Artificial Intelligence
——Perceptron Learning Algorithm

Yanghui Rao
Assistant Prof., Ph.D
School of Data and Computer Science,
Sun Yat-sen University
raoyangh@mail.sysu.edu.cn

课前问答

- k-NN模型的优缺点有哪些？用k-NN开展分类或回归任务的结果是否稳定？不稳定的话，主要的影响因素有哪些?
- 在基于k-NN的分类任务中，当k取值为训练集的样本个数时，结果是否稳定?
- 朴素贝叶斯分类模型的假设是什么?
- 在实验设计中，验证集的作用是什么?

线性回归

- Least-squares solutions

$$n^{-1}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0$$

$$n^{-1}\sum_{i=1}^{n}x_i(y_i - w_0 - w_1 x_i) = 0$$

$$Q(w_0, w_1) = \min_{w_0, w_1}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$

$$\partial Q(w_0, w_1) / \partial w_0 = 0 \qquad \partial Q(w_0, w_1) / \partial w_1 = 0$$

$$-2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0 \qquad -2\sum_{i=1}^{n}x_i(y_i - w_0 - w_1 x_i) = 0$$

线性回归

- Least-squares solutions

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$w_1 = \frac{\sum_{i=1}^{n} x_i (y_i - \bar{y})}{\sum_{i=1}^{n} x_i (x_i - \bar{x})}$$

$$= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Perceptron Learning Algorithm

- Dealing with all attributes jointly which are continuous variables
- Discrete random variables can be changed to continuous variables

Perceptron Learning Algorithm

- Dealing with all attributes jointly which are continuous variables
- Discrete random variables can be changed to continuous variables
- For $\mathbf{x}=(x_1,x_2,\ldots,x_d)$ with d features, compute a weighted 'score' and
  predict +1(good) if $\sum_{k=1}^{d} w_k x_k > threshold$
  predict -1(bad) if $\sum_{k=1}^{d} w_k x_k < threshold$
- $\mathbf{y}$={+1(good), -1(bad)}

$$h(\mathbf{x}) = sign\left(\left(\sum_{k=1}^{d} w_k x_k\right) - threshold\right)$$

Perceptron Learning Algorithm

$$h(\mathbf{x}) = sign\left(\left(\sum_{k=1}^{d} w_k x_k\right) - threshold\right)$$

$$= sign\left(\left(\sum_{k=1}^{d} w_k x_k\right) + \underbrace{(-threshold)}_{w_0} \cdot \underbrace{(+1)}_{x_0}\right)$$

$$= sign\left(\sum_{j=0}^{d} w_j x_j\right)$$

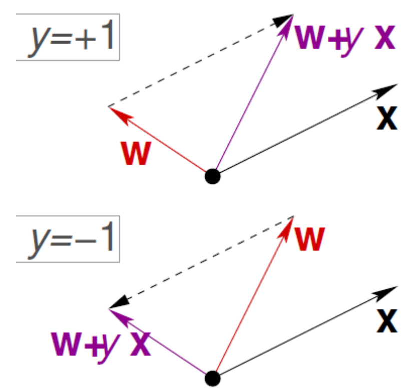
$$= sign\left(\tilde{\mathbf{W}}^{\mathrm{T}} \tilde{\mathbf{X}}\right)$$

Perceptron Learning Algorithm

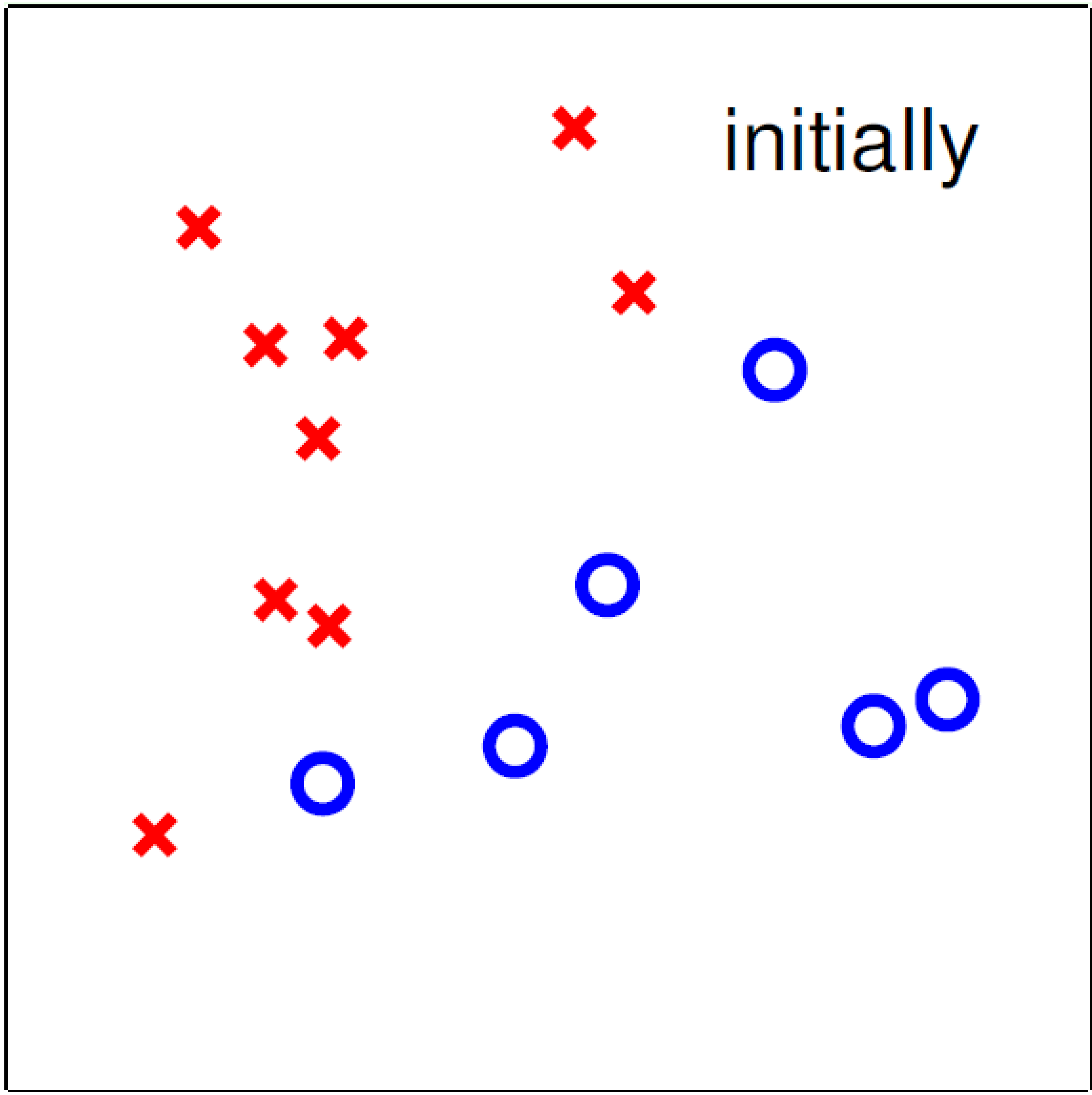
- **Difficult:** the set of $h(\mathbf{x})$ is of infinite size
- **Idea:** start from some initial weight vector $\mathbf{w}_{(0)}$, and "correct" its mistakes on D
- For t = 0, 1,…
  - find a mistake of $\mathbf{w}_{(\mathrm{t})}$ called $(\mathbf{x}_{i(t)}, y_{i(t)})$

