Artificial Intelligence & Machine Learning and Pattern Recognition ——Perceptron Learning Algorithm

Yanghui Rao
Assistant Prof., Ph.D
School of Mobile Information Engineering,
Sun Yat-sen University
raoyangh@mail.sysu.edu.cn

Decision Trees References

- 两个连续变量的线性相关度，用协方差或相关系数来衡量
  - 非线性相关度：Maximal Information Coefficient (MIC). Detecting novel associations in large data sets, *Science*, 2011.
- 两个离散变量的相关度，用互信息度量
  - 互信息倾向于选择取值更多的离散型属性：Adjusted Mutual Information (AMI). Information theoretic measures for clusterings comparison: is a correction for chance necessary? *ICML*, 2009.
  - Standardized mutual information for clustering comparisons: one step further in adjustment for chance, *ICML*, 2014.
  - Entropy evaluation based on confidence intervals of frequency estimates: application to the learning of decision trees, *ICML*, 2015

Regression Review

- Least-squares solutions

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

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

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

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

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

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

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

Regression Review

- 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 $\sum_{i=1}^{d} w_i x_i > threshold$

  predict -1(bad) if $\sum_{i=1}^{d} w_i x_i < threshold$
- $\mathbf{y}=\{+1(\text{good}), -1(\text{bad})\}$

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$

Perceptron Learning Algorithm

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$

$$= sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + \underbrace{(-threshold)}_{w_0} \cdot \underbrace{(+1)}_{x_0}\right)$$

$$= sign\left(\sum_{i=0}^{d} w_i x_i\right)$$

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

Perceptron Learning Algorithm

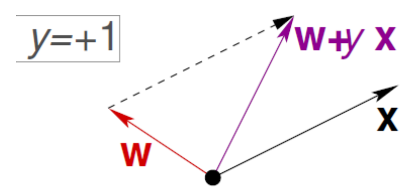
- Difficult: the set of $h(\mathbf{x})$ is of infinite size
- Idea: start from some initial weight vector $\mathbf{w}_{(0)}$, and "correct" its mistakes on D
- For t = 0, 1,…
  - find a mistake of $\mathbf{w}_{(t)}$ called $(\mathbf{x}_{n(t)}, y_{n(t)})$

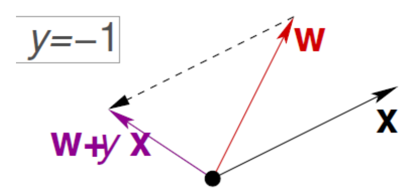
    $$sign\left(\mathbf{w}_{(t)}^{\mathrm{T}}\mathbf{x}_{n(t)}\right) \neq y_{n(t)}$$

  - (try to) correct the mistake by

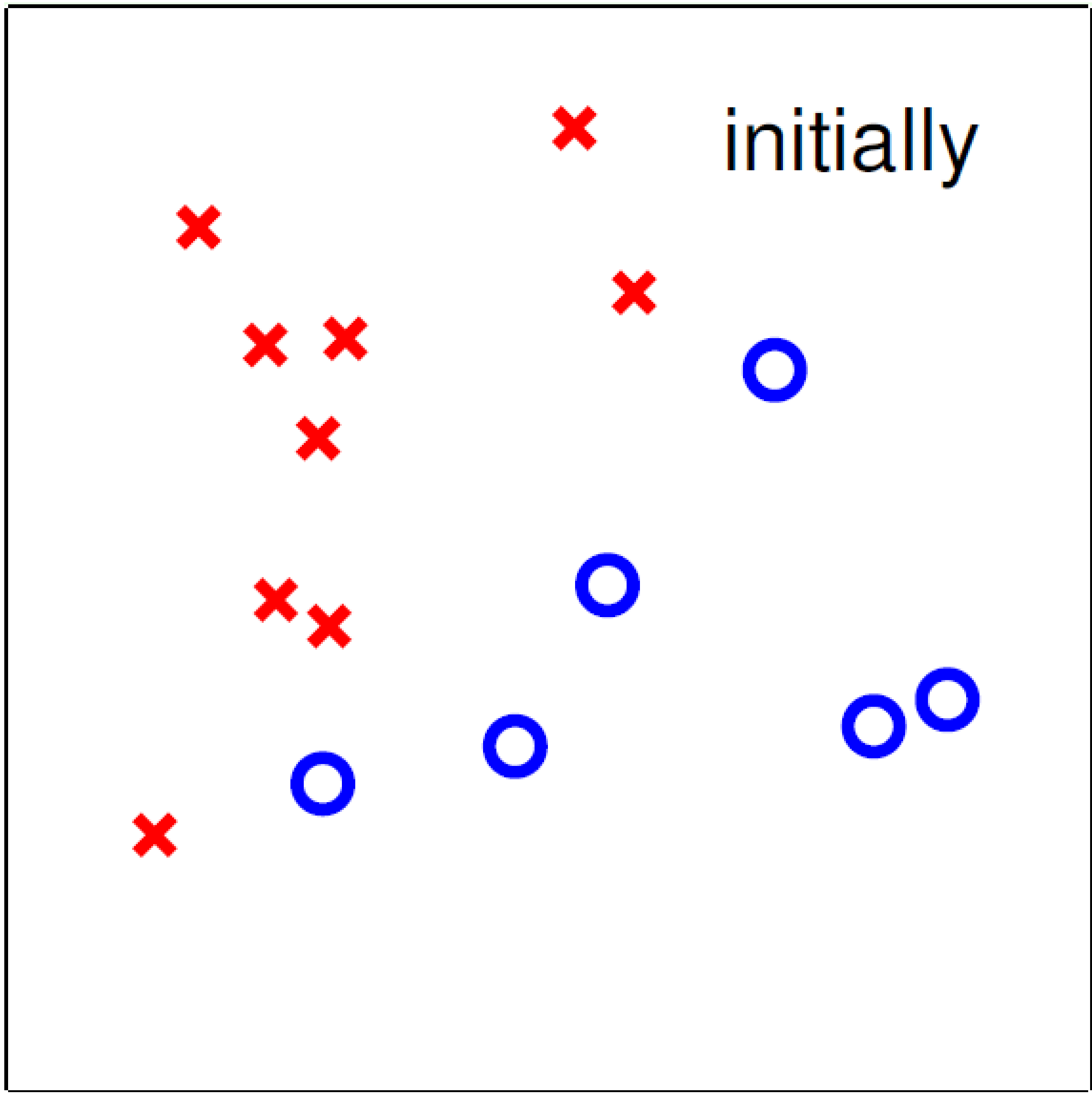
    $$\mathbf{w}_{(t+1)} \leftarrow \mathbf{w}_{(t)} + y_{n(t)}\mathbf{x}_{n(t)}$$

