# A.I. in health informatics
# **lecture 7** classification algorithms

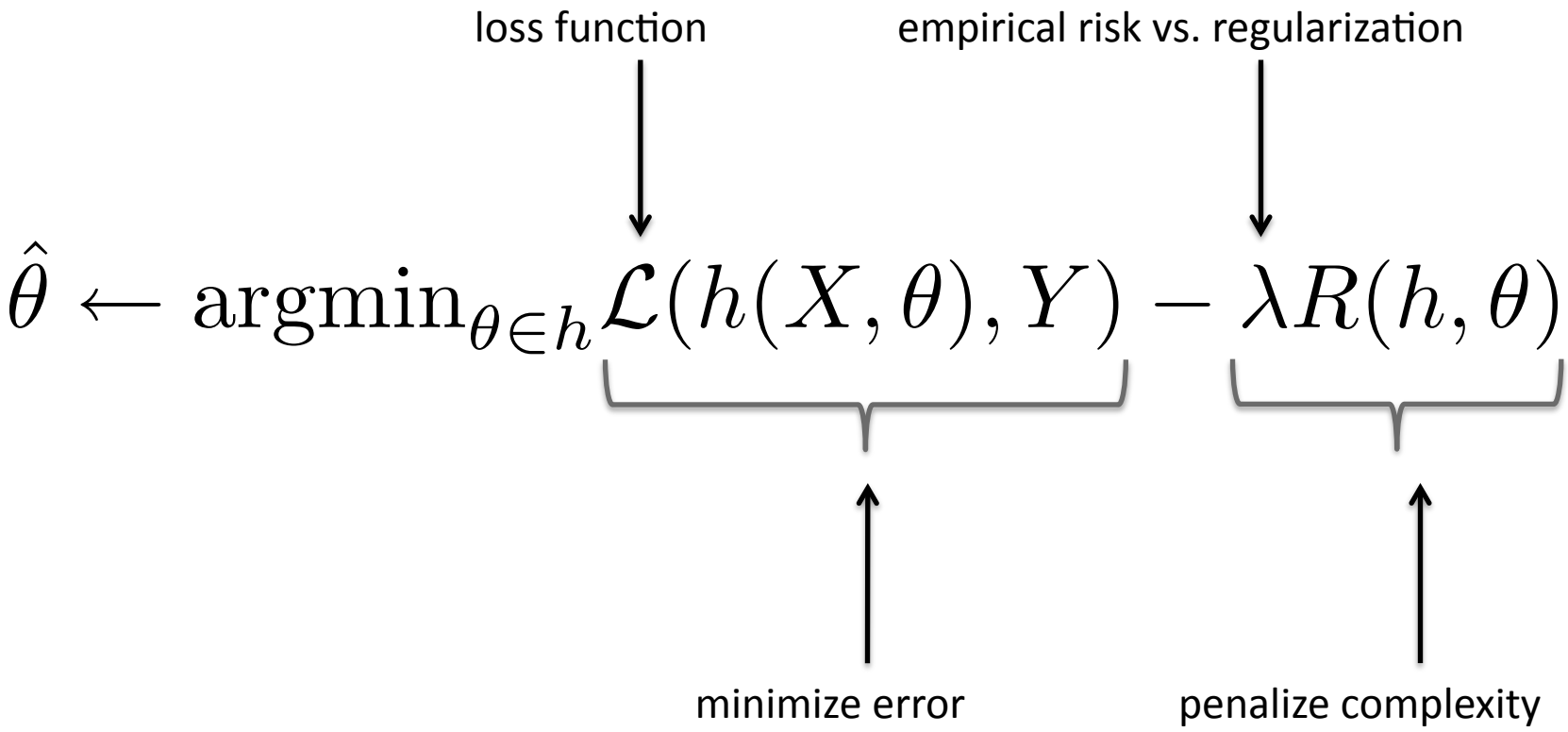
kevin small &
byron wallace

# today

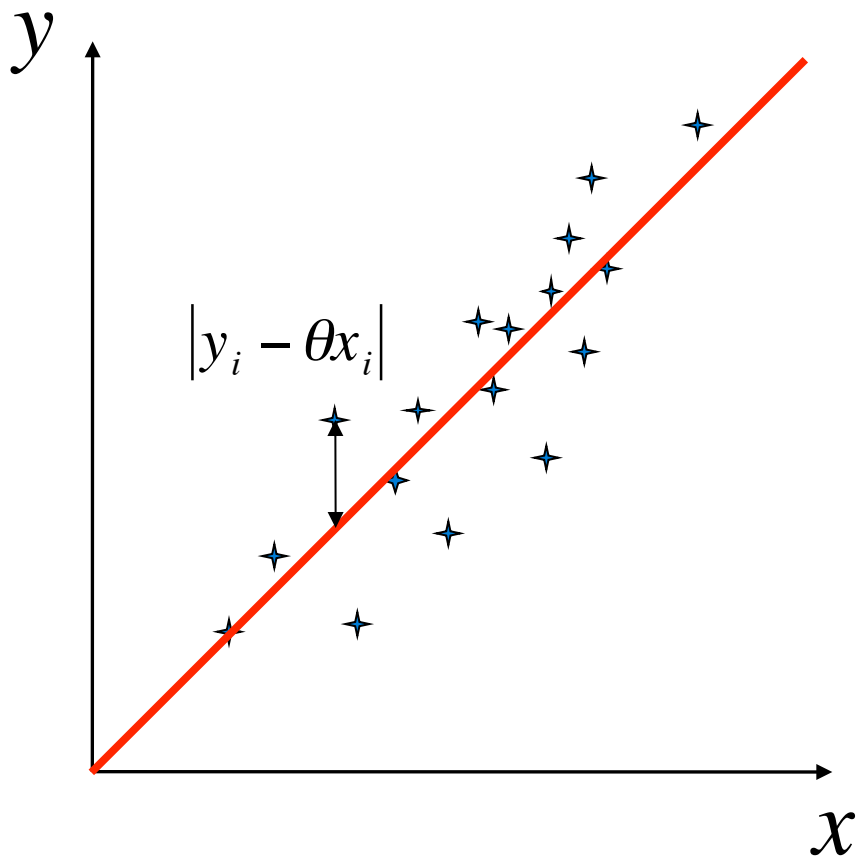
- linear classifiers
  - perceptron
  - support vector machines (SVM)