    $$sign\left(\tilde{\mathbf{w}}_{(t)}^{\mathrm{T}}\tilde{\mathbf{x}}_{i(t)}\right) \neq y_{i(t)}$$
  - (try to) correct the mistake by

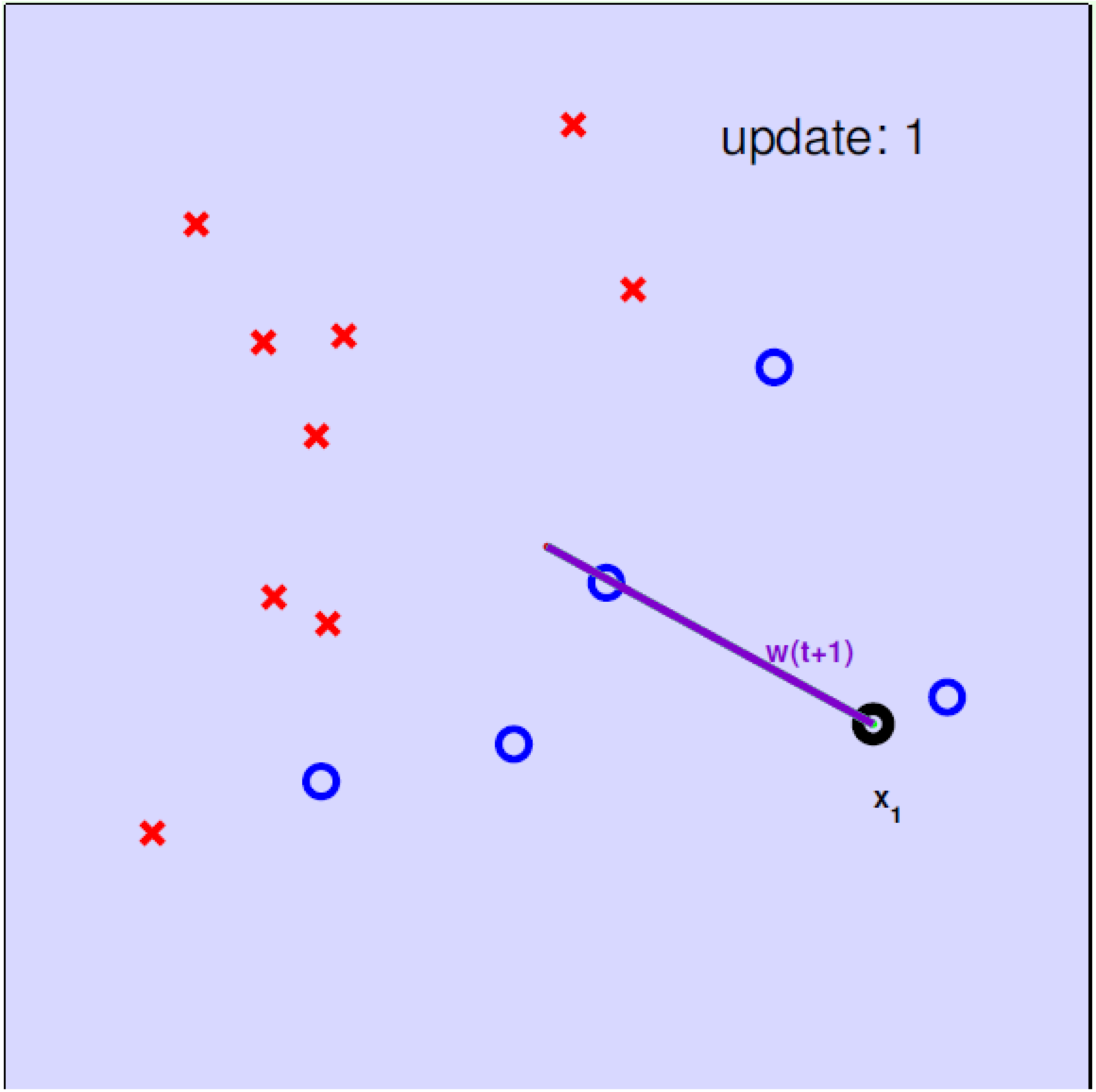
    $$\tilde{\mathbf{w}}_{(t+1)} \leftarrow \tilde{\mathbf{w}}_{(t)} + y_{i(t)}\tilde{\mathbf{x}}_{i(t)}$$
  - until no more mistakes
- Return last $\mathbf{W}$ (called $\mathbf{W}_{\mathrm{PLA}}$)



