Info 251: Applied Machine Learning
Lab 9
4/1/2020

Topics

- Support Vector Machines (SVM)
- Decision Trees
- Random Forests
- Neural Networks (next lab...)

SVM

- Dataset of N pairs x_i, y_i with $y_i \in \{-1, 1\}$ and $x_i \in \mathbb{R}^n$
- Classifier function $f(x) = sign(w^T x + b) \in \{-1, 1\}$
- Intuition: In logistic regression we classify with $g(w) = \frac{1}{1+e^{-b-w^T x}}$
- The higher the value of $w^T x$ the more confident we are that label is 1 and the lower the more likely that the label is -1.

SVM

- Dataset of$N$ pairs x_i, y_i with $y_i \in \{-1, 1\}$ and $x_i \in \mathbb{R}^n$
- Classifier function $f(x) = sign(w^T x + b)$
- Solve:

$$\begin{aligned} \max_{w,b} \quad & \alpha \\ \text{s.t.} \quad & w^T x_i + b \geq \alpha \text{ if } y_i = 1 \\ & w^T x_i + b \leq -\alpha \text{ if } y_i = -1 \\ & ||w||_2 = 1 \end{aligned} \tag{1}$$

SVM

- Dataset of N pairs x_i, y_i with $y_i \in \{-1, 1\}$ and $x_i \in \mathbb{R}^n$
- Classifier function $f(x) = sign(w^T x + b)$
- Solve:

$$\begin{aligned} \max_{w,b} \quad & \alpha \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq \alpha \; i = 1, \ldots, N \quad \textcolor{red}{Why?} \\ & ||w||_2 = 1 \end{aligned}$$

SVM

- Dataset of N pairs x_i, y_i with $y_i \in \{-1, 1\}$ and $x_i \in \mathbb{R}^n$
- Classifier function $f(x) = sign(w^T x + b)$
- Solve:

$$\begin{aligned} \max_{w,b} \quad & \alpha \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq \alpha \; i = 1, \ldots, N \\ & ||w||_2 = 1 \;\; \color{red}{non-convex :(} \end{aligned}$$

SVM

Let $\hat{\alpha} = \alpha||w||_2$ equivalent problem

$$\begin{aligned} \max_{w,b} \quad & \frac{\hat{\alpha}}{||w||_2} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq \hat{\alpha} \; i = 1, \ldots, N \end{aligned}$$

SVM

Let $\hat{\alpha} = \alpha ||w||_2$ equivalent problem

$$\begin{aligned} \max_{w,b} \quad & \frac{\hat{\alpha}}{||w||_2} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq \hat{\alpha} \; i = 1, \ldots, N \end{aligned}$$

w and b can be scaled arbitrarily so

$$\begin{aligned} \max_{w,b} \quad & \frac{1}{||w||_2} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \; i = 1, \ldots, N \end{aligned}$$

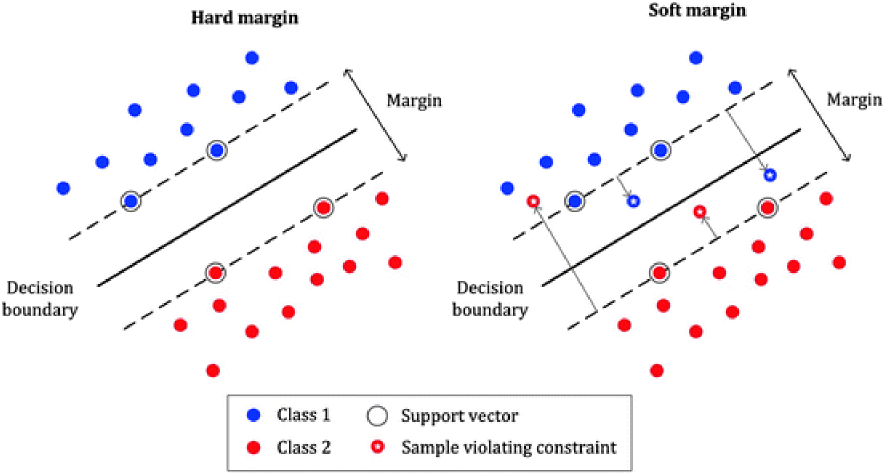
SVM

- Support vector machine optimization formulation

$$\begin{aligned} \min_{w,b} \quad & ||w||_2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \; i = 1, \ldots, N \end{aligned}$$

SVM

- If data not linear separable?



Soft Margin SVM

- Support vector machine optimization formulation

$$\begin{aligned} \min_{w,b,\xi_i} \quad & ||w||_2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ i = 1, \dots, N \end{aligned}$$

- 1-norm soft margin SVM
- Do we need to normalize our data?

SVM Pros and Cons

Pros

- Scales relatively well to high dimensions
- In practice, they tend to over-fit less

Cons

- Not suitable for large data sets
- No probabilistic explanation for the classification
- Don't perform very well, when classes overlap

Decision Trees

- SVMs are linear classifiers
- Check link in notebook for generalization of SVMs in nonlinear settings

Decision Trees

- SVMs are linear classifiers
- Check link in notebook for generalization of SVMs in nonlinear settings
- Decision trees Classification or Regression?