- decision trees

# supervised learning



# regression

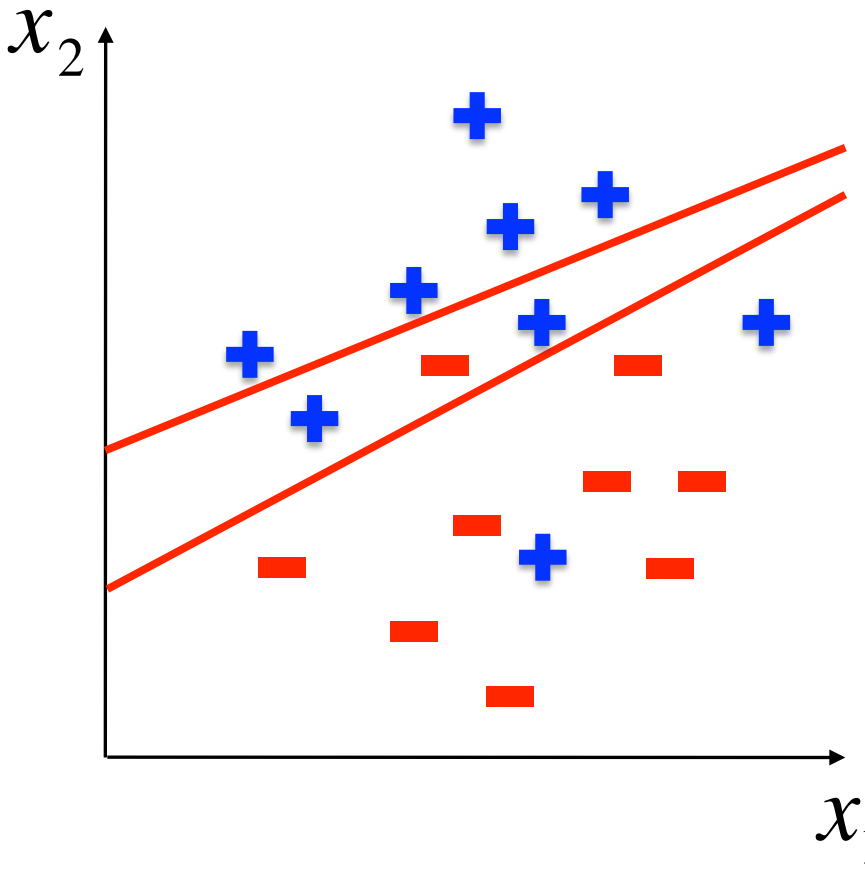



$$\mathcal{L}_{se}(h(X,\theta),Y) = \sum_{(x,y)\in\mathcal{S}} (y - \theta x)^2$$

# classification

- regression
  - given input variables, we want to predict the magnitude of output (e.g., BMI)
  - $\mathcal{Y} \in \mathbb{R}$
- (binary) classification
  - given input variables, we want to predict class membership (e.g., diabetic)
  - $\mathcal{Y} \in \{-1, 1\}$

# classification

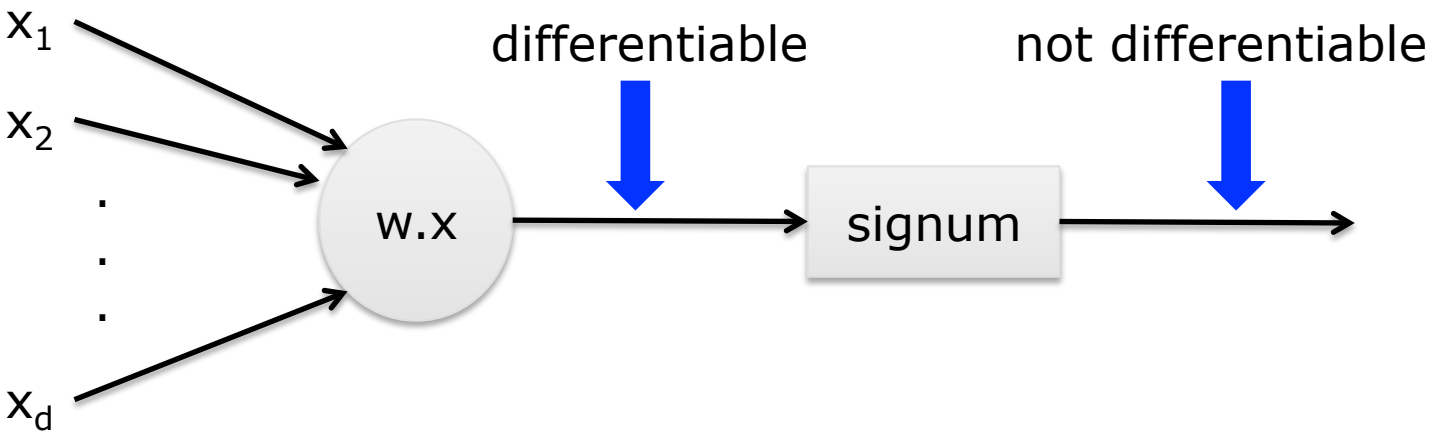



preference?

$$\mathcal{L}_{0/1}(h(X, \mathbf{w}), Y) = \sum_{(\mathbf{x},y)\in\mathcal{S}} [\![ y(\mathbf{w} \cdot \mathbf{x}) > 0 ]\!]$$

# are we done?

- no!
  - finding minimal misclassification hyperplane is NP-hard (even approximately)
  -



# stochastic gradient descent

**Input:** Labeled training data $\mathcal{S}$, Number of rounds $T$, Learning rate $\eta$

---

$\mathbf{w} \leftarrow 0$; $t \leftarrow 0$
**while** $t < T$ **do**
  **for all** $(\mathbf{x}, y) \in \mathcal{S}$ **do**
    $\mathbf{w} \leftarrow \mathbf{w} + \eta(y - \mathbf{w} \cdot \mathbf{x})\mathbf{x}$
  **end for**
**end while**

