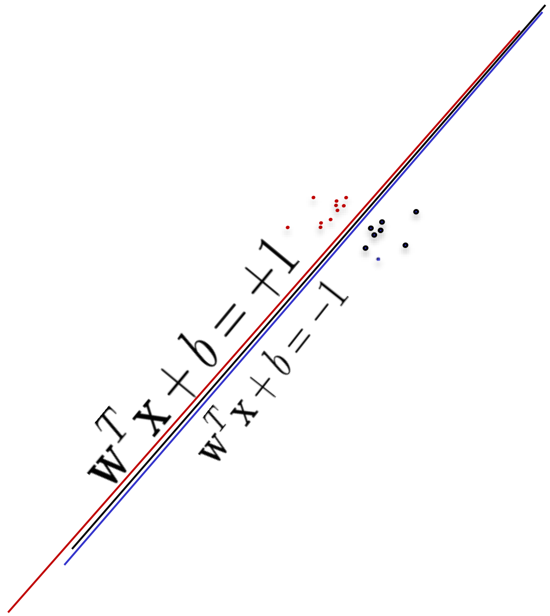
$$||\mathbf{w}||_2 = 100$$

$$\mathbf{w}^T\mathbf{x} + b = -8.0$$



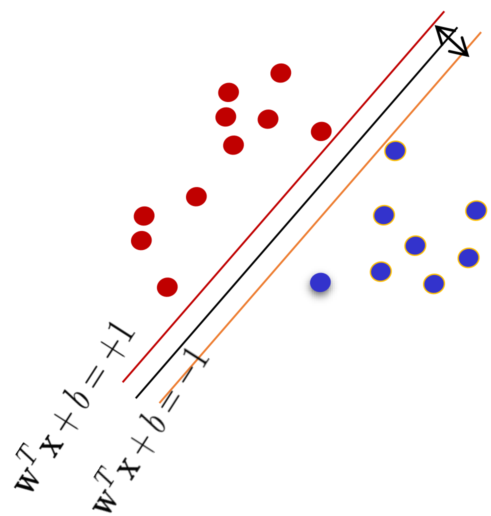
How to understand: a small margin

$$||\mathbf{w}||_2 = 10,000$$



Why margin?

$$e_{testing} \leq e_{training} + \sqrt{\frac{h(\log(2n/h+1)) - \log(\eta/4)}{n}}$$



$$M = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$$

$$\mathbf{w}^T\mathbf{w} = ||\mathbf{w}||^2$$

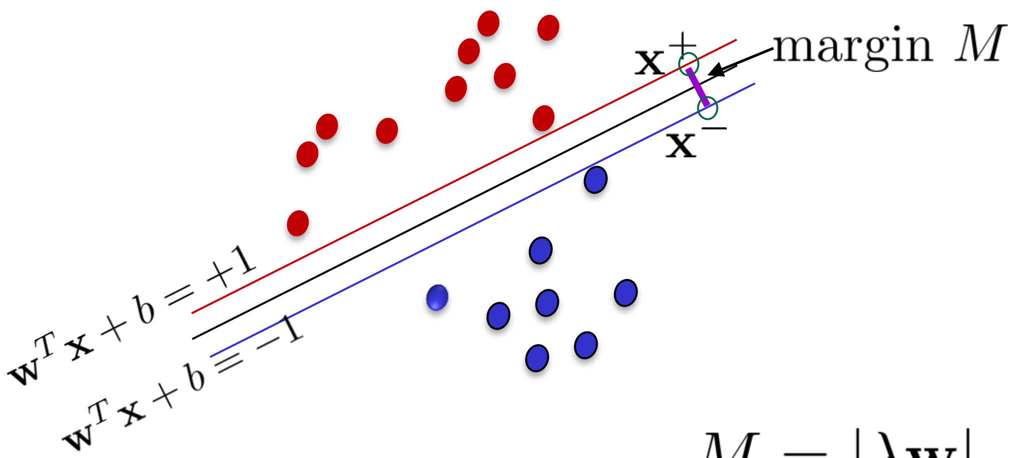
$$\text{Find: } \arg\min_{\mathbf{w}} C \times (\#training\ errors) + \frac{1}{2}||\mathbf{w}||^2$$

Why is $\sqrt{\frac{h(\log(2n/h+1)) - \log(\eta/4)}{n}}$ related to $||\mathbf{w}||^2$?

In machine learning, a term called "regularization", has been frequently used to prevent overfitting.

"Margin" is a term researchers typically use to "regularize" the underlying classifier (there are of course other ways to impose regularization https://en.wikipedia.org/wiki/Regularization_(mathematics)).

Computing the margin width



$$\mathbf{w}^T\mathbf{x}^- + b = -1$$

$$\mathbf{x}^+ = \mathbf{x}^- + \lambda\mathbf{w}$$

$$\mathbf{w}^T\mathbf{x}^+ + b = +1$$

$$\text{Margin: } M = ||\mathbf{x}^+ - \mathbf{x}^-||_2$$
$$= ||\lambda\mathbf{w}||_2 \in \mathbb{R}$$

$$M = |\lambda\mathbf{w}|$$

$$\lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} \qquad \lambda \in \mathbb{R}$$

$$||\mathbf{w}||_2 = \sqrt{\mathbf{w}^T\mathbf{w}}$$

$$\Downarrow$$

$$M = ||\lambda\mathbf{w}||_2 = \frac{2\sqrt{\mathbf{w}^T\mathbf{w}}}{\mathbf{w}^T\mathbf{w}}$$
$$= \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$$