  - until no more mistakes
- Return last $\mathbf{W}$ (called $\mathbf{W}_{\mathrm{PLA}}$)

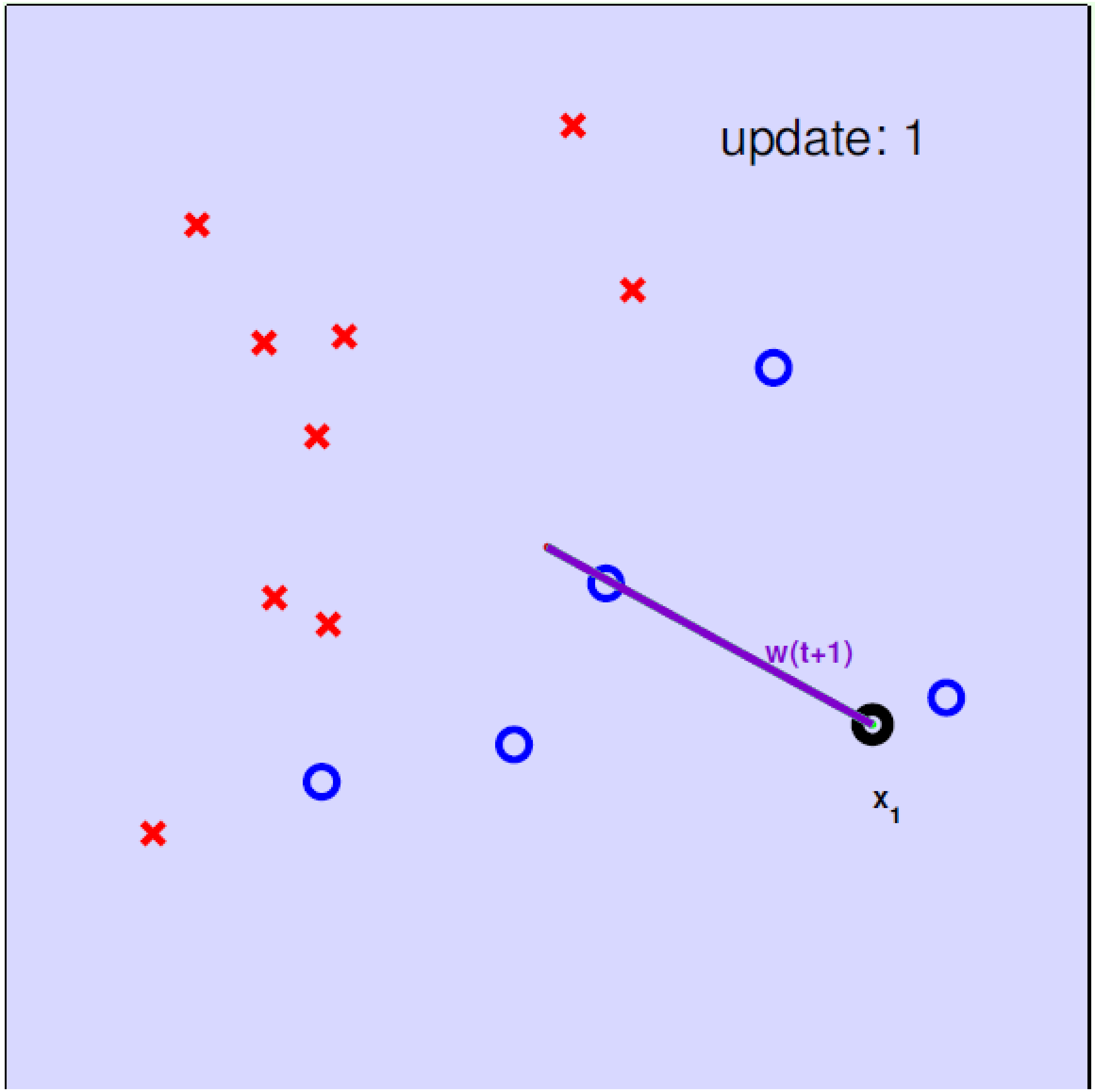




