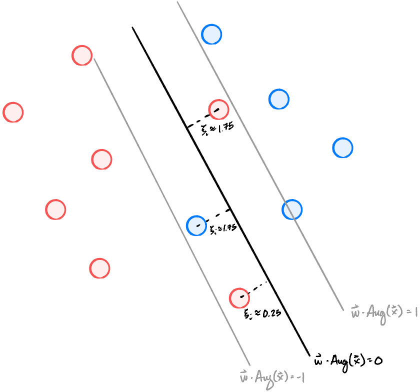


CSE 151A

Intro to Machine Learning

Lecture 12 – Part 01

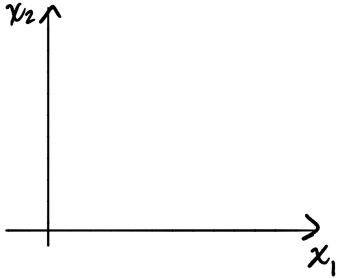
Support Vector Machines

Linear Classifiers

- **Prediction rule**: $H(\vec{x}) = \vec{w} \cdot \mathrm{Aug}(\vec{x})$
  - Predict class 1 if $H(\vec{x}) > 0$
  - Predict class -1 if $H(\vec{x}) < 0$

Decision Boundary

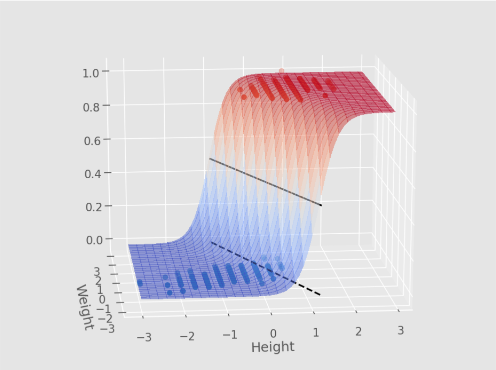
- $\text{Aug}(\vec{x}) \cdot \vec{w}$ is proportional to distance from boundary.



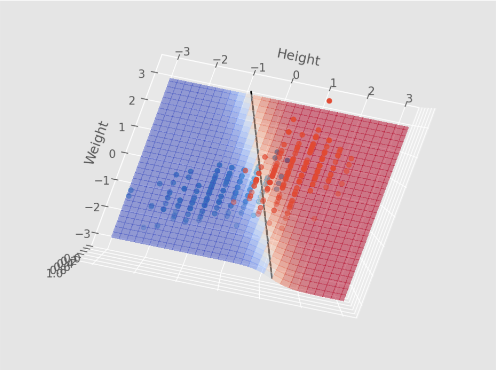
Recall: Logistic Regression

- **Prediction Rule**: $H(\vec{x}) = \sigma(\vec{w} \cdot \mathrm{Aug}(\vec{x}))$
- Find $\vec{w}$ by maximizing log likelihood.
- Predict class 1 if $H(\vec{x}) > 0.5$, class -1 otherwise.
- But $\sigma(\vec{w} \cdot \mathrm{Aug}(\vec{x})) > 0.5 \iff \vec{w} \cdot \mathrm{Aug}(\vec{x}) > 0$

Recall: Logistic Regression



Recall: Logistic Regression

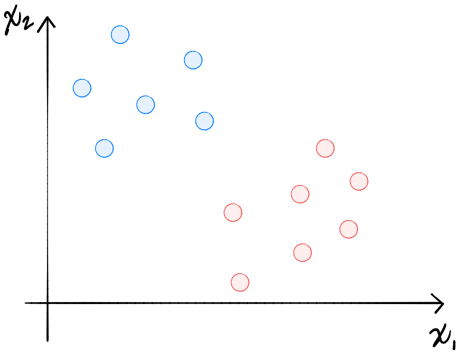


Recall: the Perceptron

- **Prediction Rule:** $H(\vec{x}) = \vec{w} \cdot \mathrm{Aug}(\vec{x})$
- Find $\vec{w}$ by minimizing perceptron risk.
- **Theorem**: if the training data is **linearly separable**, the perceptron algorithm find a dividing hyperplane.

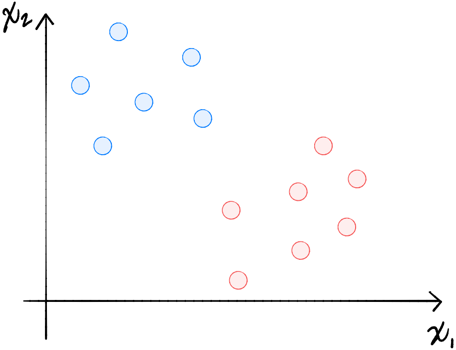
Perceptron Problems

The learned perceptron may have a small **margin**.