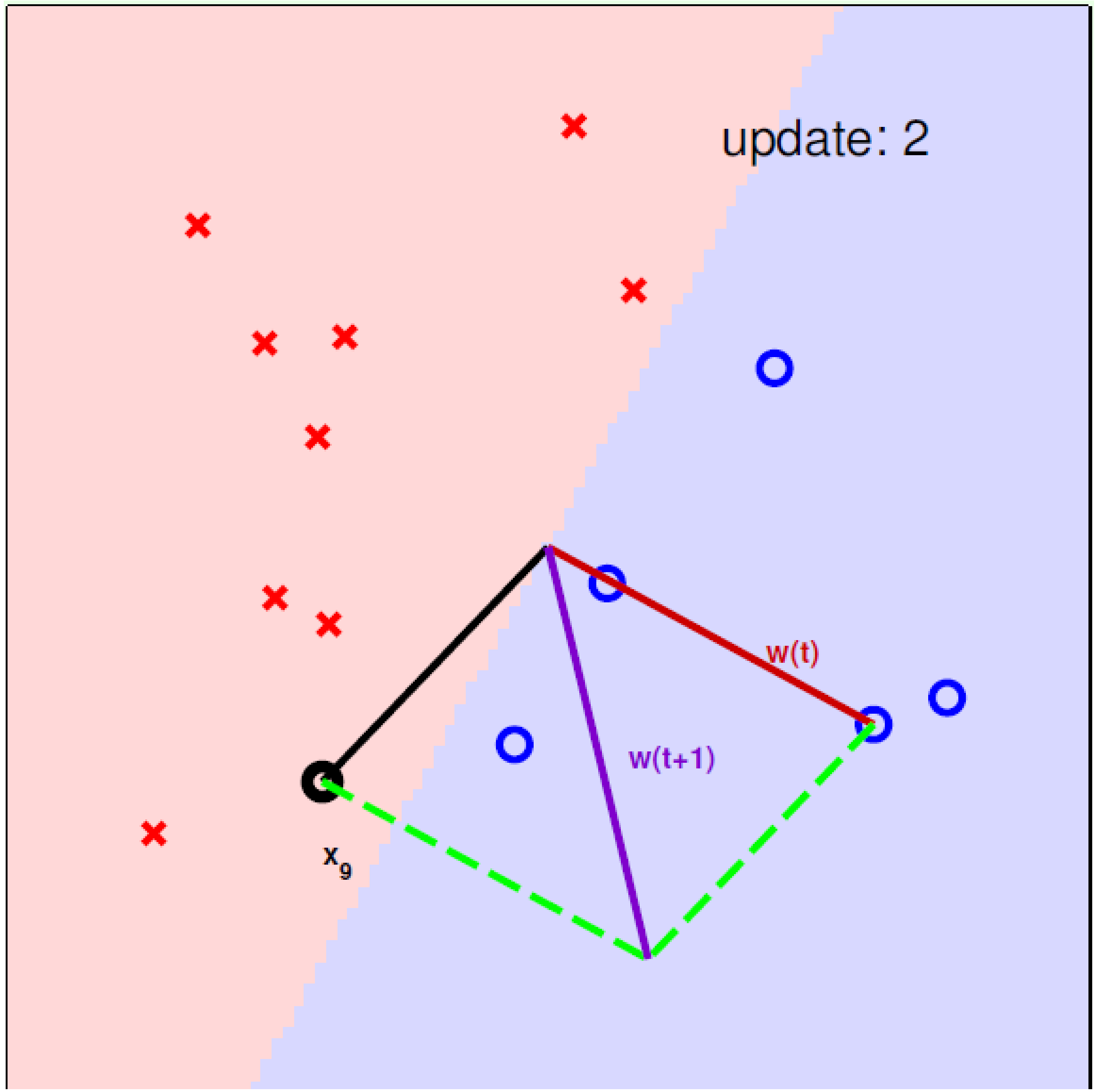
Perceptron Learning Algorithm



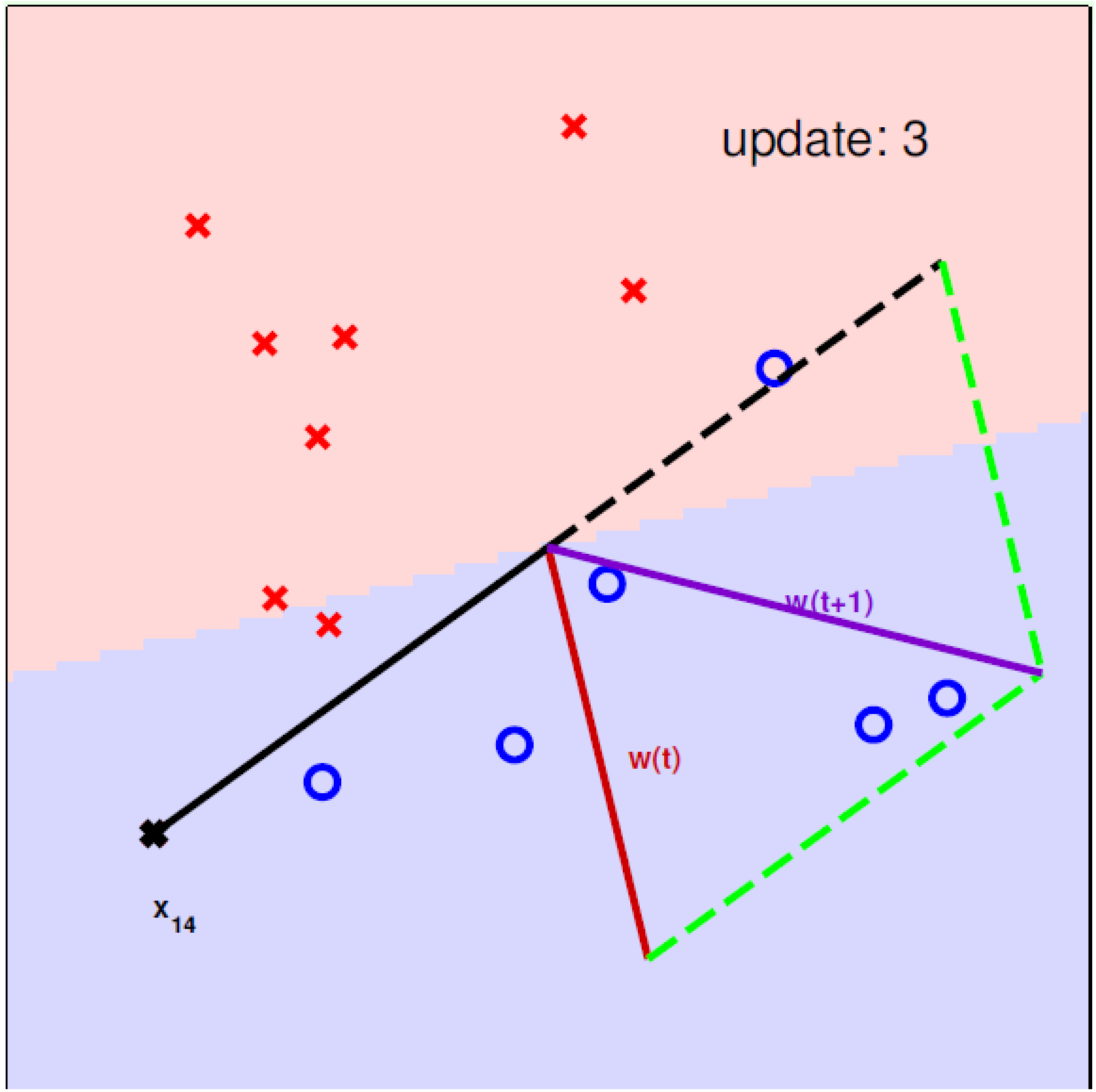
Perceptron Learning Algorithm



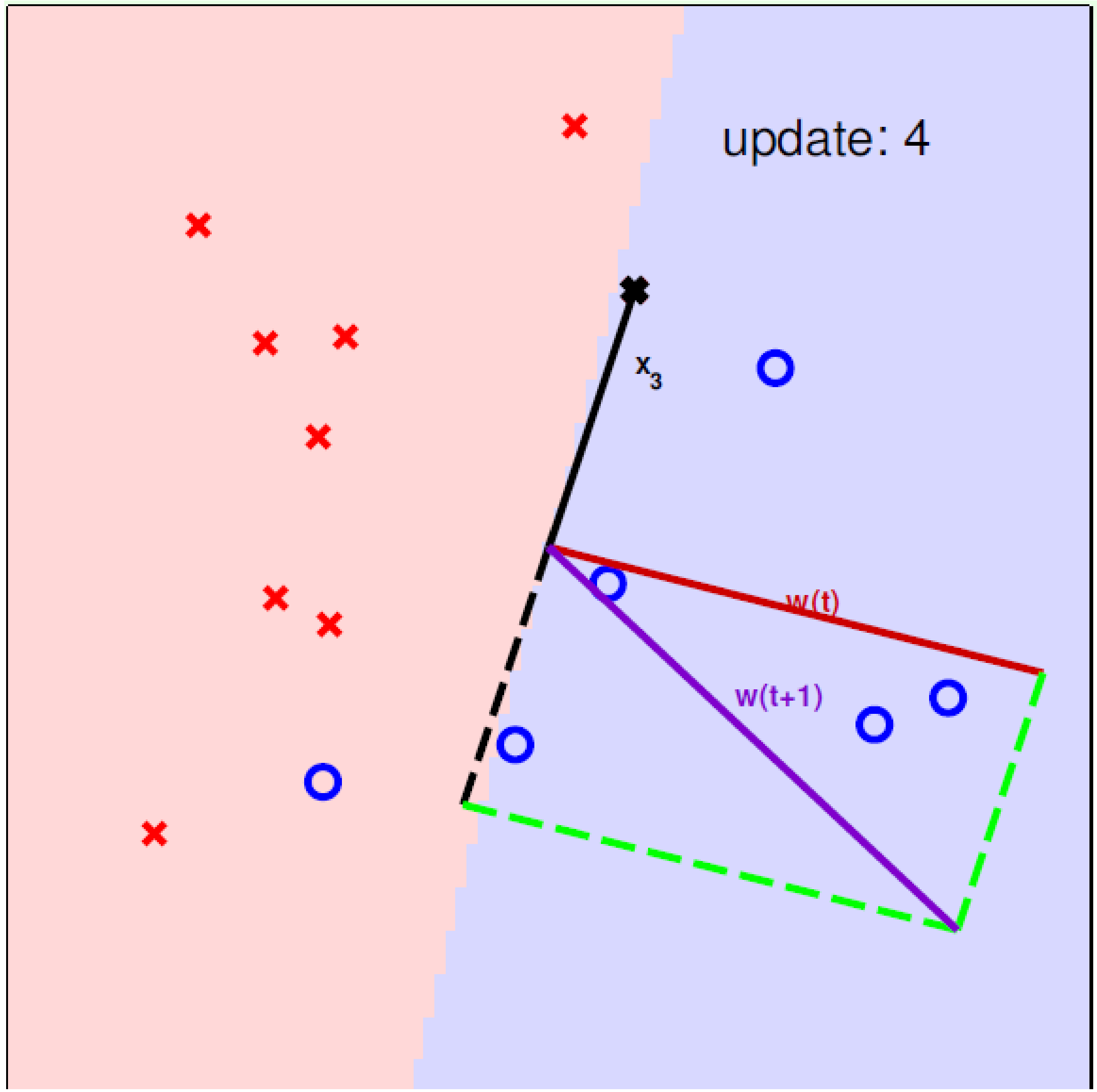