---

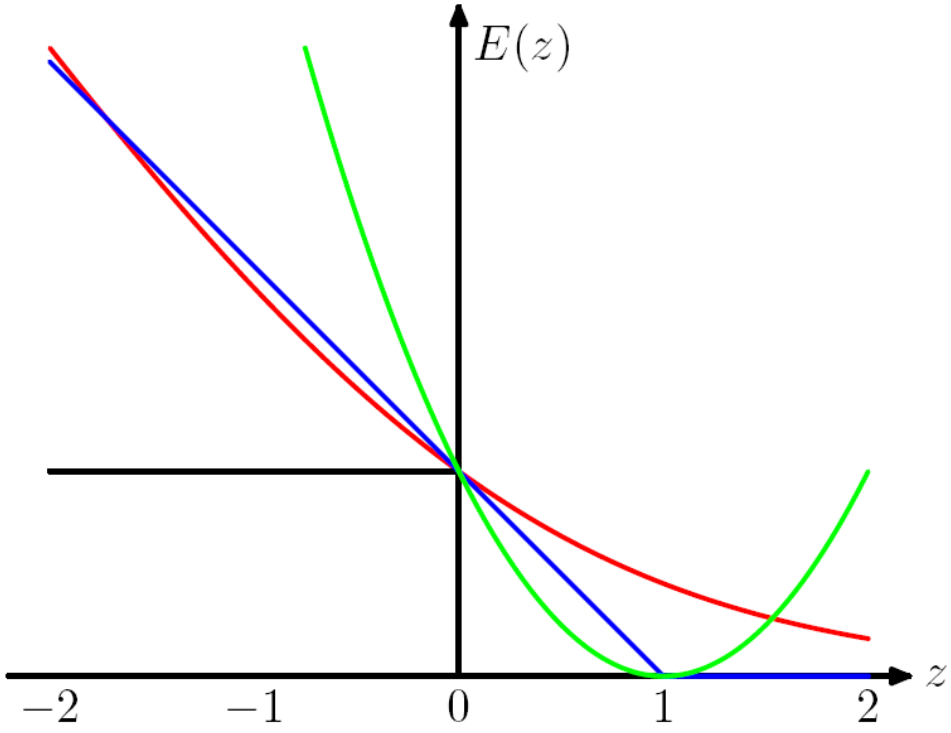
**Output:** Learned hypothesis $\mathbf{w}$

works frustratingly well

# perceptron

- linear classifier
- online
  - updates weight vector for each instance
- converges if data linearly separable
  - often the case for high dimensionality
  - (exact) mistake bound

# loss functions




$$\hat{\mathbf{w}} \leftarrow \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d} \sum_{(\mathbf{x},y)\in\mathcal{S}} \max(0, 1 - y(\mathbf{w}\cdot\mathbf{x})) - \lambda\|\mathbf{w}\|^2$$

# perceptron

**Input:** Labeled training data $\mathcal{S}$, Number of rounds $T$, Learning rate $\eta$

---

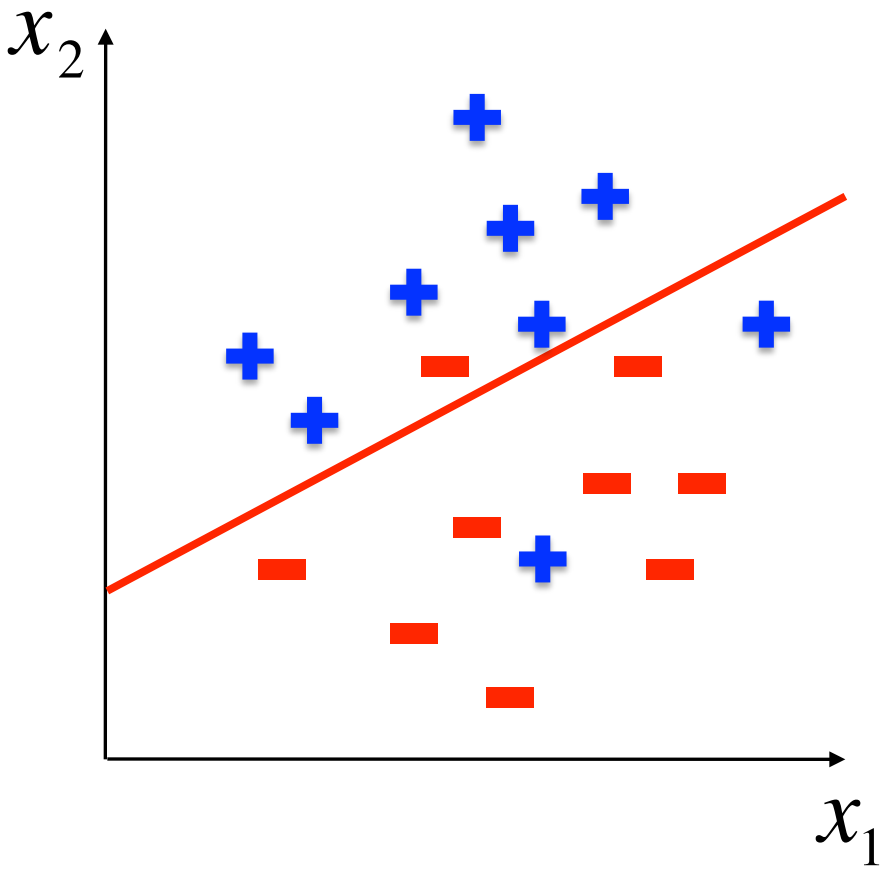
$\mathbf{w} \leftarrow 0$; $t \leftarrow 0$
**while** $t < T$ **do**
  **for all** $(\mathbf{x}, y) \in \mathcal{S}$ **do**
    **if** $y(\mathbf{w} \cdot \mathbf{x}) < 0$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \eta y \mathbf{x}$
    **end if**
  **end for**
**end while**