Perceptron Problems

The learned perceptron may have a small **margin**.



We prefer **large margins** for generalization.

Maximum Margin Classifiers

- **Assume**: linear separability (for now).
- Many possible boundaries with zero error.
- **Goal**: Find linear boundary with largest margin w.r.t. training data.

Observation

- Training data: $\{(\vec{x}^{(i)}, y_i)\}$
- Classification is correct when:

$$\begin{cases} \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) > 0, & \text{if } y_i = 1] \\ \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) < 0, & \text{if } y_i = -1] \end{cases}$$

- Equivalently, classification is correct if:

$$y_i \, \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) > 0$$

Recall

- $y_i \vec{w} \cdot \text{Aug}(\vec{x}^{(i)}) \propto$ to distance from boundary.
- Our goal: find $\vec{w}$ that maximizes the smallest distance.

$$\vec{w}_{\text{best}} = \underset{\vec{w} \in \mathbb{R}^{d+1}}{\text{argmax}} \min_{\vec{i} \in 1,\ldots,n} \left[y_i \vec{w} \cdot \text{Aug}(\vec{x}^{(i)}) \right]$$

- This looks **hard**. But there is a **trick**.

Another Observation

- If linearly separable, then there is a $\vec{w}$ such that

$$y_i \, \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) > 0$$

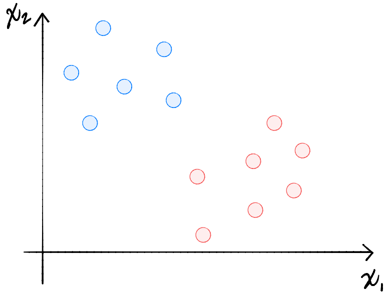
for all $i = 1, \ldots, n$.

- Actually, linearly separable $\implies$ there is a $\vec{\omega}$ s.t.

$$y_i \, \vec{\omega} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1$$

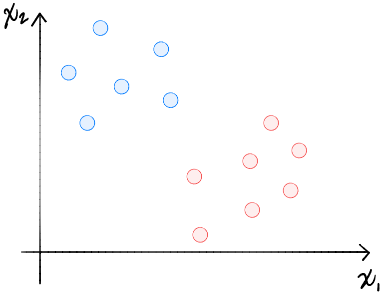
for all $i = 1, \ldots, n$.

Why?



- Suppose $\vec{w}$ separates, but $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) = 0.01$
- Define $\vec{\omega} = \frac{1}{0.01}\vec{w} = 100\vec{w}$.
- Then $y_i \vec{\omega} \cdot \mathrm{Aug}(\vec{x}^{(i)}) = 1$
- **Note**: $\|\vec{\omega}\|$ is large!

Why?



- Suppose $\vec{w}$ separates, but $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) = 0.5$
- Define $\vec{\omega} = \frac{1}{0.5}\vec{w} = 2\vec{w}$.
- Then $y_i \vec{\omega} \cdot \mathrm{Aug}(\vec{x}^{(i)}) = 1$
- **Note**: $\|\vec{\omega}\|$ is smaller!

The Trick

- We will demand that

$$y_i \, \vec{\omega} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1$$

- The larger $\|\vec{\omega}\|$, the smaller the margin.
- **New Goal**: Minimize $\|\vec{w}\|^2$ subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1$ for all i.

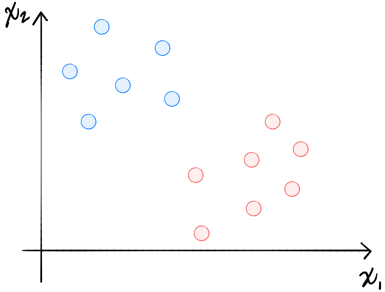
Optimization

- Minimize $\|\vec{w}\|^2$ subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1$ for all i.
- This is a **convex, quadratic** optimization problem.
- Can be solved efficiently with **quadratic programming**.
  - But there is no exact general formula for the solution

Support Vectors

- A **support vector** is a training point $\vec{x}^{(i)}$ such that

$$y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) = 1$$



Support Vector Machines (SVMs)

- Then maximum margin solution $\vec{w}$ is a linear combination of the support vectors.

