CS 461: Machine Learning
Lecture 3

Dr. Kiri Wagstaff
wkiri@wkiri.com

Questions?

- Homework 2
- Project Proposal
- Weka
- Other questions from Lecture 2

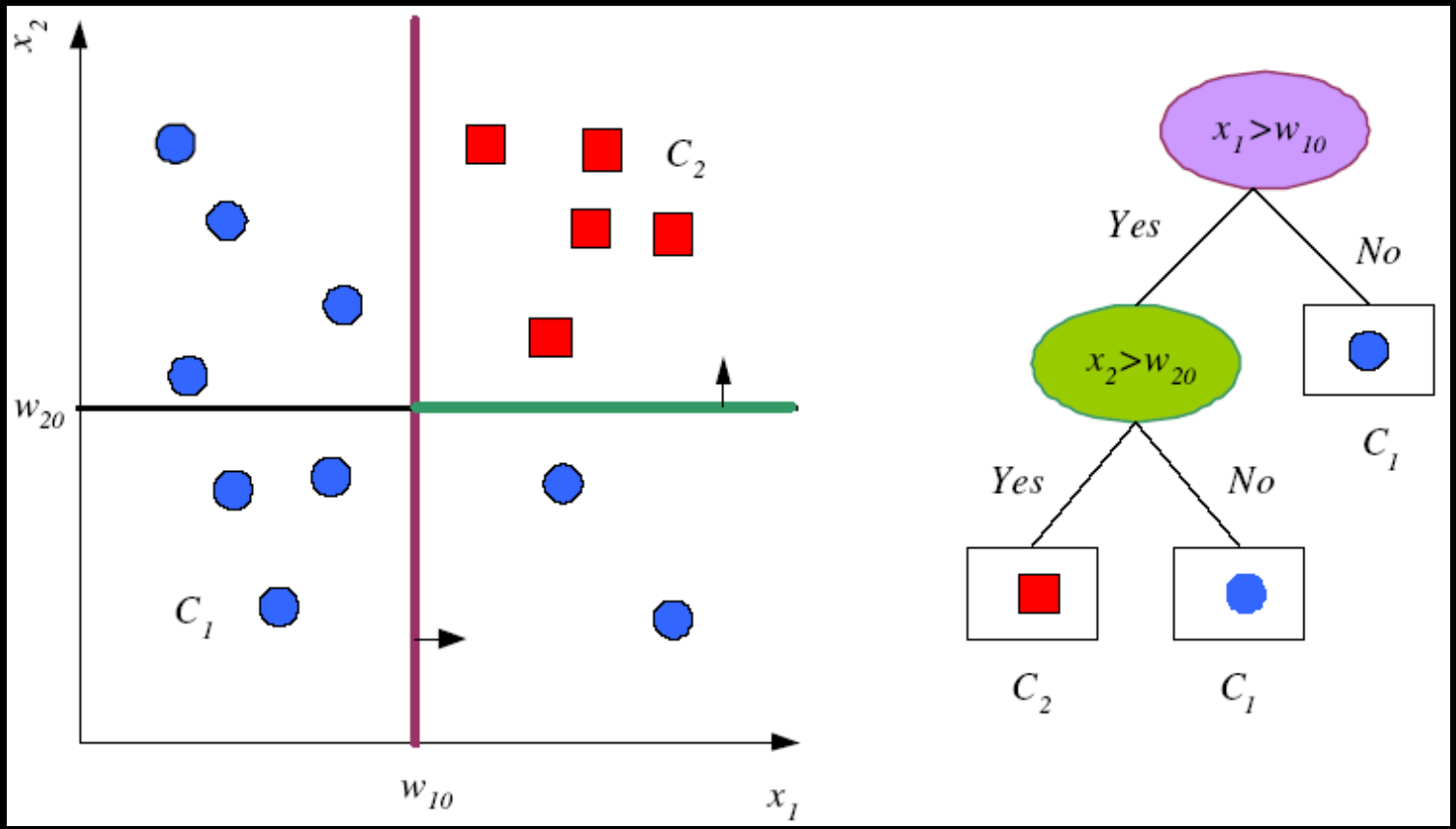
Review from Lecture 2

- Representation, feature types
  - numeric, discrete, ordinal
- Decision trees: nodes, leaves, greedy, hierarchical, recursive, non-parametric
- Impurity: misclassification error, entropy
- Evaluation: confusion matrix, cross-validation