---

**Output:** Learned hypothesis $\mathbf{w}$

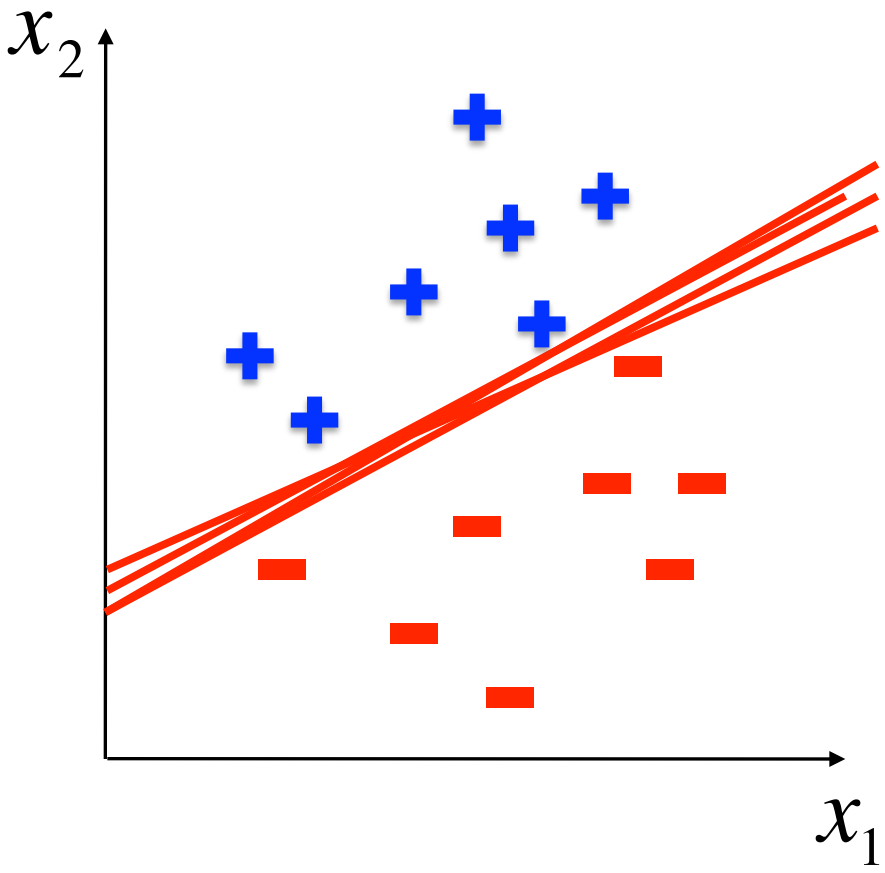
note regarding bias element

# perceptron




can be though of as iteratively pulling hyperplane toward misclassified examples

# perceptron



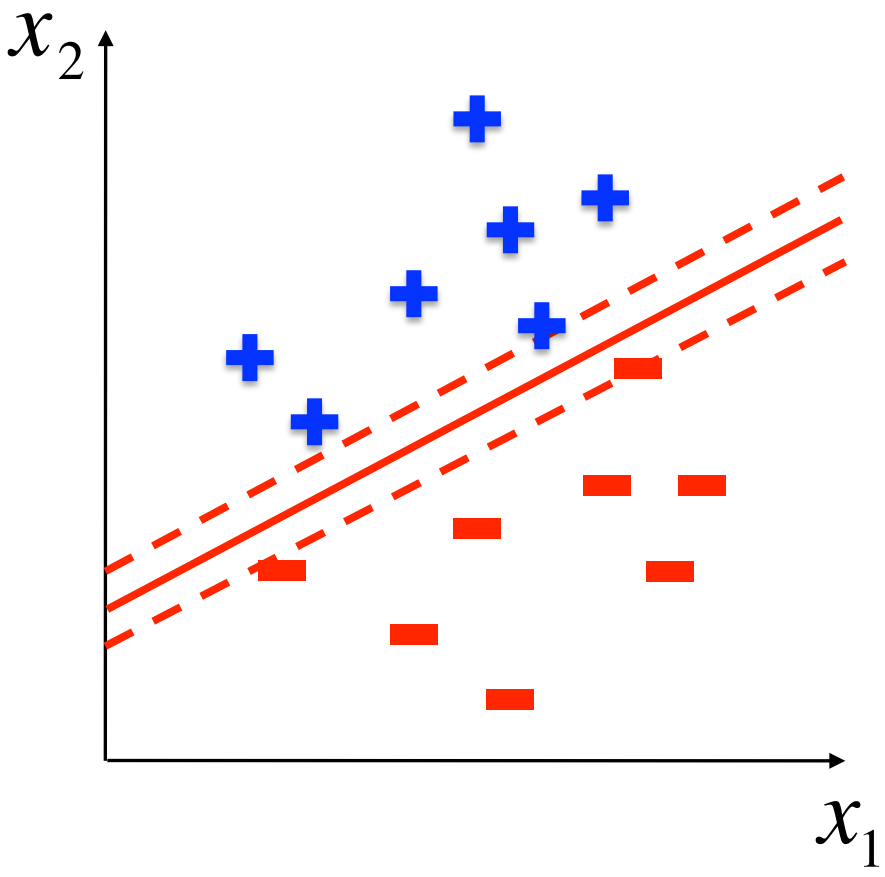

what about regularization?
how to select amongst consistent hypotheses?

# support vector machines (SVM)

- linear classifier*
- batch*
  - uses complete data sample at once
- converges by construction
  - computationally intensive*

