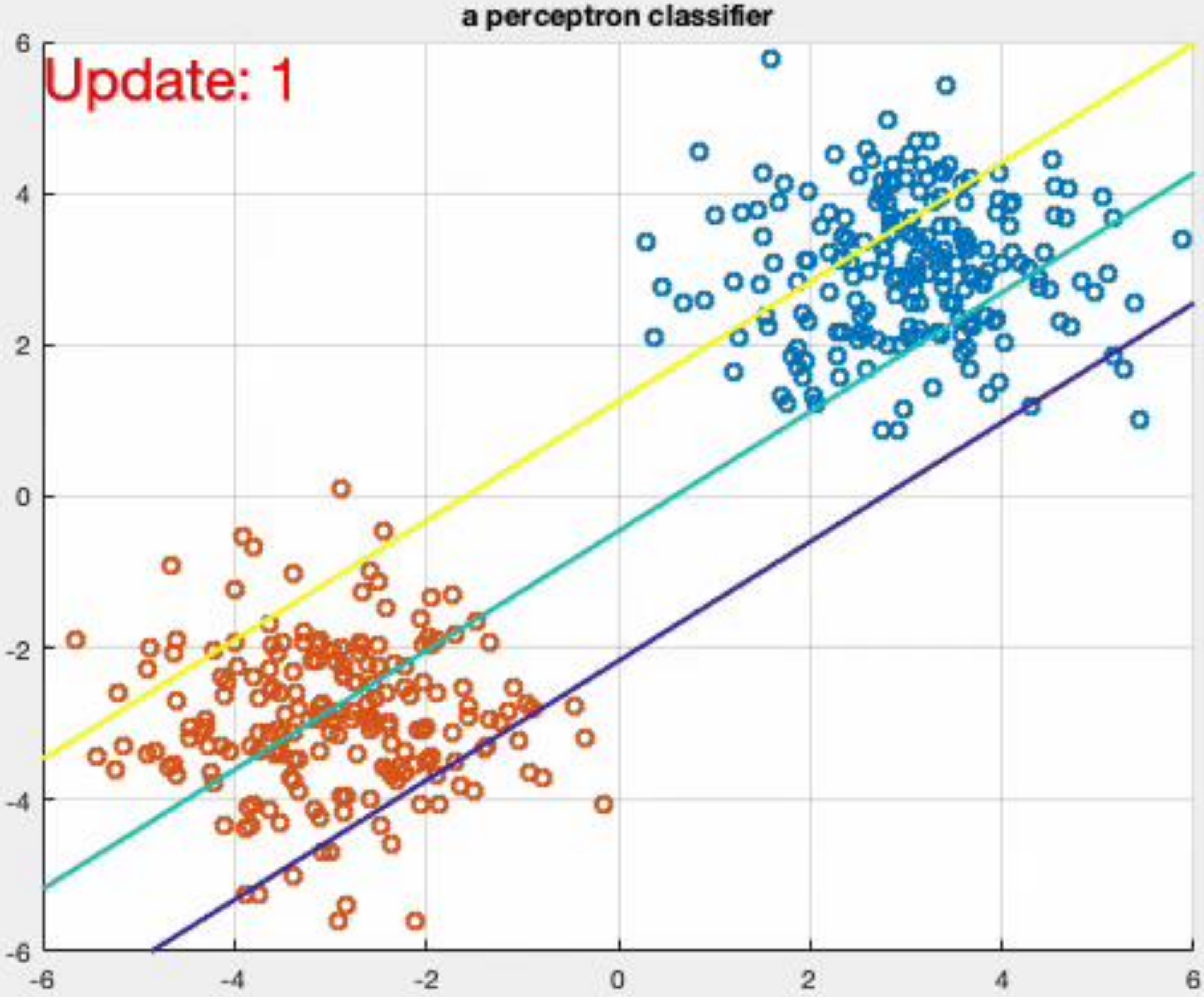
# Computing Lab

Song Liu (song.liu@bristol.ac.uk)


a perceptron classifier
Update: 1
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