Plan for Today

- Decision trees
  - Regression trees, pruning, rules
  - Benefits of decision trees
- Evaluation
  - Comparing two classifiers
- Support Vector Machines
  - Classification
    - Linear discriminants, maximum margin
    - Learning (optimization)
    - Non-separable classes
  - Regression

Remember Decision Trees?



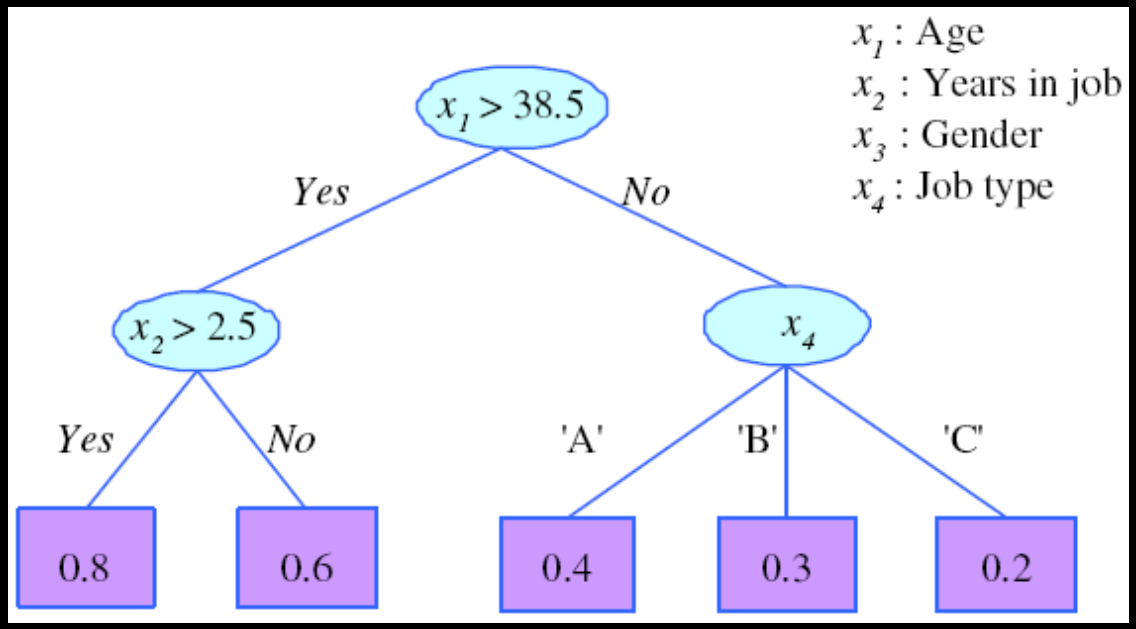
Algorithm: Build a Decision Tree

```
GenerateTree(X)
    If NodeEntropy(X)< θ_I /* eq. 9.3
        Create leaf labelled by majority class in X
        Return
    i ← SplitAttribute(X)
    For each branch of x_i
        Find X_i falling in branch
        GenerateTree(X_i)
SplitAttribute(X)
    MinEnt← MAX
    For all attributes i = 1,...,d
        If x_i is discrete with n values
            Split X into X_1,...,X_n by x_i
            e ← SplitEntropy(X_1,...,X_n) /* eq. 9.8 */
            If e<MinEnt MinEnt ← e; bestf ← i
        Else /* x_i is numeric */
            For all possible splits
                Split X into X_1, X_2 on x_i
                e←SplitEntropy(X_1, X_2)
                If e<MinEnt MinEnt ← e; bestf ← i
    Return bestf
```