Perceptron Learning Algorithm



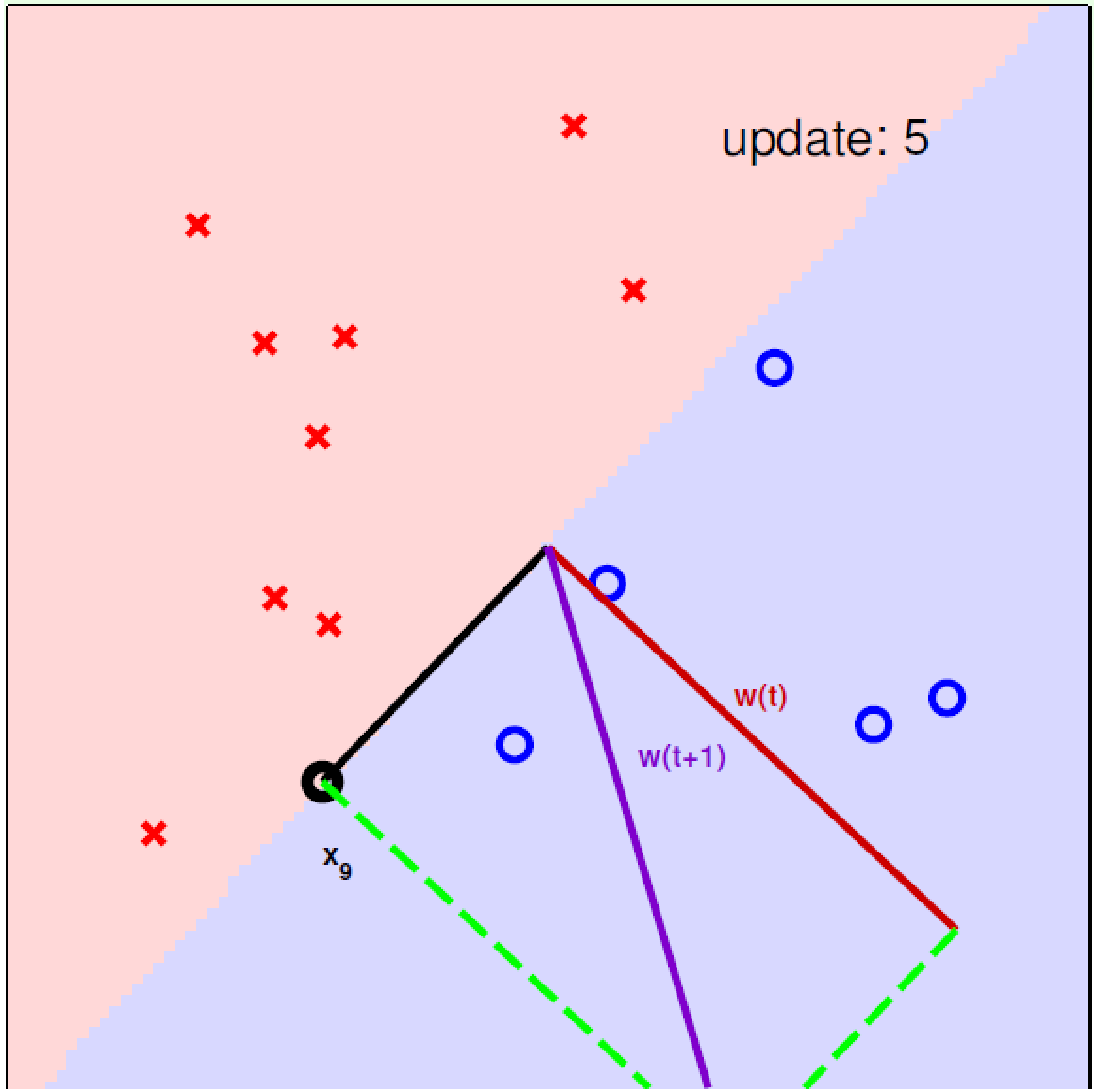
Perceptron Learning Algorithm



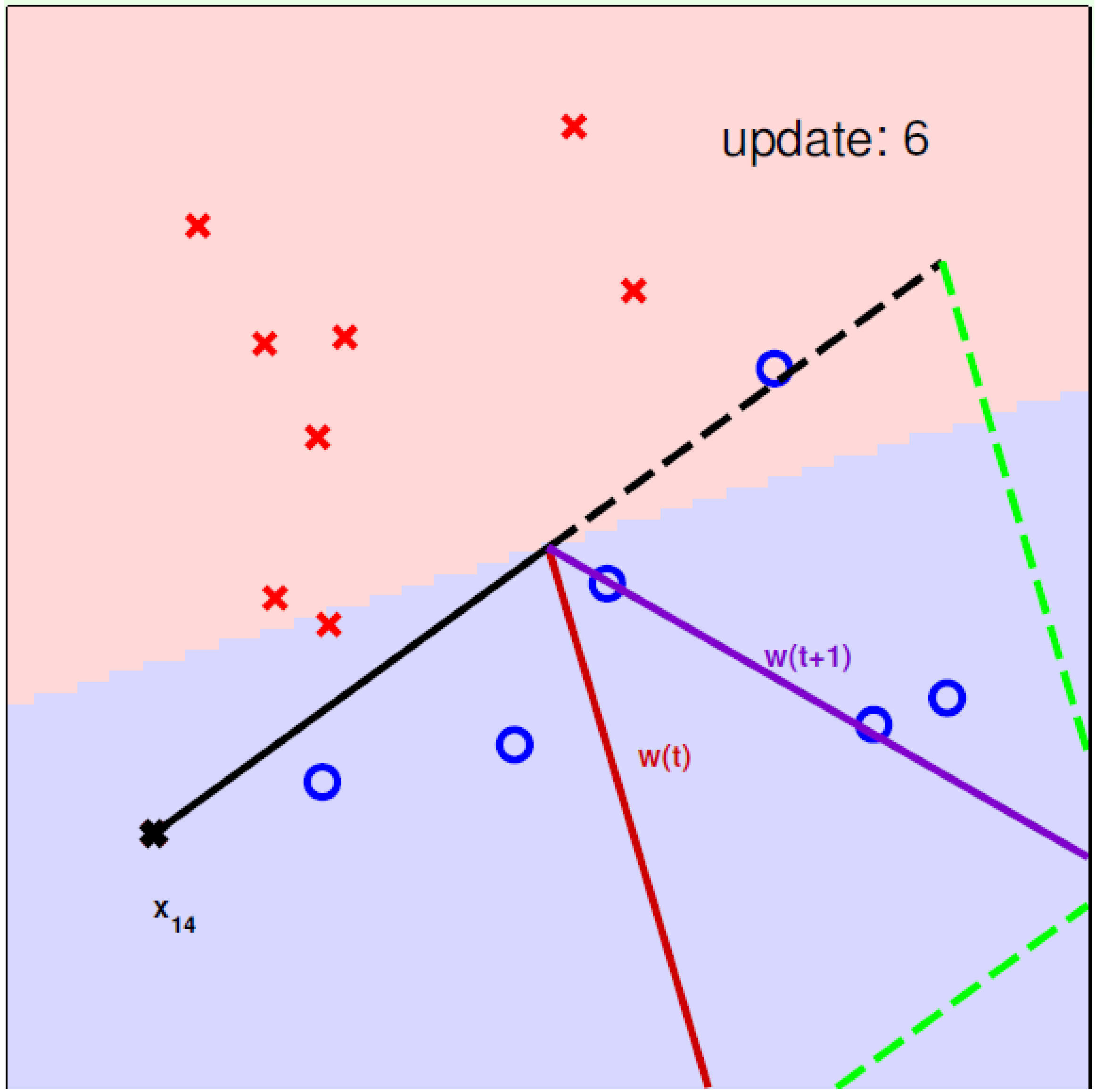
Perceptron Learning Algorithm