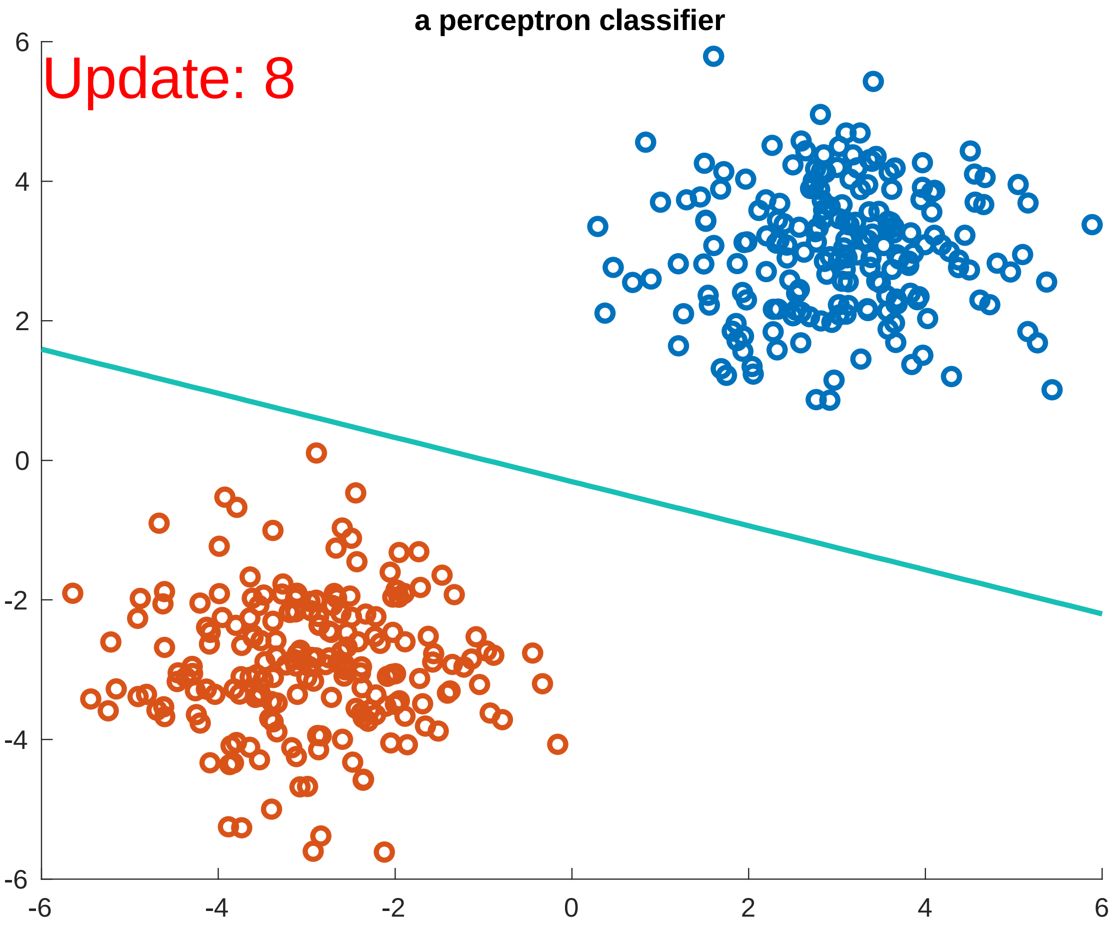
# From Perceptron to SVM

- We modify the perceptron classifier implemented last week by just a bit, to make it a proper SVM classifier!

- Recall our perceptron classifier:

```
• Initialize $\boldsymbol{w}$ by random
• For iter = 1 to max_iteration
    • Set step size $\eta = \frac{\eta_0}{\text{iter}}$
    • For $i \in D$
        • If $y_i \cdot f(\boldsymbol{x}_i; \boldsymbol{w}) \le 0$
        • $\boldsymbol{w}' = \boldsymbol{w} + \eta \cdot y_i \cdot \widetilde{\boldsymbol{x}}_i$   , where $\widetilde{\boldsymbol{x}} = [\boldsymbol{x}, 1]$
```

# Perceptron Does Not Care Margin



- This decision boundary has a thin margin!

# From Perceptron to SVM

- Perceptron does not like datapoints are on the wrong side of the decision boundary, a.k.a.,
- If $y_i \cdot f(\boldsymbol{x}_i; \boldsymbol{w}) \leq 0$, then modify $\boldsymbol{w}$.
- Otherwise don't care.

- SVM does not like datapoints are on the wrong side of the margin, a.k.a.,
- If $y_i \cdot f(\boldsymbol{x}_i; \boldsymbol{w}) \leq 1$, then modify $\boldsymbol{w}$.
- Otherwise don't care.