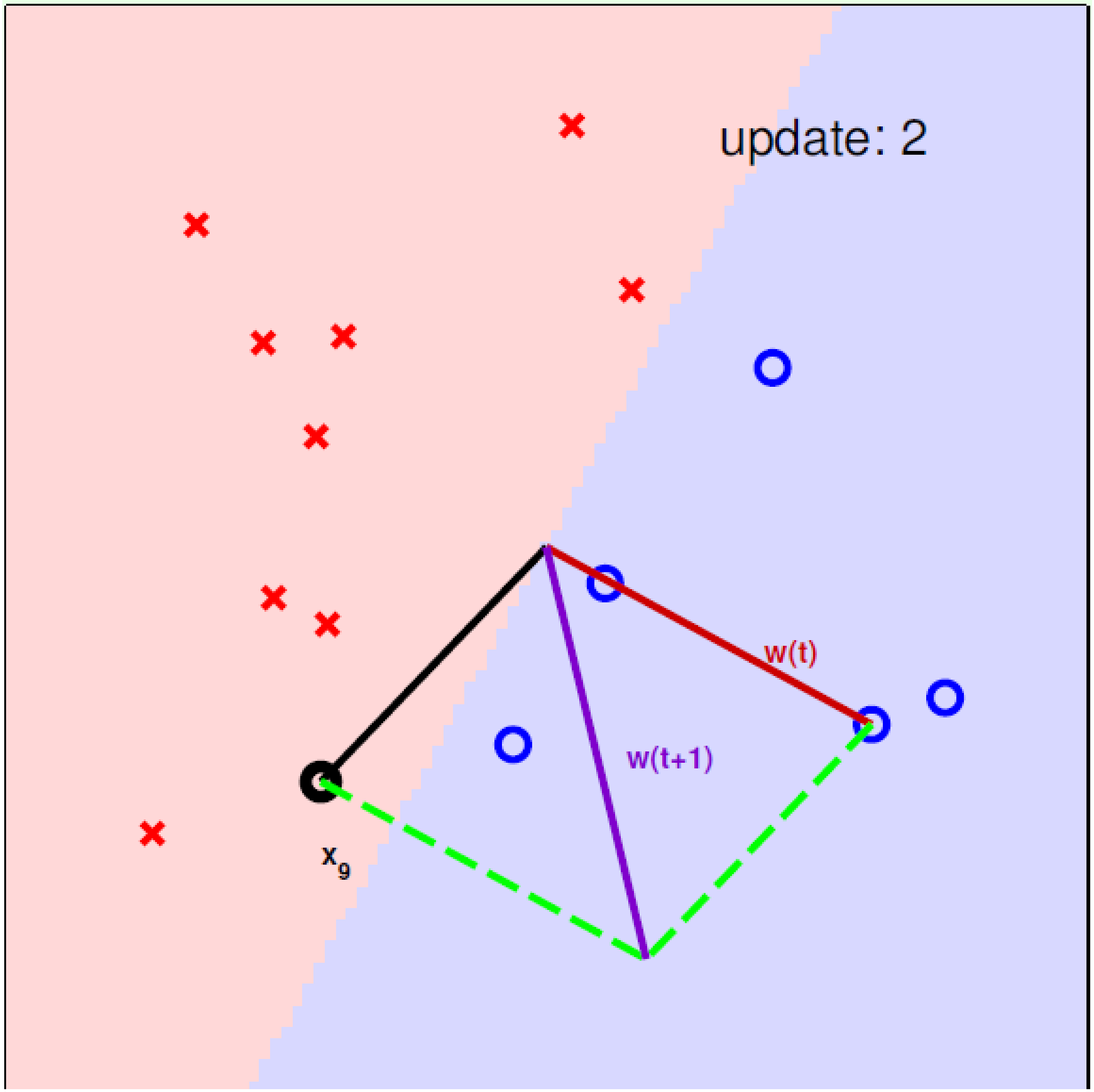
Perceptron Learning Algorithm



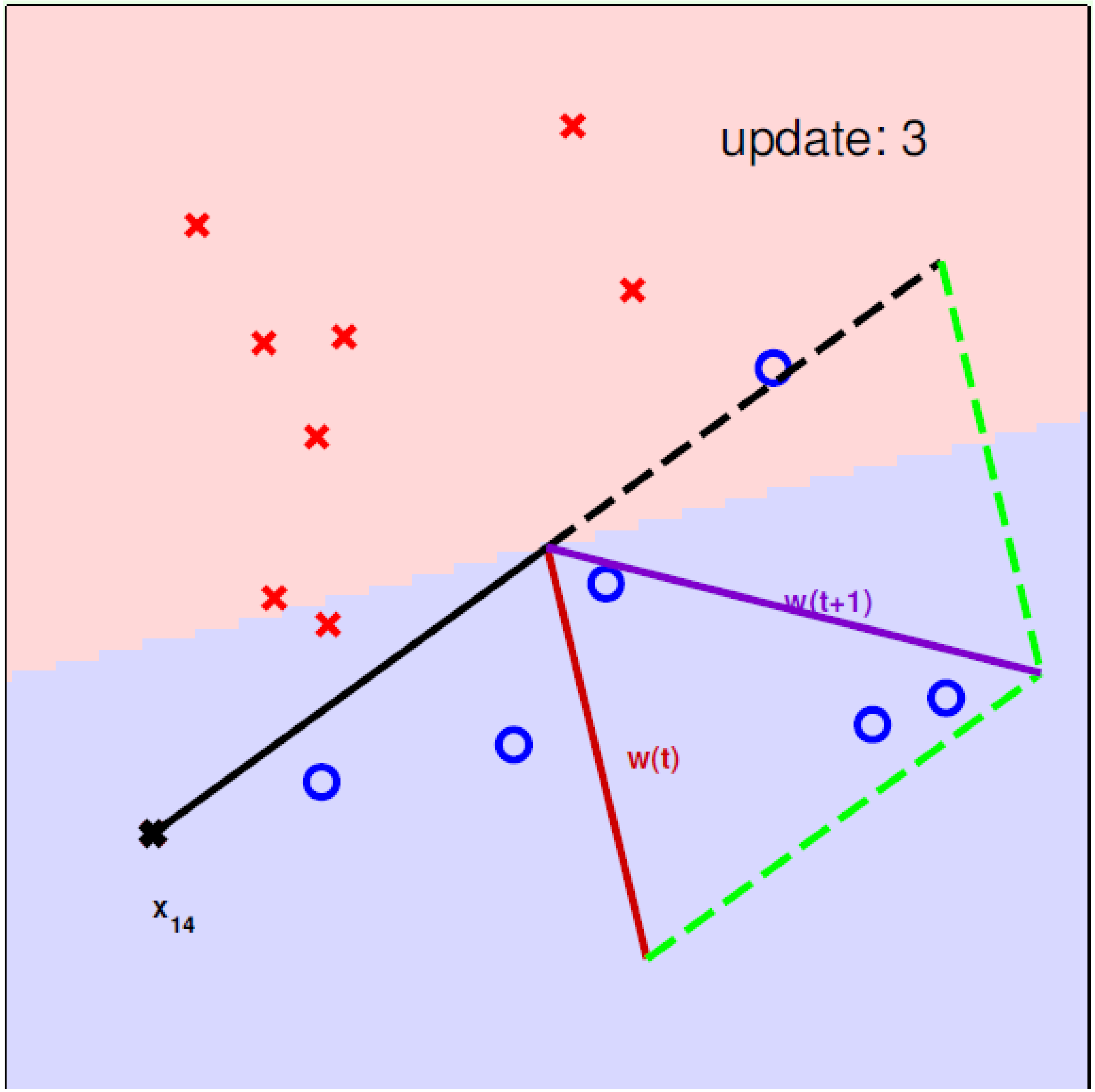
Perceptron Learning Algorithm