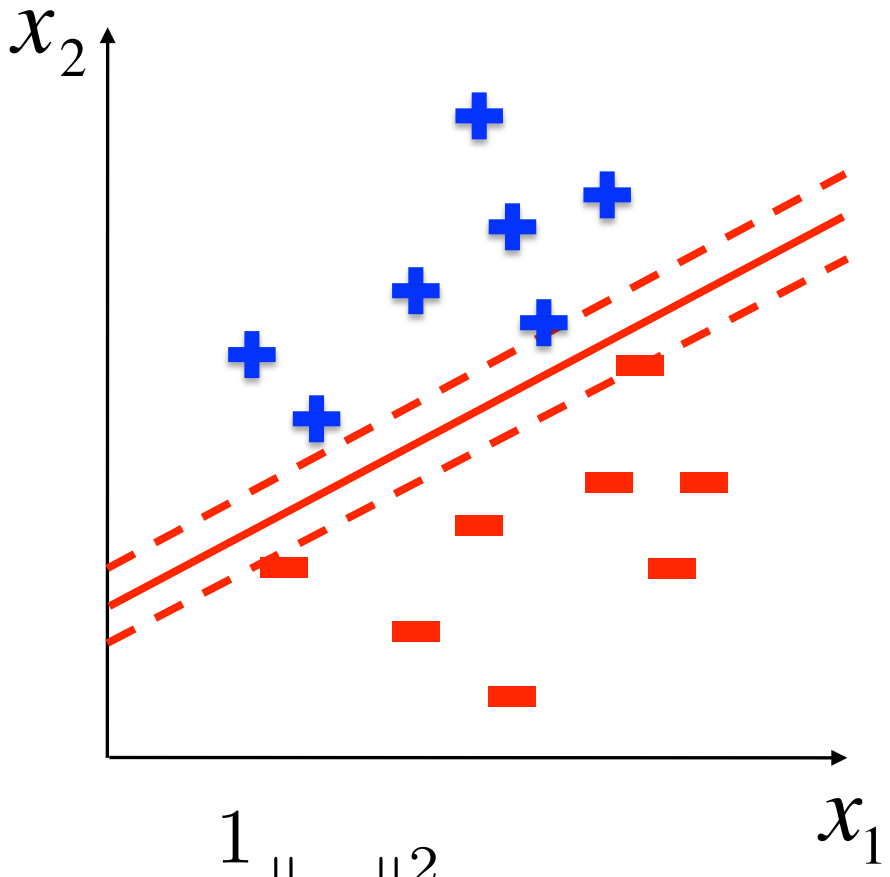
*recent work poses caveats for all these statements

# SVM




primary idea is max-min margin

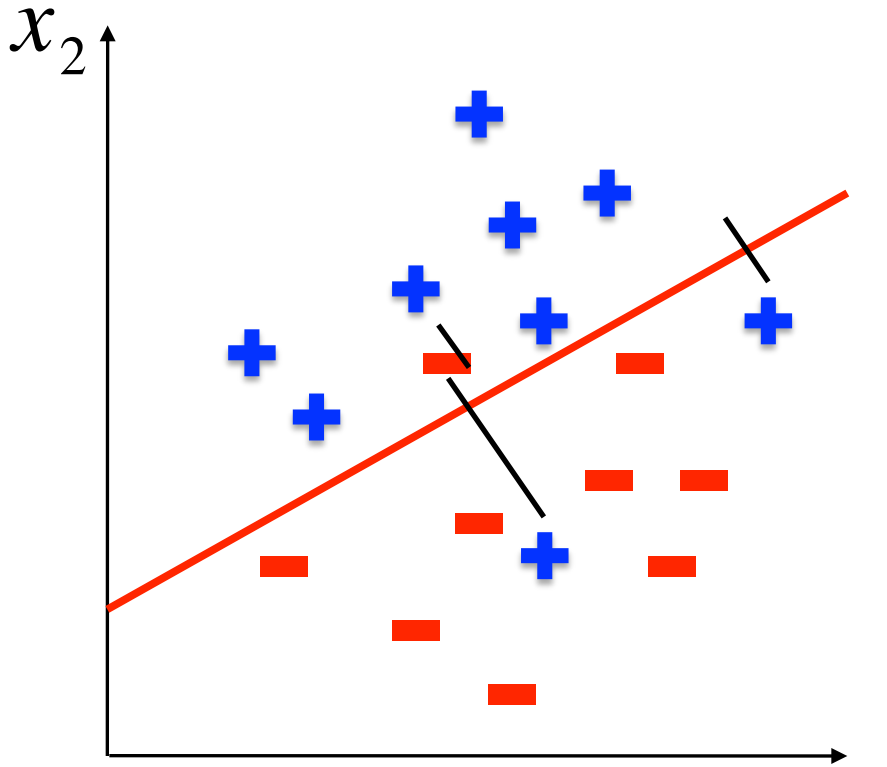
# SVM (separable)




$$\operatorname{argmin}_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \quad y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 \;\; \forall i = 1 \ldots m$$

# SVM (non-separable)

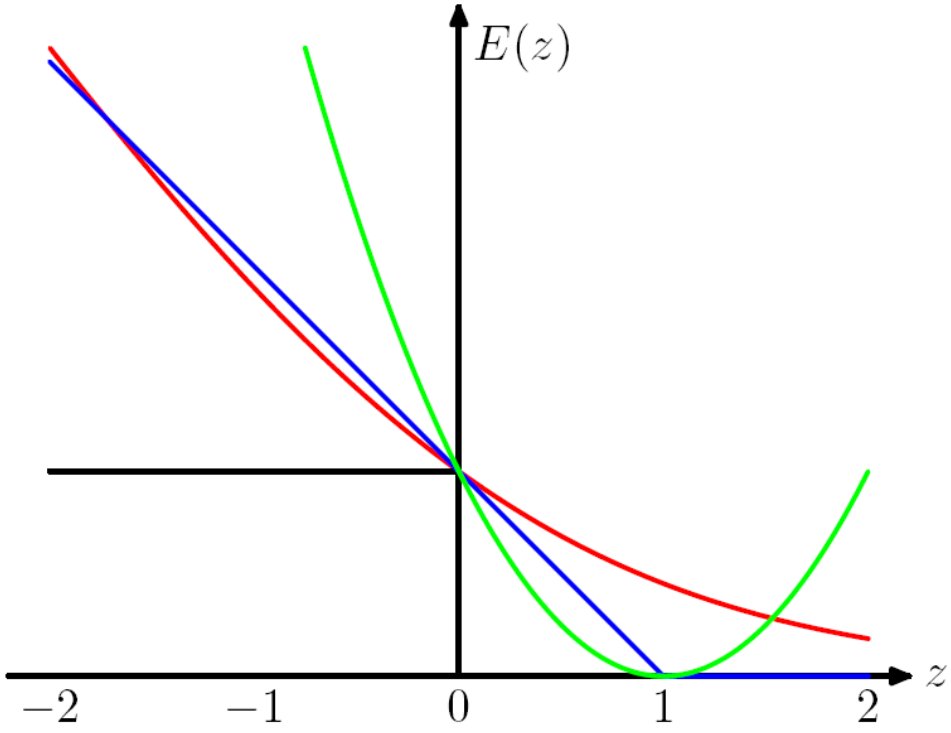



$$\text{argmin}_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t. \quad y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i \quad \forall i = 1\ldots m$$