# Maximizing Margin

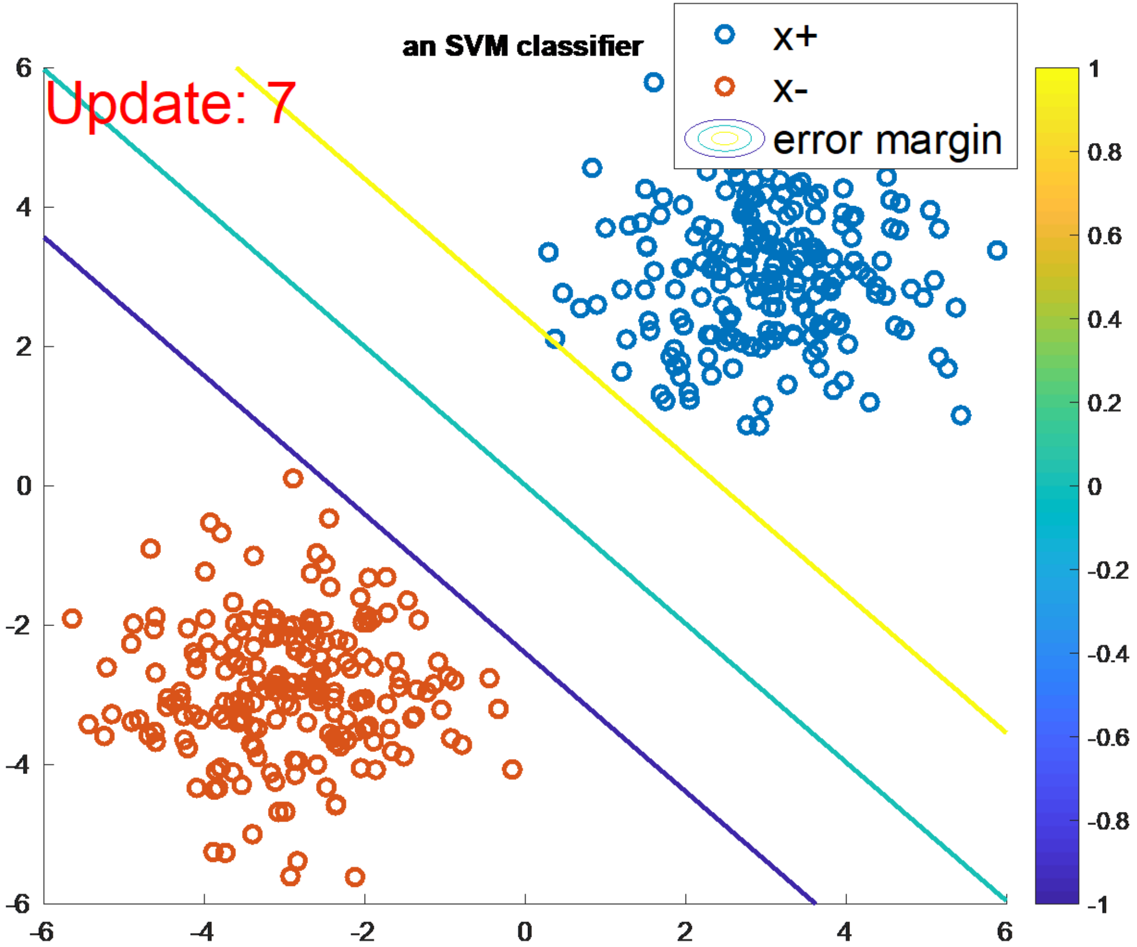
- SVM also does not like margin small.
- Each step, SVM increases the margin, by minimizing $||\boldsymbol{w}'||^2$.