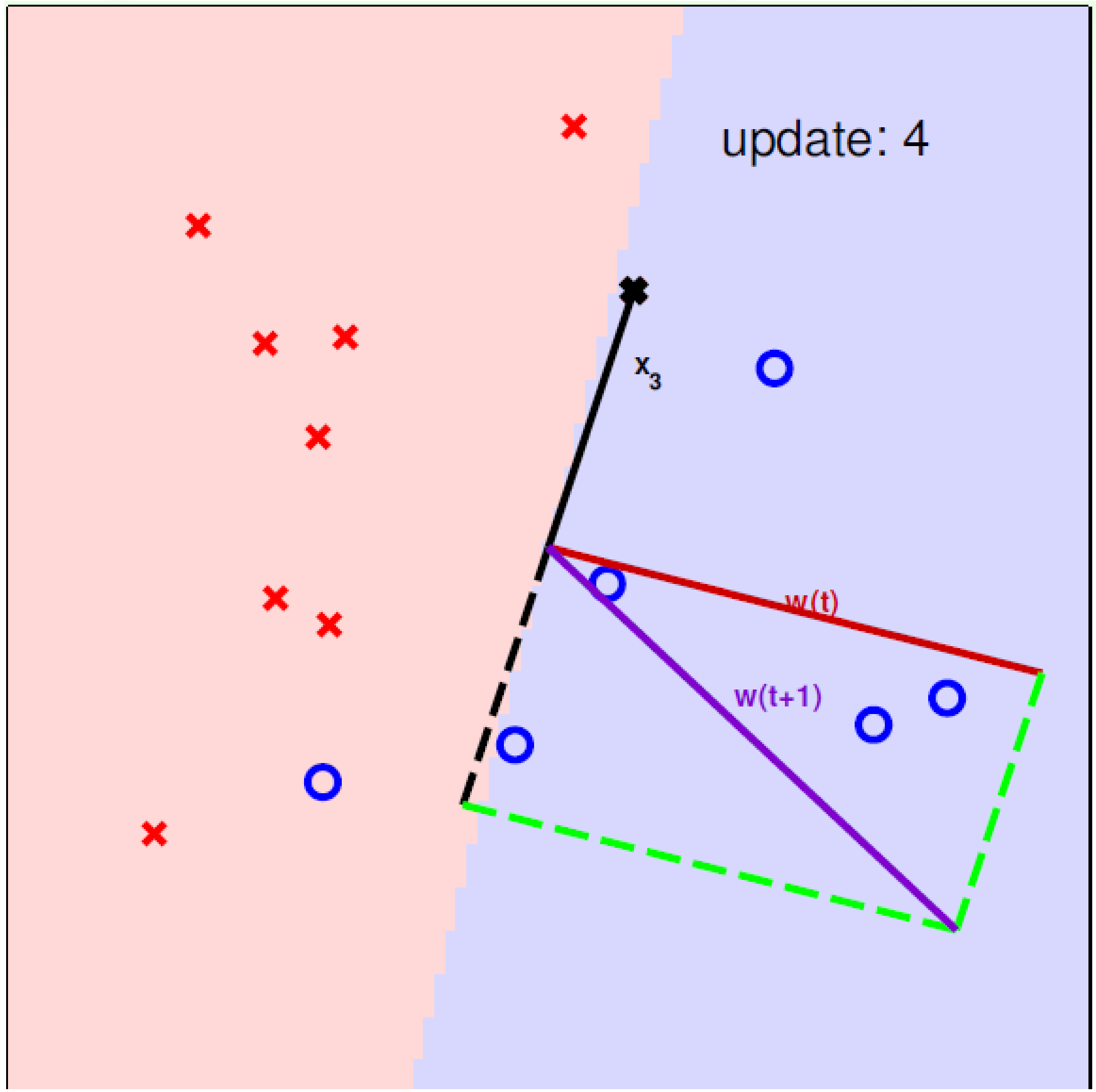
Perceptron Learning Algorithm



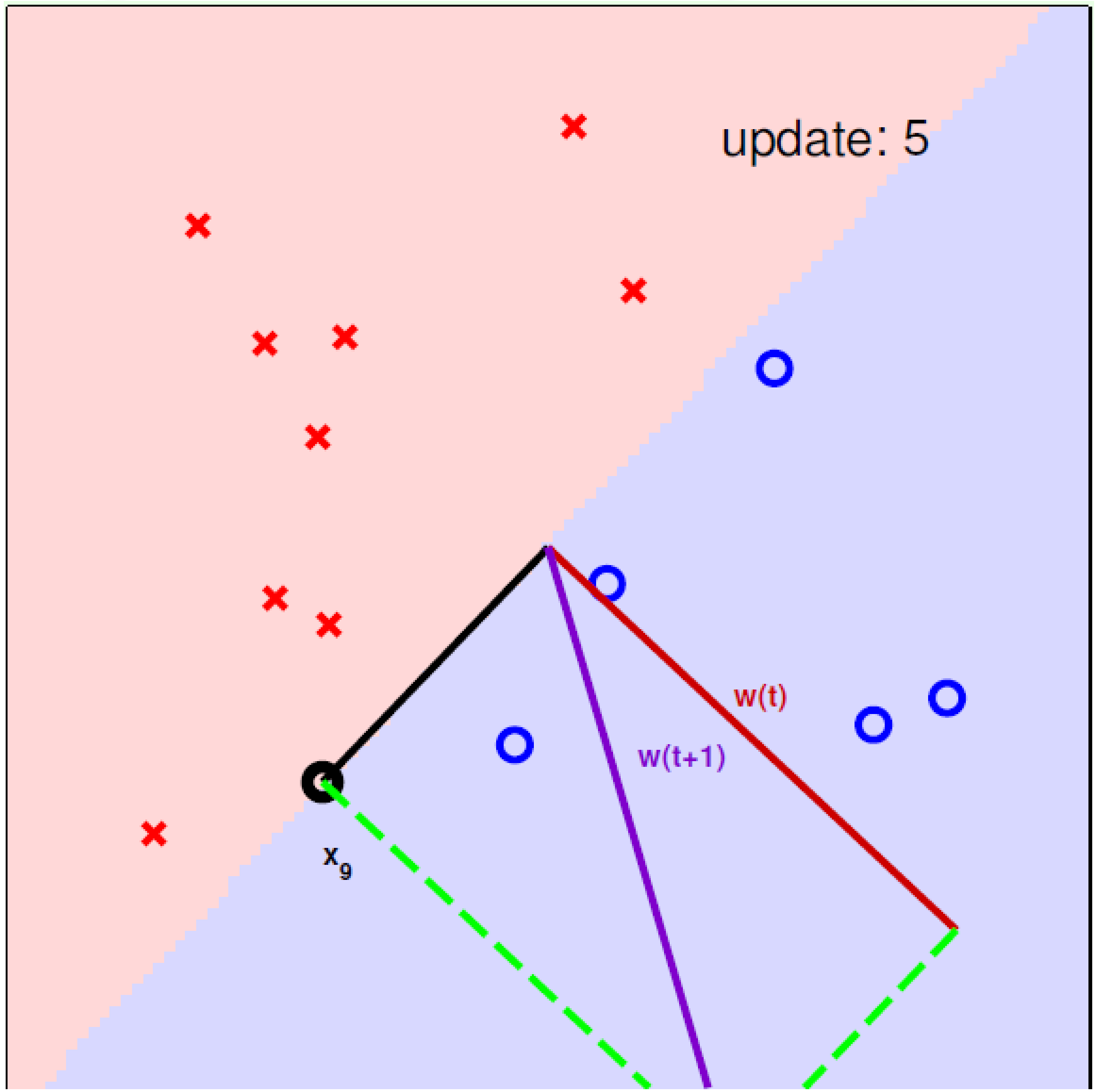
Perceptron Learning Algorithm