Building a Regression Tree

- Same algorithm… different criterion
- Instead of impurity, use Mean Squared Error (in local region)
  - Predict mean output for node
  - Compute training error
  - (Same as computing the variance for the node)
- Keep splitting until node error is acceptable; then it becomes a leaf
  - Acceptable: error < threshold

Turning Trees into Rules



R1: IF (age>38.5) AND (years-in-job>2.5) THEN y =0.8
R2: IF (age>38.5) AND (years-in-job≤2.5) THEN y =0.6
R3: IF (age≤38.5) AND (job-type='A') THEN y =0.4
R4: IF (age≤38.5) AND (job-type='B') THEN y =0.3
R5: IF (age≤38.5) AND (job-type='C') THEN y =0.2

Comparing Two Algorithms

Chapter 14

Machine Learning Showdown!

- McNemar's Test

e_{00}: Number of examples misclassified by both	e_{01}: Number of examples misclassified by 1 but not 2
e_{10}: Number of examples misclassified by 2 but not 1	e_{11}: Number of examples correctly classified by both

- Under H_0, we expect $e_{01} = e_{10} = (e_{01} + e_{10})/2$

$$\frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \sim \chi_1^2$$

Accept if $< \chi^2_{\alpha,1}$

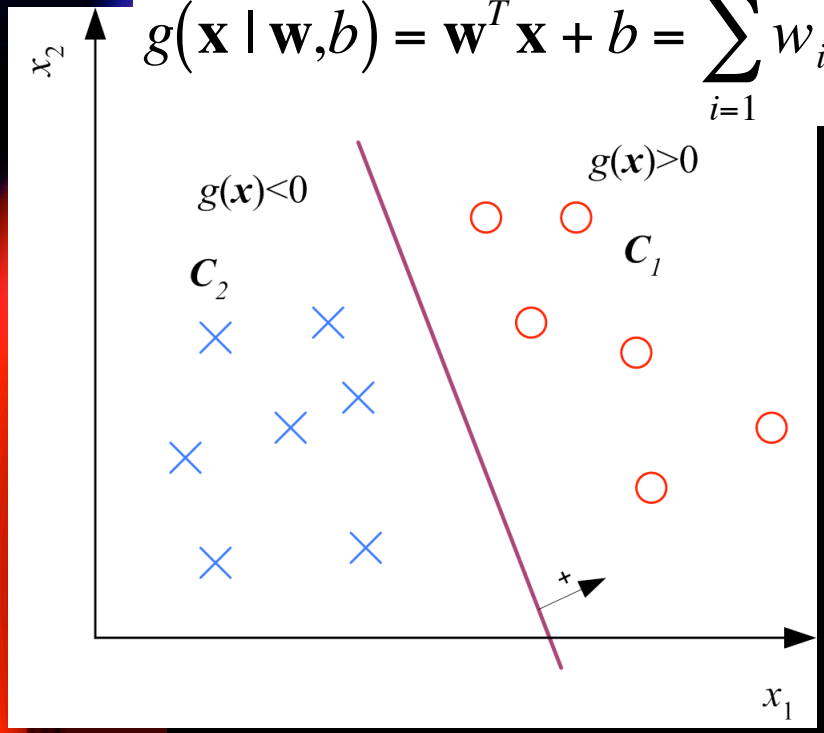
Support Vector Machines

Chapter 10

Linear Discrimination

- Model class boundaries (not data distribution)
- Learning: maximize accuracy on labeled data
- Inductive bias: form of discriminant used

$$g(\mathbf{x} \mid \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^{d} w_i x_i + b$$



$$\text{choose} \begin{cases} C_1 & \text{if } g(\boldsymbol{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

How to find best w, b?