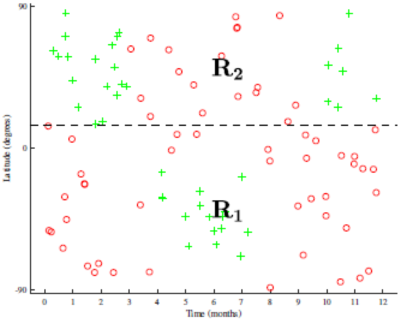
Decision Trees

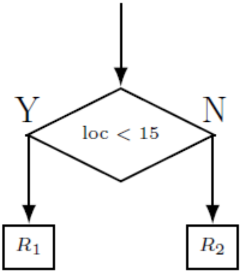
- SVMs are linear classifiers
- Check link in notebook for generalization of SVMs in nonlinear settings
- Decision trees Classification or Regression?
- Decision trees Linear or Non-linear?

Decision Trees (Example)

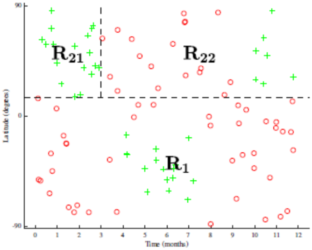
- Assume we want to predict given a time and a location whether or not skiing is possible
- Features: latitude (-90 to 90 degrees) and time of the year (month)

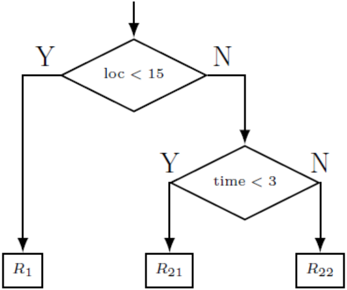
Decision Trees (Example)



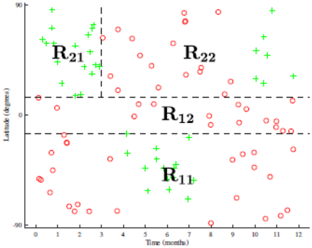


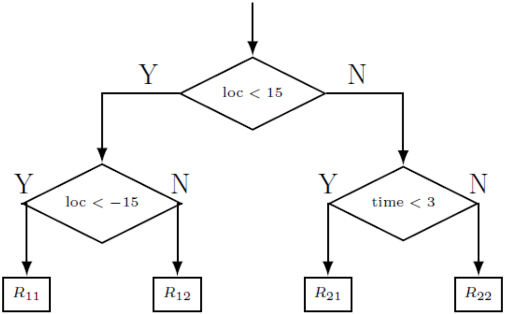
Decision Trees (Example)





Decision Trees (Example)



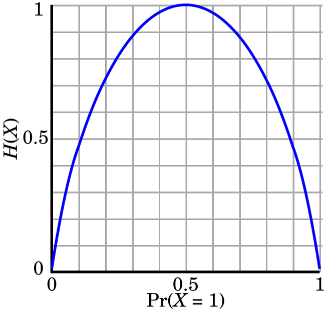


Decision Trees

- Optimal regions is an intractable problem
- Instead greedy recursive-partitioning
- How split node?
- Choose split with maximum information gain
- Feature normalization?

Decision Trees

- entropy $H = -\sum p_i log_2(p_i)$
- $H = -plog_2(p) - (1-p)log_2(1-p)$
- Information gain is the entropy of parent node minus weighted entropies of children



Information Gain (Example)

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Information Gain (Example)

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information Gain (Example)

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Decision Trees

- Possible to keep splitting until all training points are classified correctly (over-fitting)
- We want the smallest tree that explains the data

Decision Trees

- Ways to avoid over-fitting
- Minimum Leaf Size – Do not split tree if its cardinality falls below a fixed threshold.
- Maximum Depth – Do not split R if more than a fixed threshold of splits were already taken to reach R.
- Maximum Number of Nodes – Stop if a tree has more than a fixed threshold of leaf nodes

- Tree pruning
- Pruning: After whole tree is generated compute significance level for each split (eg χ^2 "measures independence" between features)
- Starting from leaves delete split that has χ^2 score less than threshold
- Even simpler, starting from leaves delete a split if that makes performance on test set increase

DT Pros and Cons

Pros

- Simple to understand and interpret
- No normalization, little data processing

Cons

- Unstable as small changes in features can lead to very different trees
- Other methods usually perform better with similar data

Random Forests

- Train decision trees by sampling from data set with replacement
- Use all these models for prediction (bagging)
- Works better with uncorrelated models
- At each split use a random subset of the features ("feature bagging")

Random Forests

Pros

- Very good predictive performance
- Reliable feature importance estimate
- Stable

Cons

- Harder to interpret than a single DT
- Computationally more challenging