- This can be done by iteratively setting
- $\boldsymbol{w}'_{\text{new}} := \boldsymbol{w}' - c \cdot \boldsymbol{w}'$**,** where $c < 1$ is a const.
- **Show** $||\boldsymbol{w}'_{\text{new}}||^2 \leq ||\boldsymbol{w}'||^2$

# SVM Implementation

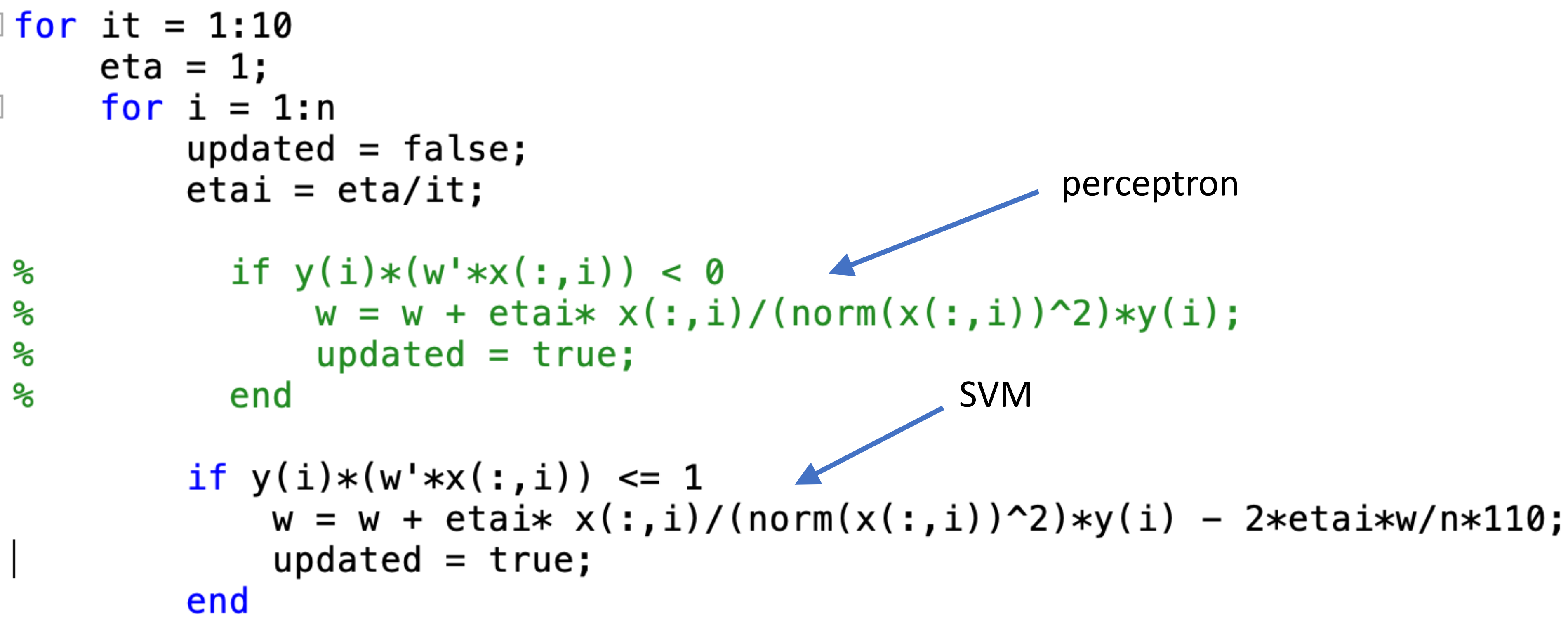
- Initialize $\boldsymbol{w}$ by random
- For iter = 1 to max_iteration
  - Set step size $\eta = \frac{\eta_0}{\text{iter}}$
  - For $i \in D$
    - If $y_i \cdot f(\boldsymbol{x}_i; \boldsymbol{w}) \leq 1$
    - $\boldsymbol{w}_{\text{new}}{}' = \boldsymbol{w}' + \eta \cdot (y_i \cdot \widetilde{\boldsymbol{x}}_i - c \cdot \boldsymbol{w}')$ ,
      - where $\widetilde{\boldsymbol{x}} = [\boldsymbol{x}, 1]$
    - $w_{0,\text{new}} = w_0 + \eta \cdot (y_i)$

# Toy Example



- $c = \ 0.55$

# MATLAB Code



```
for it = 1:10
    eta = 1;
    for i = 1:n
        updated = false;
        etai = eta/it;

%         if y(i)*(w'*x(:,i)) < 0
%             w = w + etai* x(:,i)/(norm(x(:,i))^2)*y(i);
%             updated = true;
%         end

        if y(i)*(w'*x(:,i)) <= 1
            w = w + etai* x(:,i)/(norm(x(:,i))^2)*y(i) - 2*etai*w/n*110;
            updated = true;
        end
```

perceptron

SVM

- Try different *c* and see what happens!