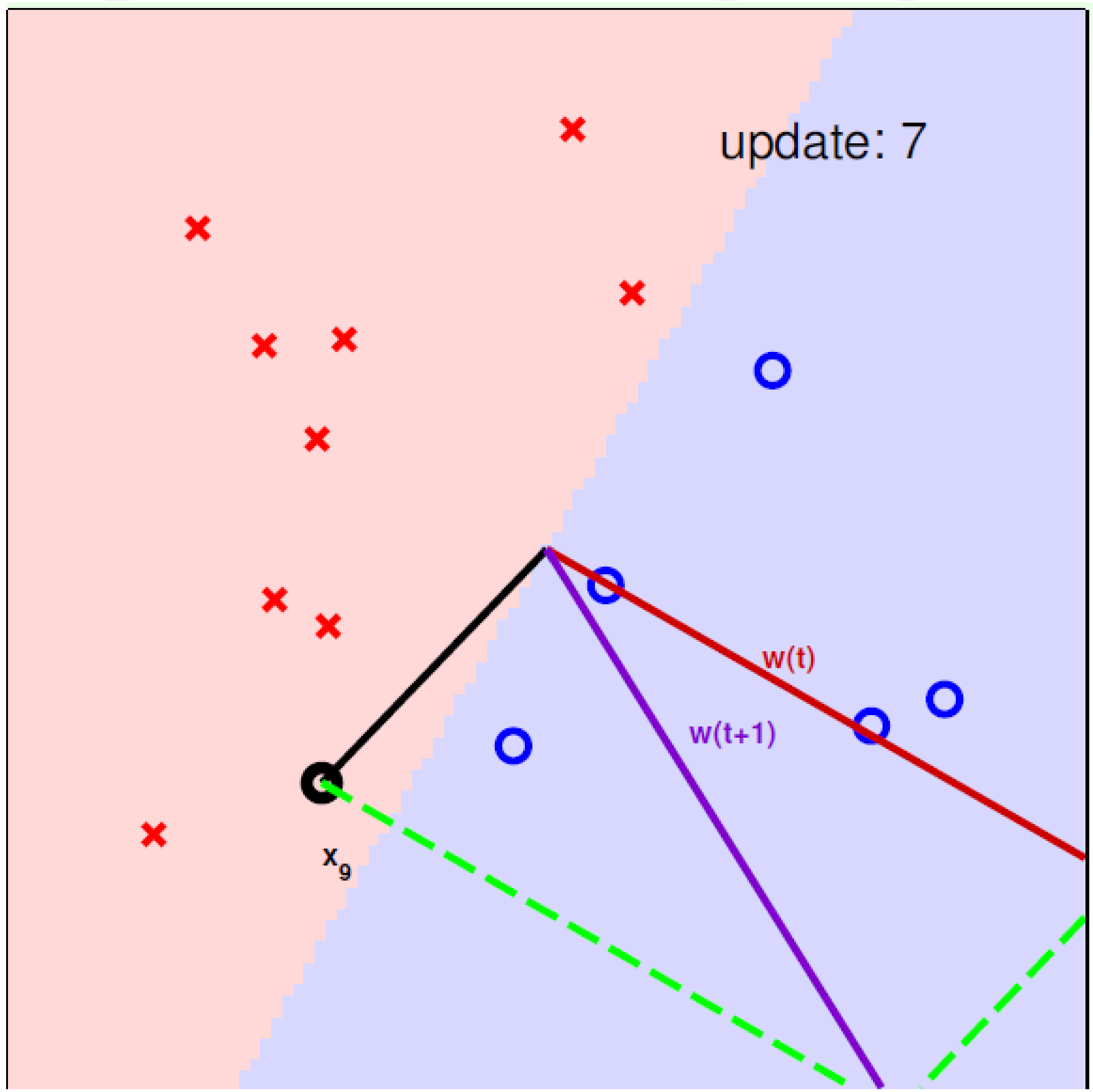
Perceptron Learning Algorithm



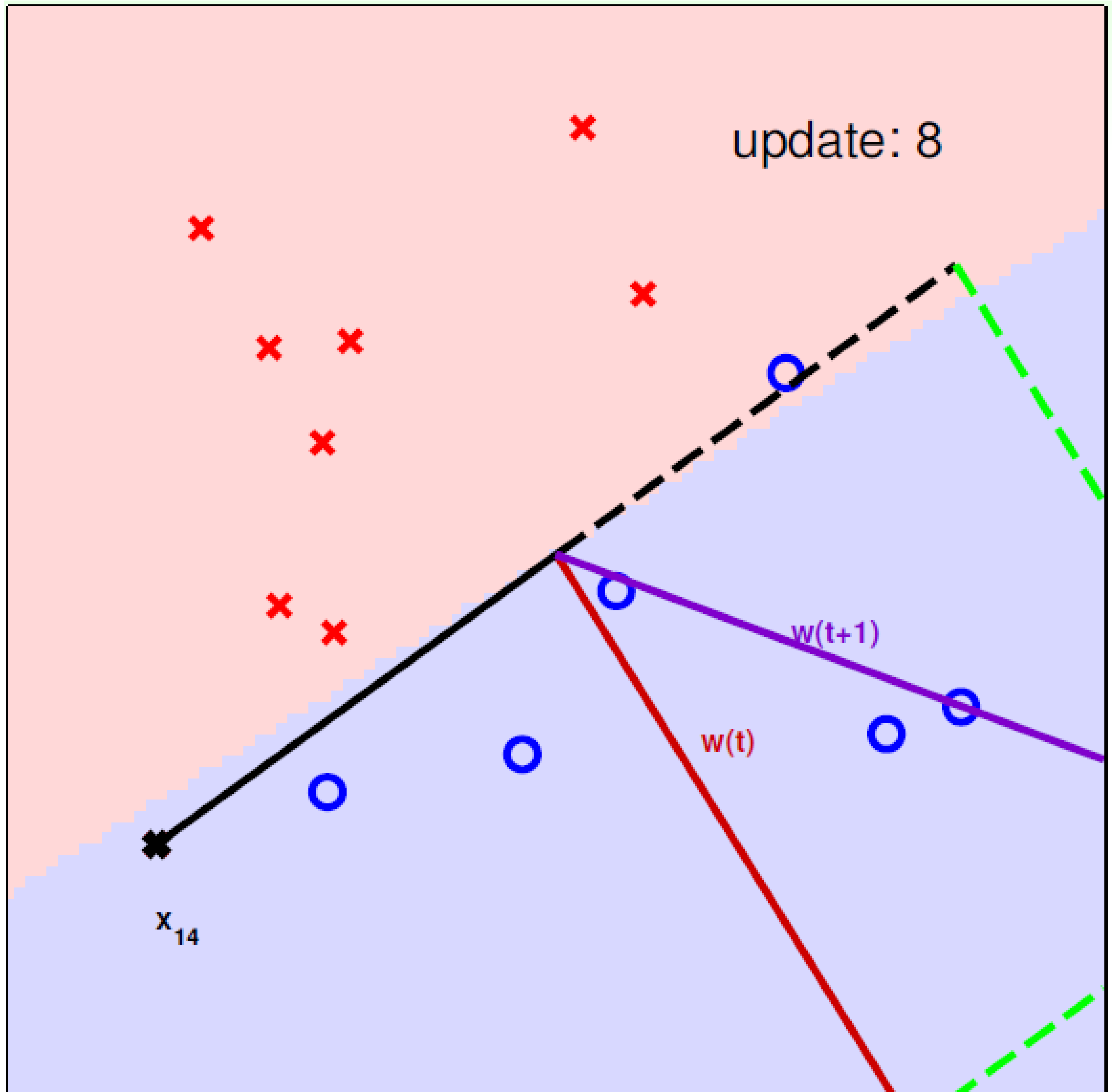
Perceptron Learning Algorithm



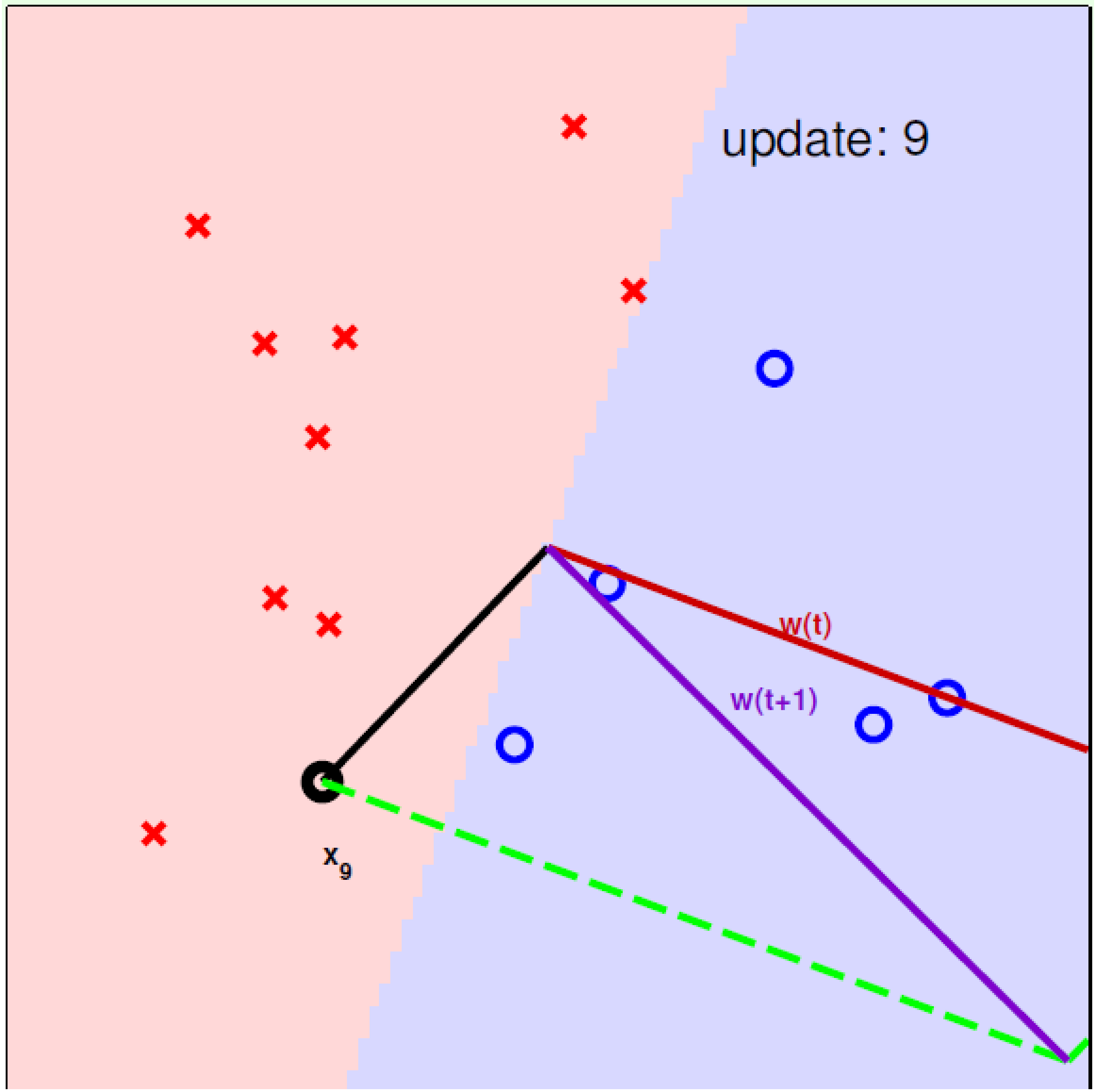