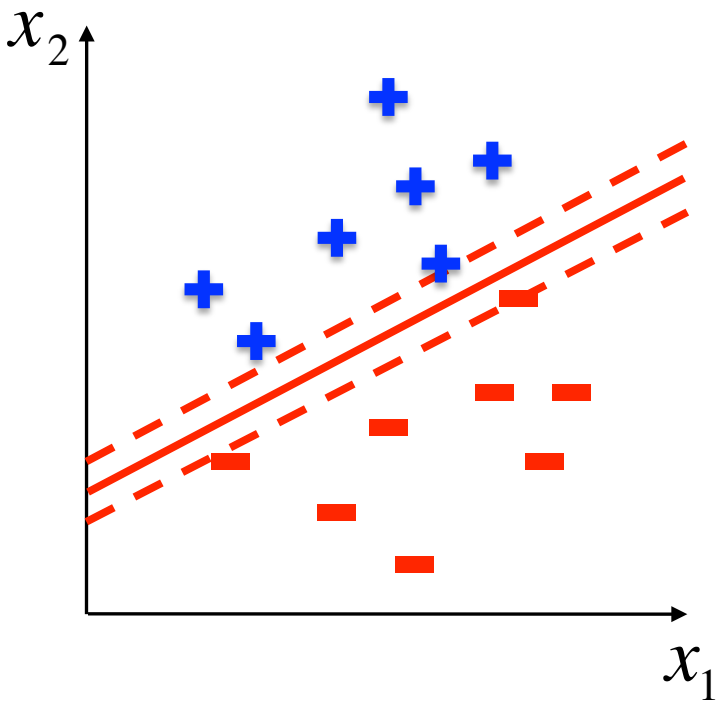
$$\xi_i \geq 0 \qquad\qquad \forall i = 1\ldots m.$$

# loss functions




$$\hat{\mathbf{w}} \leftarrow \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d} \sum_{(\mathbf{x},y)\in\mathcal{S}} \max(0, 1 - y(\mathbf{w}\cdot\mathbf{x})) - \lambda\|\mathbf{w}\|^2$$

# SVM dual (separable)

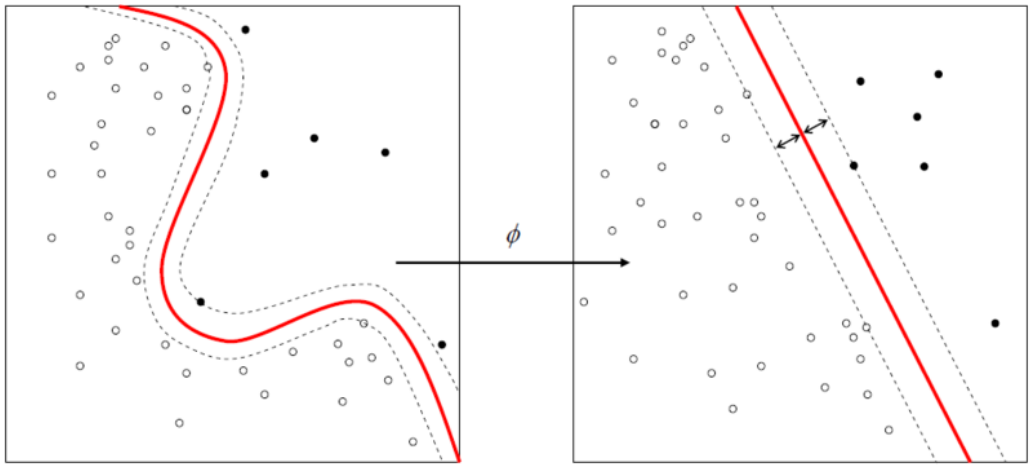



$$\begin{aligned} \text{argmax}_\alpha \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ s.t. \quad & \alpha_i \geq 0 \qquad \forall i = 1 \ldots m \\ & \sum_{i=1}^{m} \alpha_i y_i = 0 \qquad \forall i = 1 \ldots m. \end{aligned}$$

$$\mathbf{w} \leftarrow \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

# kernel “trick”

- $x_i \cdot x_j$ can be replaced with a kernel function $K(x_i, x_j)$
- maps data into a higher dimensional space which may be separable

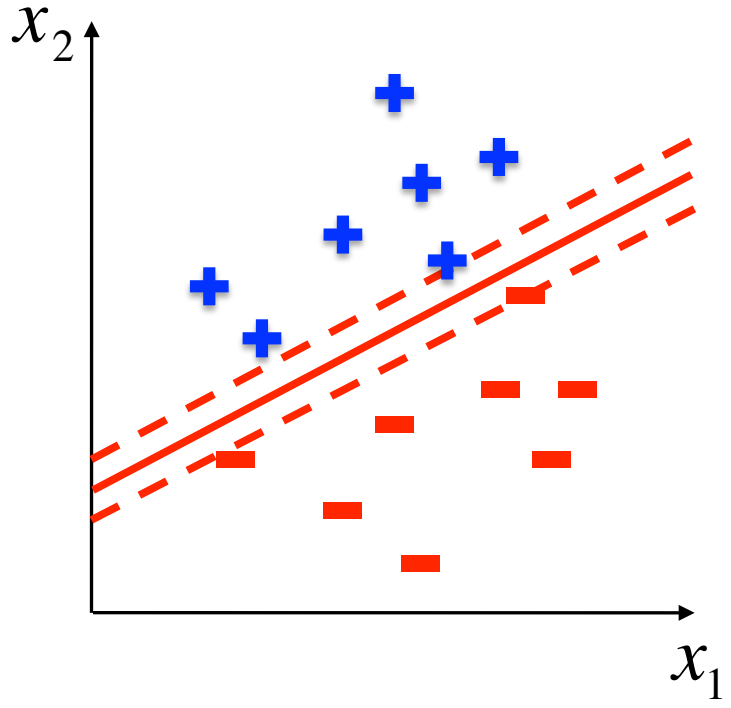


# kernel functions

- linear $\mathcal{K}_{linear}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- polynomial $\mathcal{K}_{poly}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$
- radial basis $\mathcal{K}_{rbf}(\mathbf{x}_i, \mathbf{x}_j) = e^{\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$
- many specialized kernels