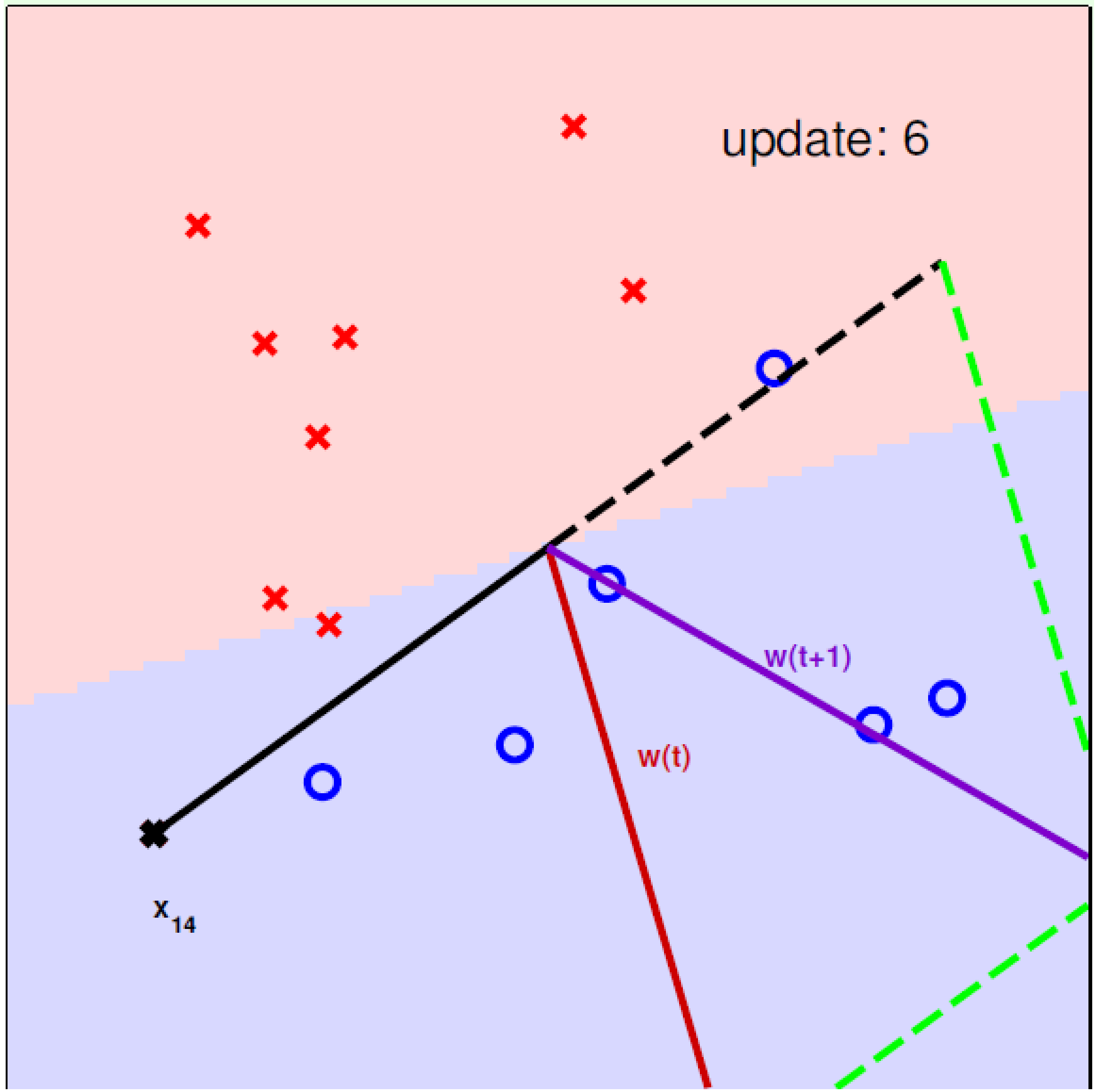
Perceptron Learning Algorithm



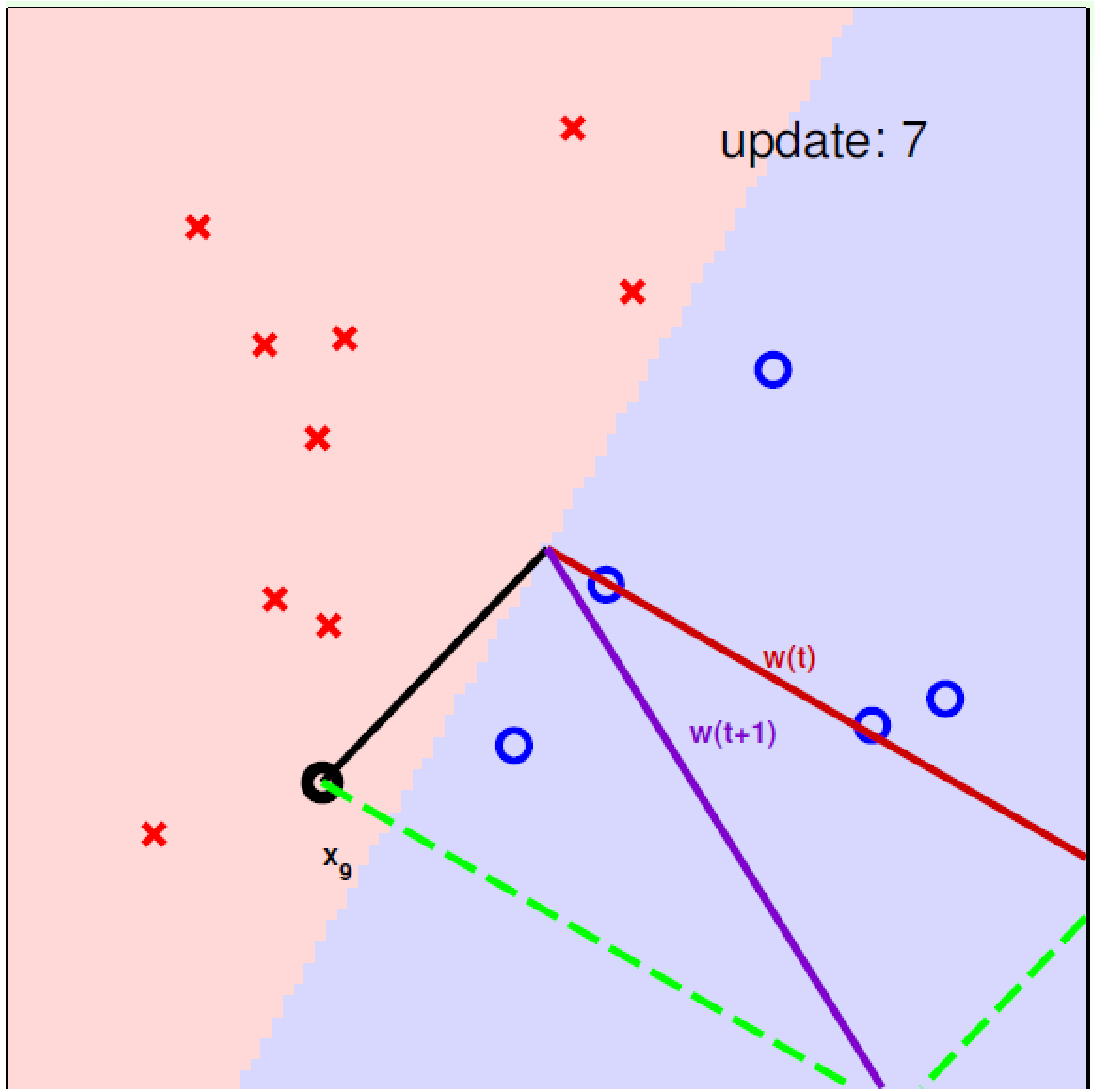
Perceptron Learning Algorithm