- Let S be the set of support vectors. Then

$$\vec{w} = \sum_{i \in S} y_i \alpha_i \operatorname{Aug}(\vec{x}^{(i)})$$

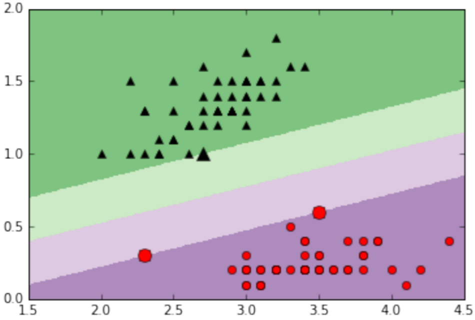
Example: Irises

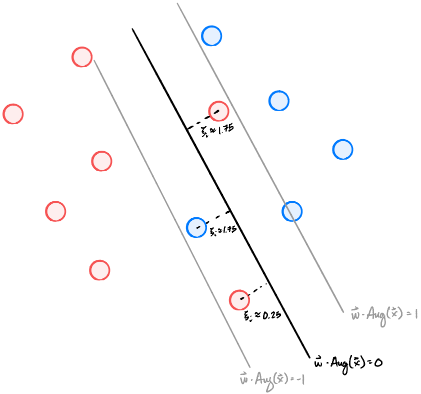


- 3 classes: *iris setosa*, *iris versicolor*, *iris virginica*
- 4 measurements: petal width/height, sepal width/height

Example: Irises

- Using only sepal width/petal width
- Two classes: versicolor (black), setosa (red)



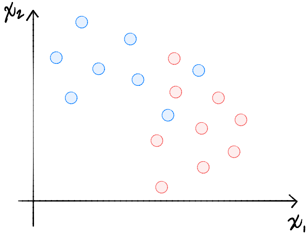


CSE 151A
Intro to Machine Learning

Lecture 12 – Part 02
Soft-Margin SVMs

Non-Separability

- So far we've assumed data is linearly separable.
- What if it isn't?

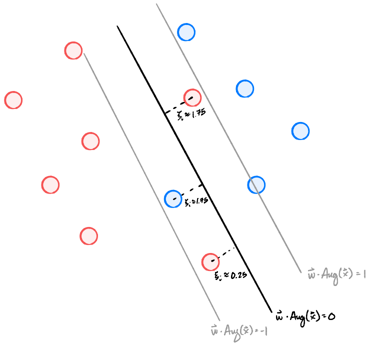


The Problem

- **Old Goal**: Minimize $\|\vec{w}\|^2$ subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1$ for all i.
- This **no longer makes sense**.

Cut Some Slack

- **Idea**: allow some classifications to be ξ_i wrong, but not too wrong.



Cut Some Slack

- **New problem.** Fix some number $C \geq 0$.

$$\min_{\vec{w} \in \mathbb{R}^{d+1}, \vec{\xi} \in \mathbb{R}^n} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1 - \xi_i$ for all i, $\vec{\xi} \geq 0$.

The Slack Parameter, C

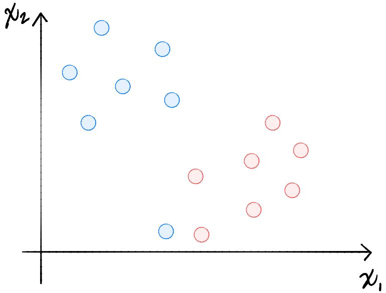
- C controls how much slack is given.

$$\min_{\vec{w} \in \mathbb{R}^{d+1}, \vec{\xi} \in \mathbb{R}^n} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

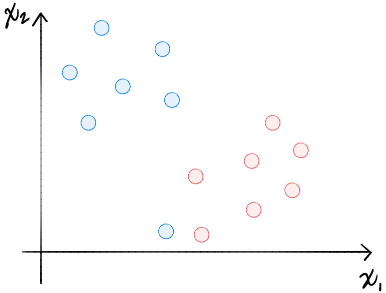
subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1 - \xi_i$ for all i, $\vec{\xi} \geq 0$.

  - Large C: don't give much slack. Avoid misclassifications.
  - Small C: allow more slack at the cost of misclassifications.

Example: Small C



Example: Large C



Soft and Hard Margins

- Max-margin SVM from before has **hard margin**.
- Now: the **soft margin** SVM.
- As $C \to \infty$, the margin hardens.