# revisiting perceptron

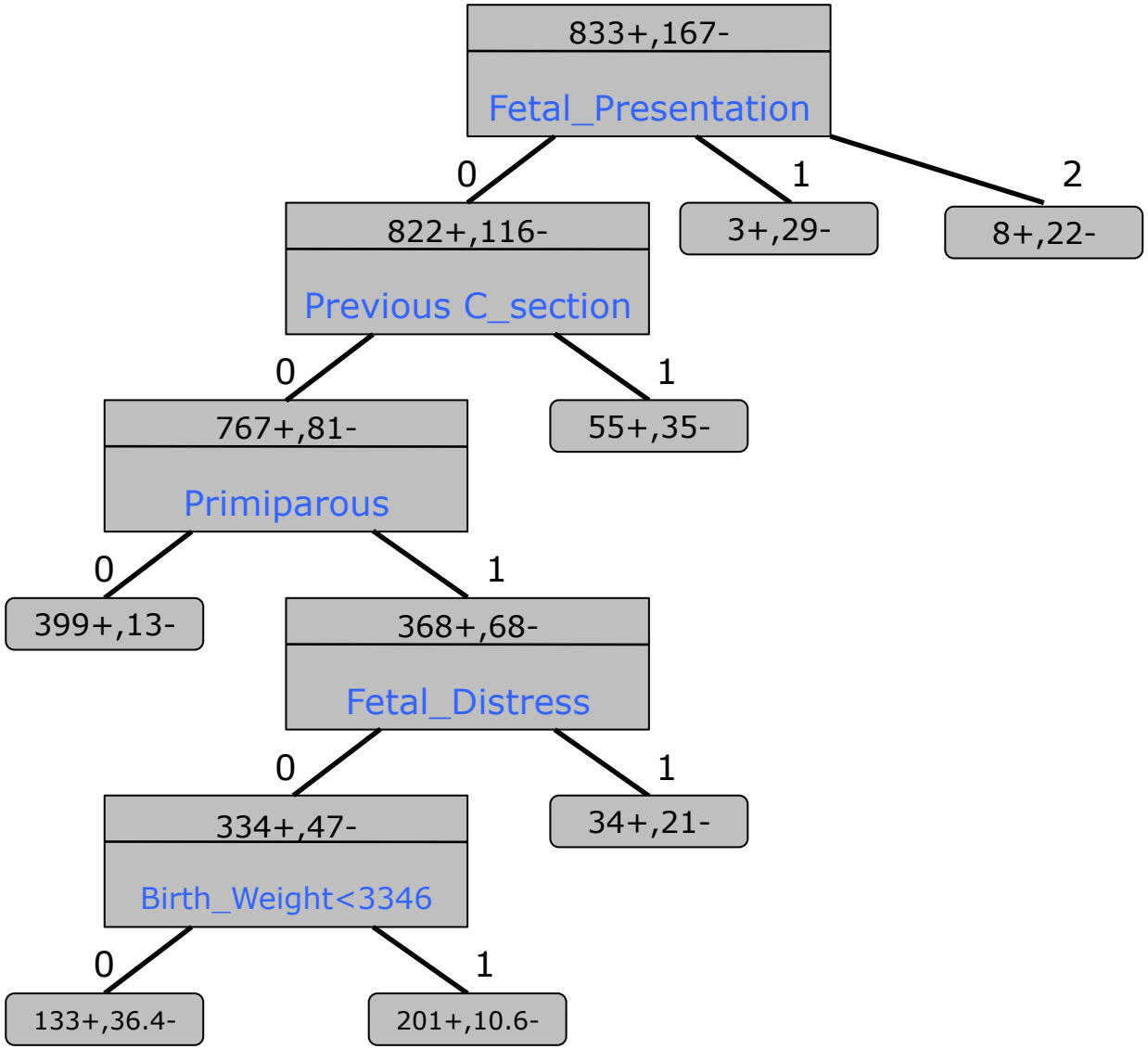
- max-margin heuristic
- hypothesis voting
- hypothesis averaging
- kernel perceptron