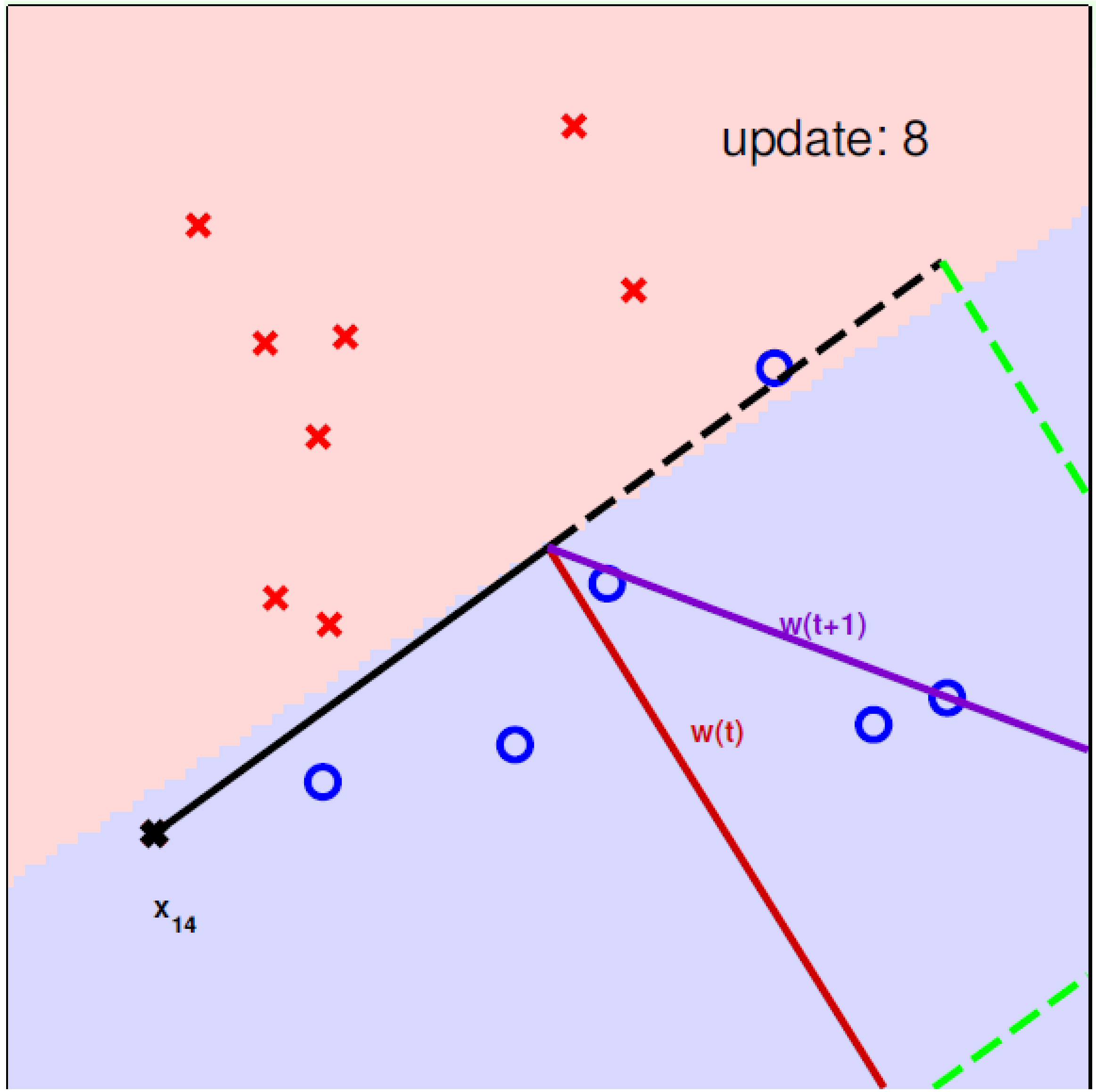
Perceptron Learning Algorithm



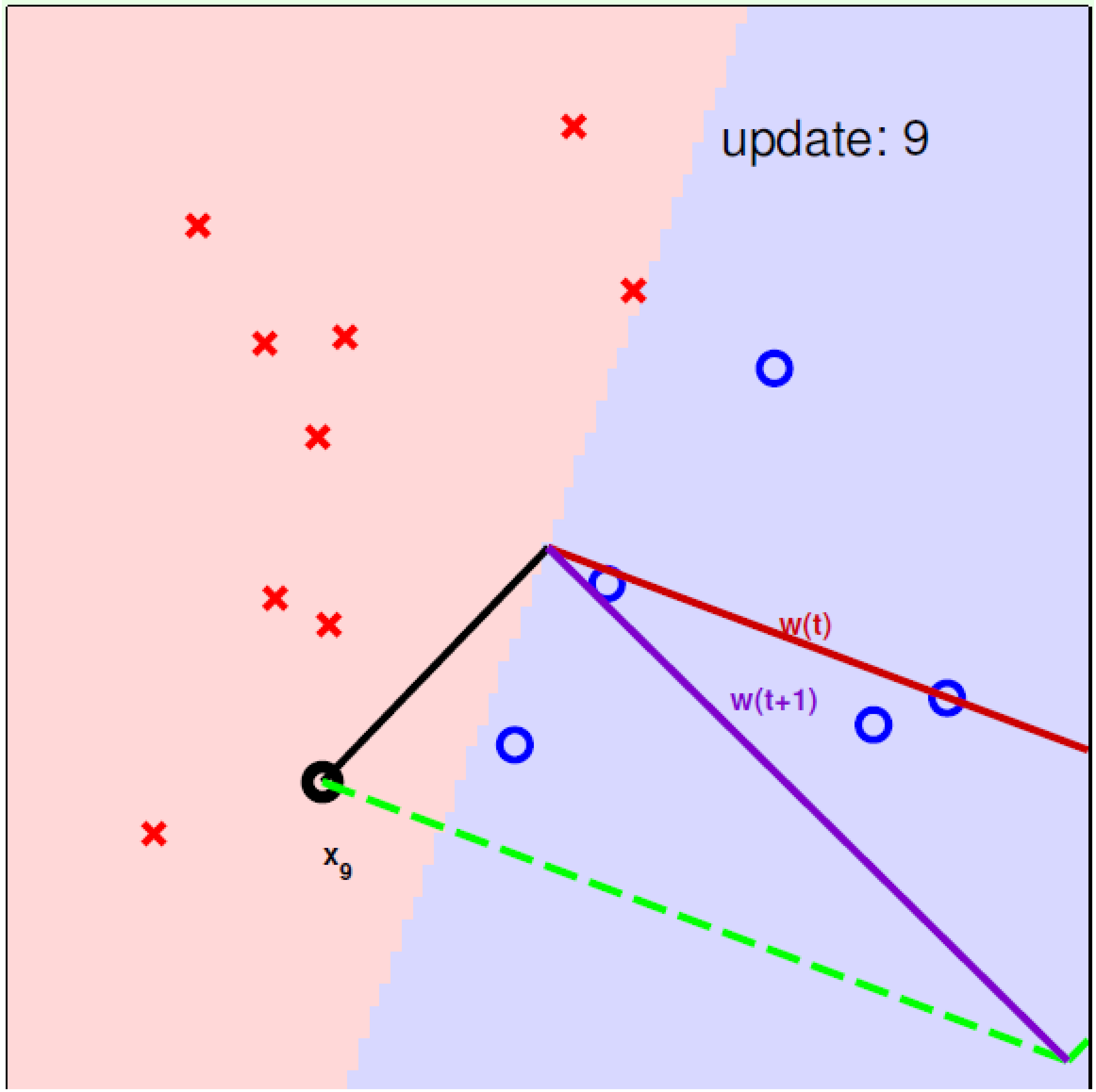
Perceptron Learning Algorithm