- $E(\boldsymbol{w}|\mathcal{X})$ is error with parameters $\boldsymbol{w}$ on sample $\mathcal{X}$

  $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w} \mid \mathcal{X})$

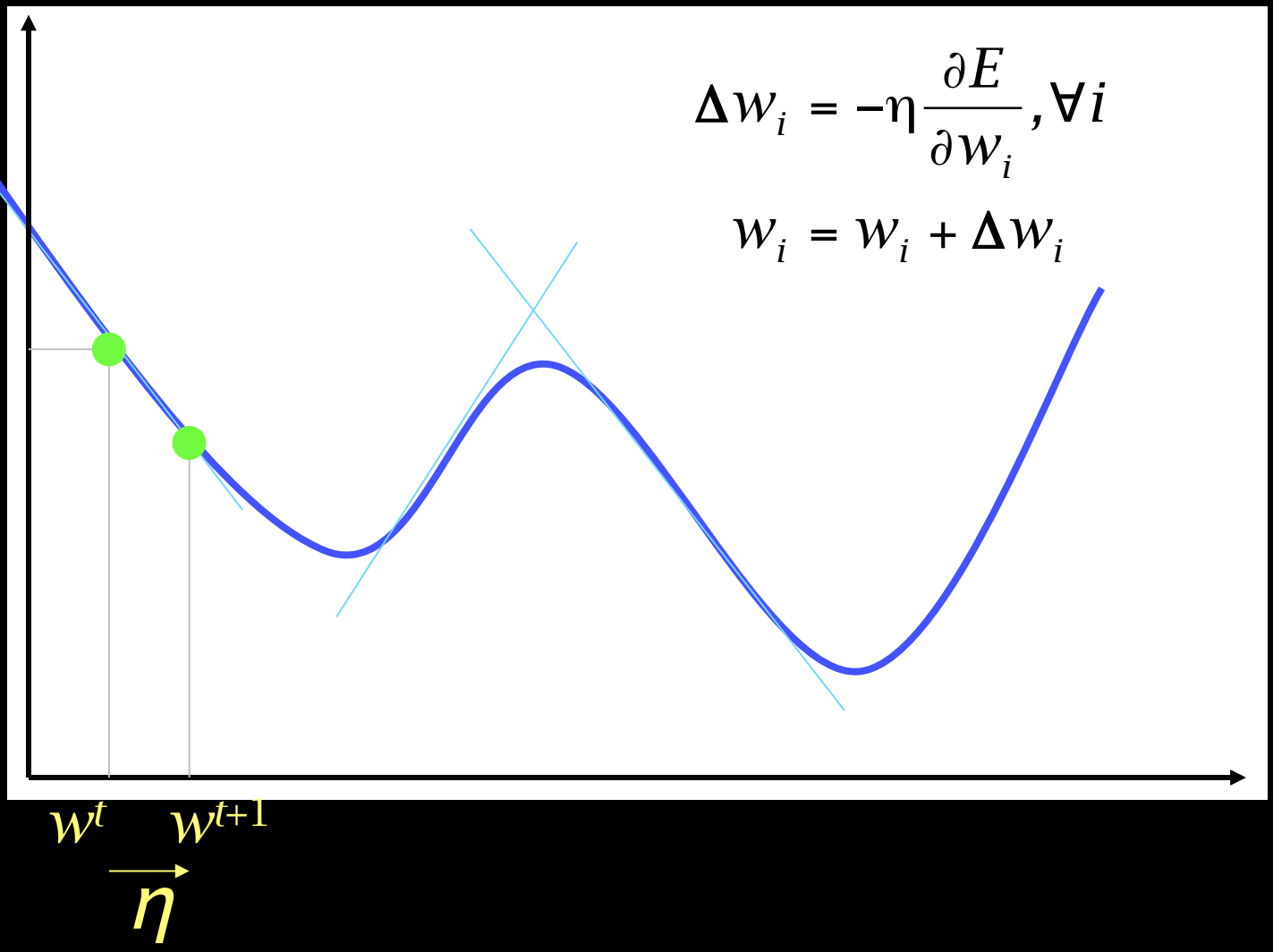
- Gradient

$$\nabla_w E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_d} \right]^T$$

- Gradient-descent:

  Starts from random $\boldsymbol{w}$ and updates $\boldsymbol{w}$ iteratively in the negative direction of gradient

Gradient Descent



Support Vector Machines

- Maximum-margin linear classifiers
  - Imagine: Army Ceasefire
- How to find best w, b?
  - Quadratic programming:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } y^t\left(\mathbf{w}^T\mathbf{x}^t + b\right) \geq +1, \forall t$$

Optimization (primal formulation)

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } y^t\left(\mathbf{w}^T\mathbf{x}^t + b\right) \geq +1, \forall t$$

Must get training data right!

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N}\alpha^t\left[y^t\left(\mathbf{w}^T\mathbf{x}^t + b\right) - 1\right]$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N}\alpha^t y^t\left(\mathbf{w}^T\mathbf{x}^t + b\right) + \sum_{t=1}^{N}\alpha^t$$

N + d +1 parameters

Optimization (dual formulation)

We know:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{t=1}^{N} \alpha^t y^t \mathbf{x}^t$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{t=1}^{N} \alpha^t y^t = 0$$

So re-write:

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t y^t \left(\mathbf{w}^T \mathbf{x}^t + b\right) + \sum_{t=1}^{N} \alpha^t$$

$$\begin{aligned} L_d &= \frac{1}{2}\left(\mathbf{w}^T \mathbf{w}\right) - \mathbf{w}^T \sum_t \alpha^t y^t \mathbf{x}^t - b \sum_t \alpha^t y^t + \sum_t \alpha^t \\ &= -\frac{1}{2}\left(\mathbf{w}^T \mathbf{w}\right) + \sum_t \alpha^t \\ &= -\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s y^t y^s \left(\mathbf{x}^t\right)^T \mathbf{x}^s + \sum_t \alpha^t \end{aligned}$$

subject to $\sum_t \alpha^t y^t = 0$ and $\alpha^t \geq 0, \forall t$

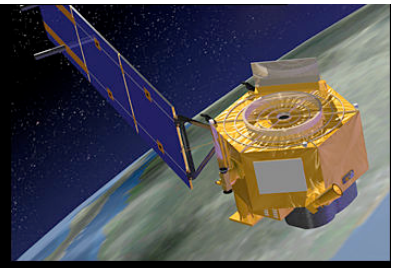
$\alpha^t > 0$ are the SVs, capped by C

N parameters. Where did **w** and b go?

What if Data isn't Linearly Separable?

1. Embed data in higher-dimensional space
   - Explicit: Basis functions (new features)
     - Visualization of 2D -> 3D
   - Implicit: Kernel functions (new dot product/similarity)
     - Polynomial
     - RBF/Gaussian
     - Sigmoid
     - SVM applet
   - Still need to find a linear hyperplane
2. Add "slack" variables to permit some errors

Example: Orbital Classification



EO-1

- Linear SVM flying on EO-1 Earth Orbiter since Dec. 2004
  - Classify every pixel
  - Four classes
  - 12 features (of 256 collected)

Hyperion

Classified

Ice
Water
Land
Snow

[Castano et al., 2005]

SVM in Weka

- SMO: Sequential Minimal Optimization
  - Faster than QP-based versions
  - Try linear, RBF kernels

Summary: What You Should Know

- Decision trees
  - Regression trees, pruning, rules
  - Benefits of decision trees
- Evaluation
  - Comparing two classifiers (McNemar's test)
- Support Vector Machines
  - Classification
    - Linear discriminants, maximum margin
    - Learning (optimization)
    - Non-separable classes

Next Time

- Reading
  - Evaluation
    (read Ch. 14.7)
  - Support Vector Machines
    (read Ch. 10.1-10.4, 10.6, 10.9)
- Questions to answer from the reading
  - Posted on the website
  - Three volunteers: Sen, Jimmy, and Irvin