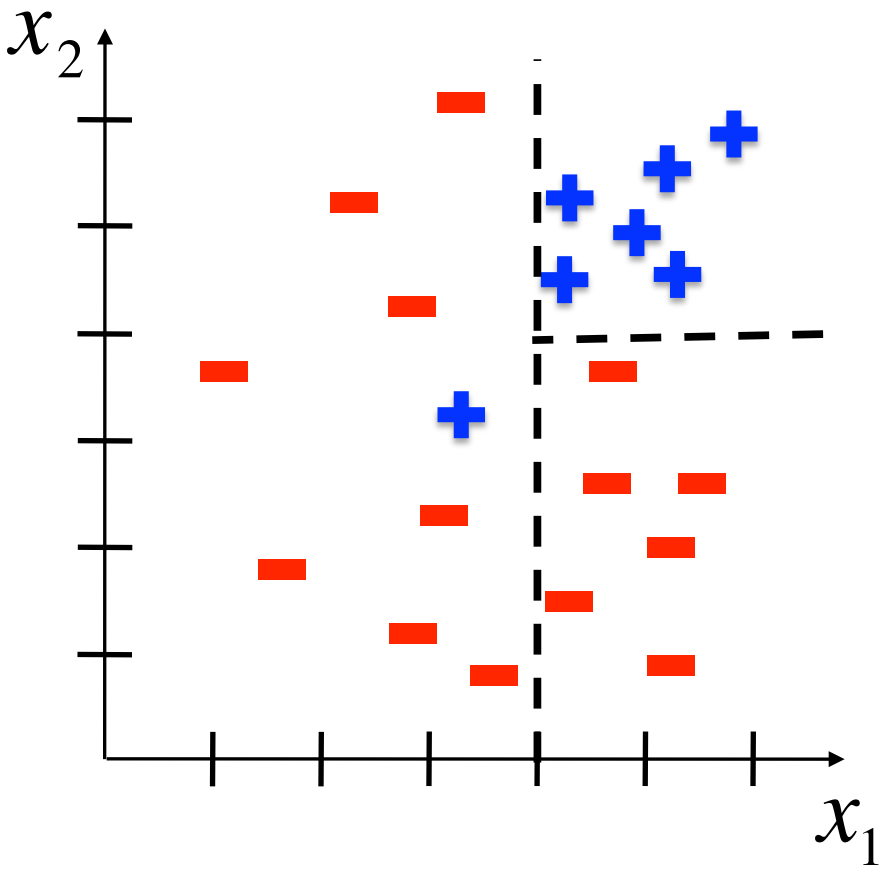
# decision trees

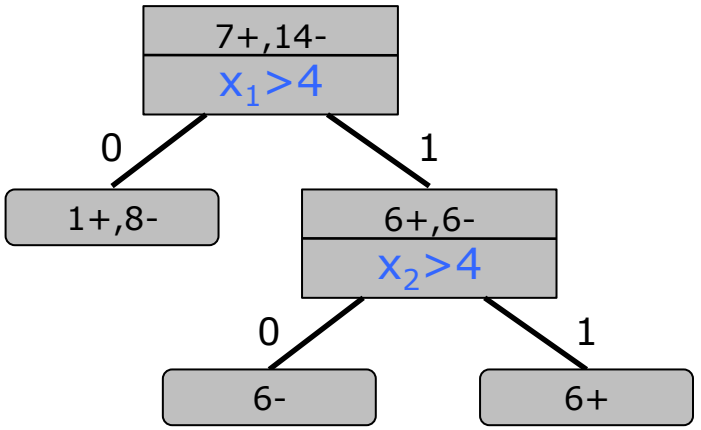
- defines regions via linear constraints
- eminently interpretable
  - important for interacting with humans
- batch
- converges by overfitting

# decision trees



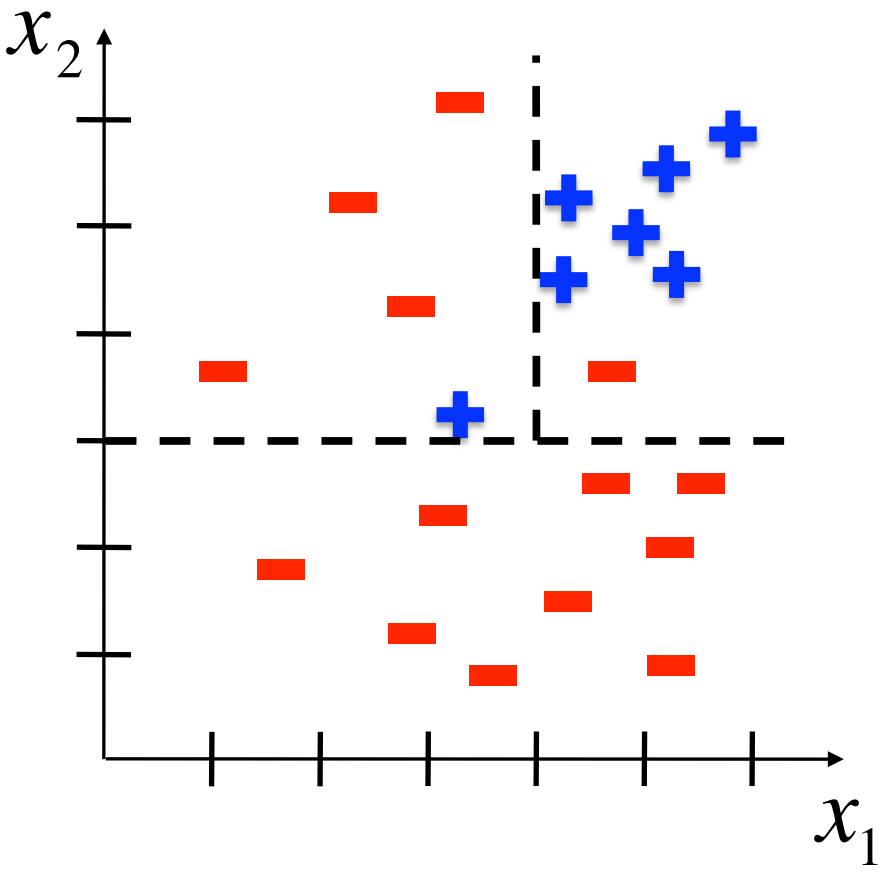
# decision trees

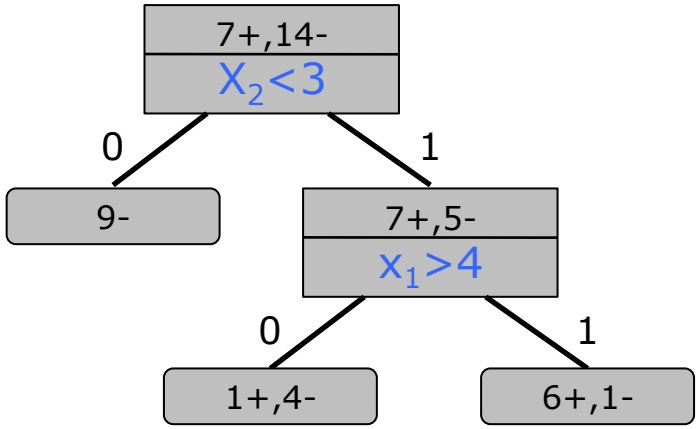



continue splitting?
where exactly to split?
arity of split?

# decision trees




generally use some form
of information gain
(prefers shorter trees)

# decision trees

- defines regions via linear constraints
  - allows for disjunctive functions
- eminently interpretable
  - important for interacting with humans
- batch
  - a search procedure
- converges by overfitting
  - ameliorated by structural constraints

# multiclass learning

- decision trees capable
- linear functions require networks
  - one-versus-all
  - all-versus-all
  - Kesler's construction