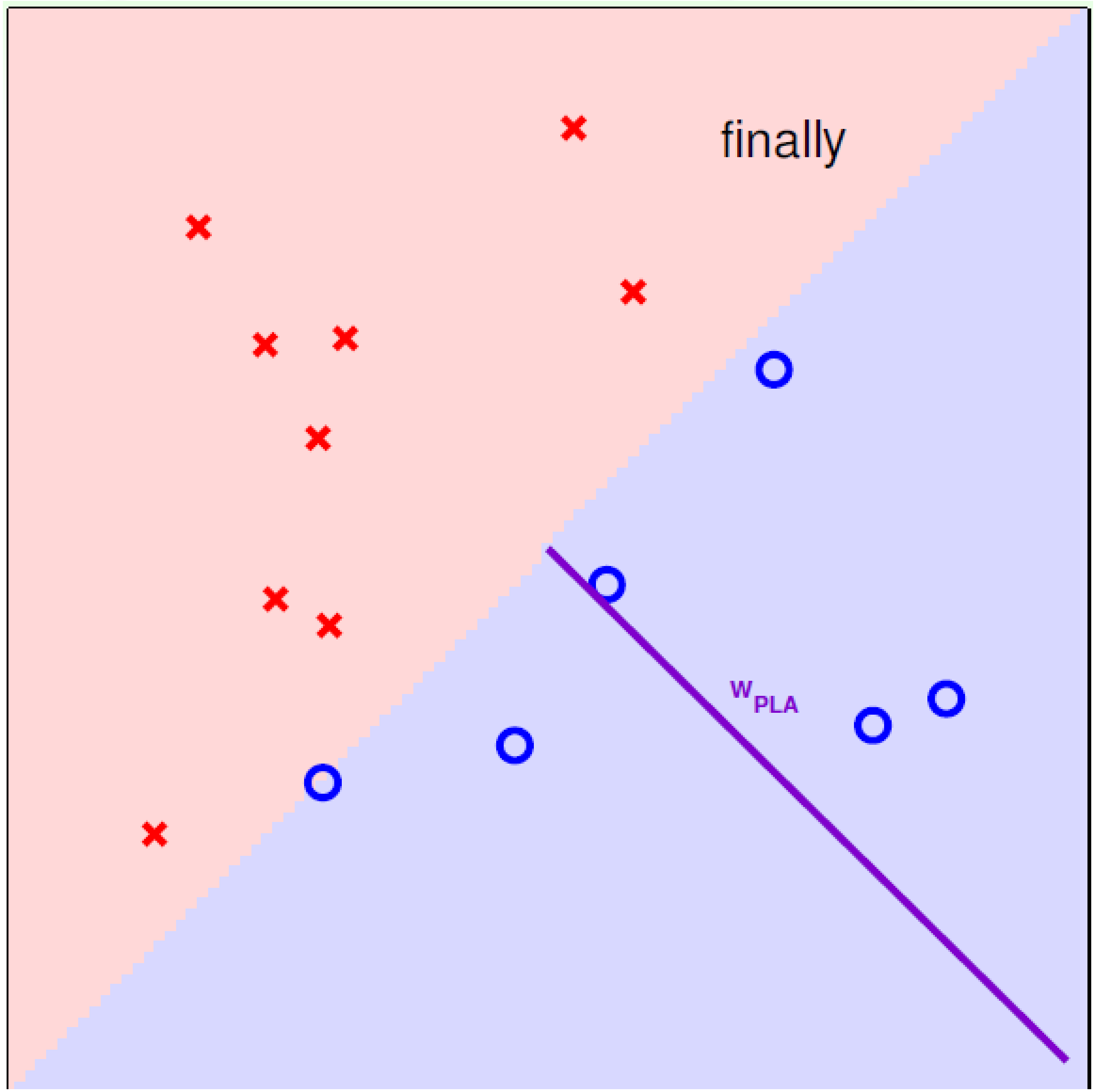
Perceptron Learning Algorithm



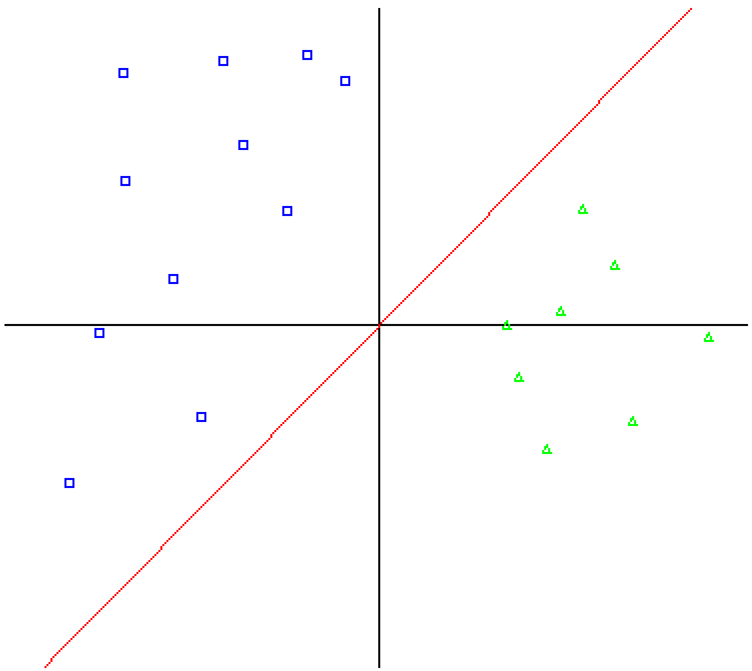
Perceptron Learning Algorithm