Another View: Loss Functions

- Recall our problem:

$$\min_{\vec{w} \in \mathbb{R}^{d+1}, \vec{\xi} \in \mathbb{R}^n} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)}) \geq 1 - \xi_i$ for all i, $\vec{\xi} \geq 0$.

- **Note**: if $\vec{x}^{(i)}$ is misclassified, then

$$\xi_i = 1 - y_i \vec{w} \cdot \mathrm{Aug}(\vec{x}^{(i)})$$

Another View: Loss Functions

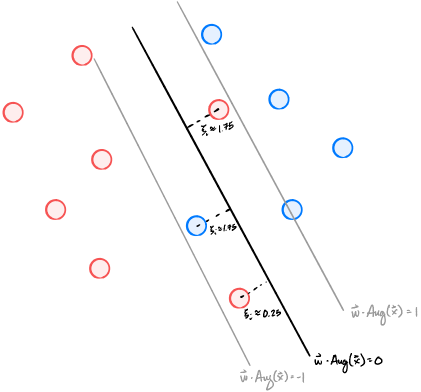
- New problem:

$$\min_{\vec{w}\in\mathbb{R}^{d+1},\vec{\xi}\in\mathbb{R}^n} \|\vec{w}\|^2 + C\sum_{i=1}^{n} \max\{0, 1 - y_i \vec{w}\cdot\vec{x}^{(i)}\}$$

- $\max\{0, 1 - y_i \vec{w}\cdot\vec{x}^{(i)}\}$ is called the **hinge loss**.

Another Way to Optimize

- We can use **subgradient descent** to minimize SVM risk.



CSE 151A
Intro to Machine Learning

Lecture 12 – Part 03
Sentiment Analysis

Why use linear predictors?

- Linear classifiers look to be very simple.
- That can be both **good** and **bad**.
  - **Good**: the math is tractable, less likely to overfit
  - **Bad**: may be too simple, underfit
- They can work surprisingly well.

Sentiment Analysis

- **Given**: a piece of text.
- **Determine**: if it is **postive** or **negative** in tone
- Example: “Needless to say, I wasted my money.”

The Data

- Sentences from reviews on Amazon, Yelp, IMDB.
- Each labeled (by a human) **positive** or **negative**.
- Examples:
  - **“Needless to say, I wasted my money.”**
  - **“I have to jiggle the plug to get it to line up right.”**
  - **“Will order from them again!”**
  - **“He was very impressed when going from the original battery to the extended battery.”**

The Plan

- We'll train a soft-margin SVM.
- **Problem**: SVMs take **fixed-length vectors** as inputs, not sentences.

Bags of Words

To turn a document into a fixed-length vector:

- First, choose a **dictionary** of words:
  - E.g.: `["wasted", "impressed", "great", "bad", "again"]`
- Count number of occurrences of each dictionary word in document.
  - "It was bad. So bad that I was impressed at how bad it was." $\rightarrow (0, 1, 0, 3, 0)^T$
- This is called a **bag of words** representation.

Choosing the Dictionary

- Many ways of choosing the dictionary.
- Easiest: take all of the words in the training set.
  - Perhaps throw out **stop words** like “the”, “a”, etc.
- Resulting dimensionality of feature vectors: large.

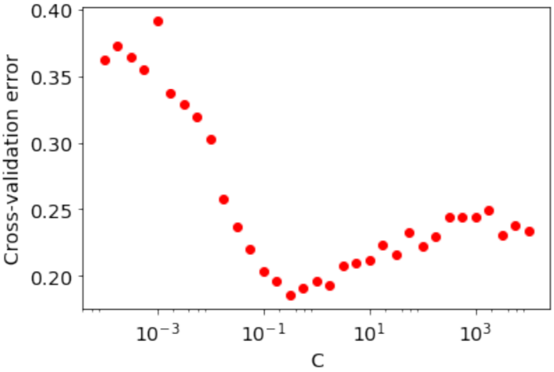
Experiment

- Bag of words features with 4500 word dictionary.
- 2500 training sentences, 500 test sentences.
- Train a soft margin SVM.

Choosing C

- We have to choose the slack parameter, C.
- Use **cross validation**!

Cross Validation



Results

- With $C = 0.32$, test error $\approx 15.6\%$.

C	training error (%)	test error (%)	# support vectors
0.01	23.72	28.4	2294
0.1	7.88	18.4	1766
1	1.12	16.8	1306
10	0.16	19.4	1105
100	0.08	19.4	1035
1000	0.08	19.4	950