Perceptron Learning Algorithm



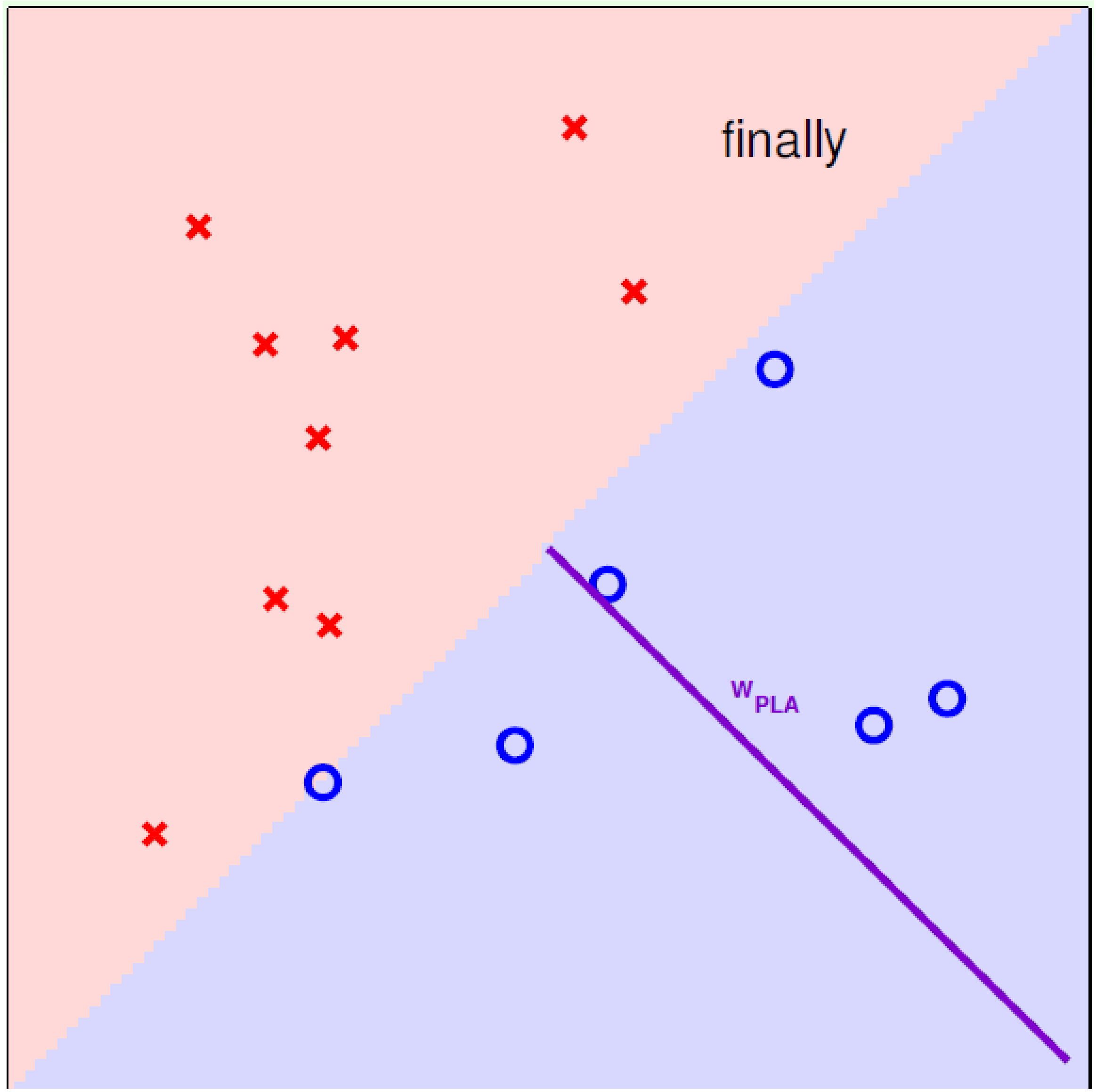
Perceptron Learning Algorithm



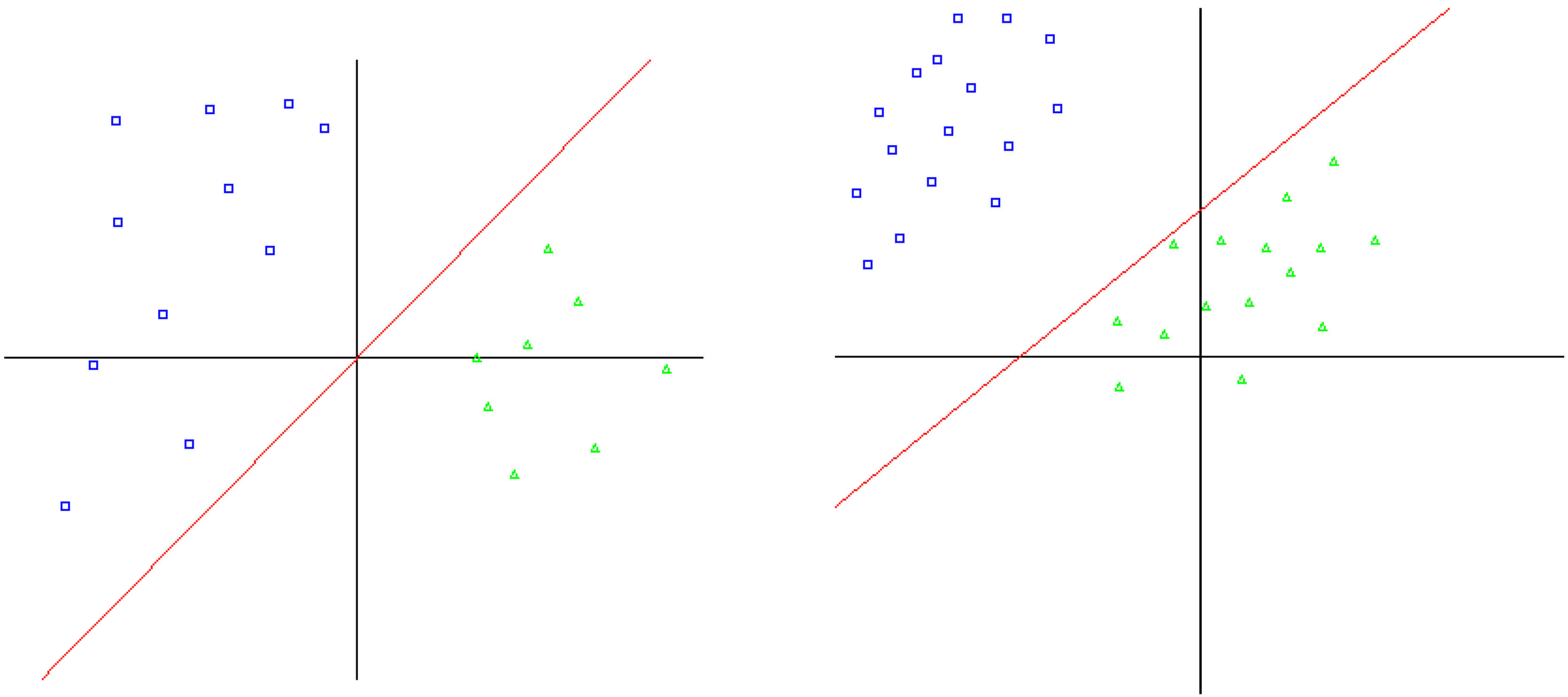
Perceptron Learning Algorithm



Perceptron Learning Algorithm



Perceptron Learning Algorithm



Perceptron Learning Algorithm

- Only if there exists an hyperplane that correctly classifies the data, the Perceptron procedure is guaranteed to converge; furthermore, the algorithm may give different results depending on the order in which the elements are processed, indeed several different solutions exist.

Perceptron Learning Algorithm