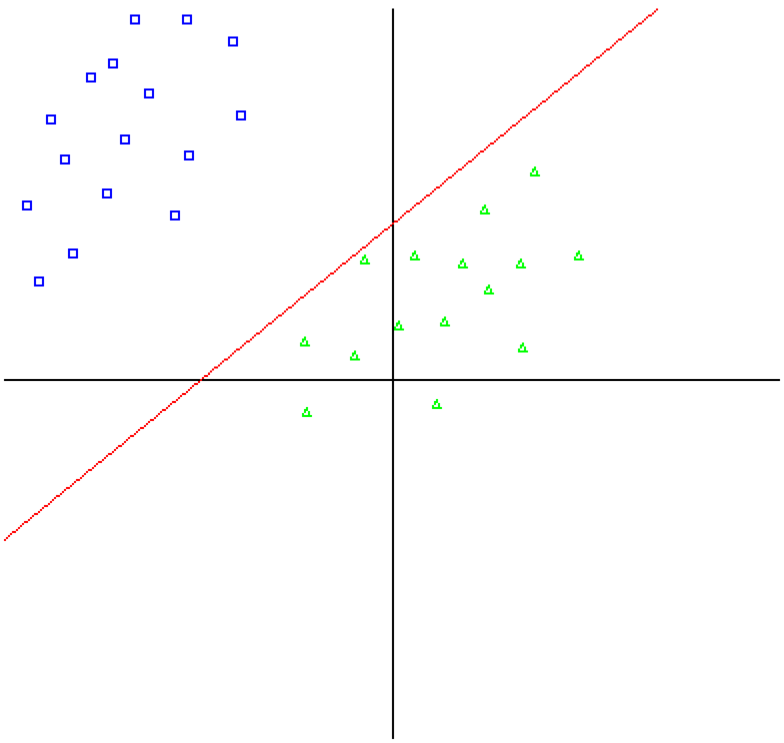


Perceptron Learning Algorithm



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