CS 378: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs378/
Machine Learning

Data → Algorithm → Model

\[ f(x) \]
Announcements

FRI Survey – please take the time to respond
Announcements

“According to predicted probabilities in this study, out of every 100 students who enter college, 17 more will complete an undergraduate degree if they complete FRI. For every 100 students who graduate, 23 more will stay in a STEM major if they complete FRI.”

- just accepted paper on the benefits of FRI
Announcements

• An additional half-time (20 hrs/week) summer fellowship is available
• The award is $1,250
• Lasts 8 weeks, both start and end dates as well as hours are very flexible
• If you'd like it, email me ASAP
Announcements

Final Projects Presentation Date:
Thursday, May 12, 9:00-12:00 noon
Project Deliverables

- Final Report (6+ pages in PDF)
- Code and Documentation (posted on github)
- Presentation including video and/or demo
Project Report Structure / Outline

- Abstract
- Introduction
- Background and/or Related Work
- Technical Approach
- Experiments and/or Evaluation and/or Example Demonstration
- Conclusion and Future Work
Machine Learning
Main Reference

Alex Smola and S.V.N. Vishwanathan, 
*Introduction to Machine Learning*, 
Chapter 1, Cambridge University Press, 2008
What is Machine Learning?
Traditional Programming

Data → Computer → Output

Program → Computer

Machine Learning

Data → Computer → Program

Output → Computer
What do we mean by program?
What do we mean by program?

- A robot's controller
- A decision function (i.e., classification function)
- A neural network
- A recommendation system
- etc.
“The machine learning algorithm wants to know if we’d like a dozen wireless mice to feed the Python book we just bought.”
# Machine Learning Frameworks

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification or categorization</td>
<td>Clustering</td>
</tr>
<tr>
<td>Regression</td>
<td>Dimensionality reduction and manifold learning</td>
</tr>
</tbody>
</table>
Classification
Fig. 1.5. Binary classification; separate stars from diamonds. In this example we are able to do so by drawing a straight line which separates both sets. We will see later that this is an important example of what is called a *linear classifier*. 
Fig. 1.6. Left: binary classification. Right: 3-class classification. Note that in the latter case we have much more degree for ambiguity. For instance, being able to distinguish stars from diamonds may not suffice to identify either of them correctly, since we also need to distinguish both of them from triangles.
Classification

Inputs:

\[ \mathbf{X} := \{x_1, \ldots, x_m\} \]

where \[ x_i \in \mathbb{R}^k \]

Outputs:

\[ \mathbf{Y} := \{y_1, \ldots, y_m\} \]

set of classes:

\[ y_i \in \{1, \ldots, n\} \]
Machine Learning Framework

\[ y = f(x) \]

- **Training**: given a *training set* of labeled examples \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \), estimate the function \( f \) by minimizing the error on the training set.

- **Testing**: apply \( f \) to a never before seen *test example* \( x \) and output the predicted value \( y = f(x) \)

Slide credit: L. Lazebnik
Training and Testing Pipeline

Training

- Training Images
- Training Labels
- Classifier Training
- Trained Classifier

Testing

- Test Image
- Image Features
- Trained Classifier
- Prediction: Outdoor

Slide: Derek Hoiem
Classification using K-Nearest Neighbors

Training examples from class 1

Training examples from class 2

Slide credit: L. Lazebnik
Classification using K-Nearest Neighbors

\[ f(x) = \text{label of the training example nearest to } x \]

Test example

Training examples from class 1

Training examples from class 2

Slide credit: L. Lazebnik
1-Nearest Neighbor

![Graph showing 1-Nearest Neighbor](image-url)
3-Nearest Neighbor
Examples of distances

**Euclidean distance**

$\text{dist}(a,b) = \|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$

**Manhattan distance**

$\text{dist}(a,b) = \|a - b\|_1 = \sum_i |a_i - b_i|$

**Cosine distance**

$\text{dist}(a,b) = \cos^{-1} \left( \frac{\langle a, b \rangle}{\|a\| \|b\|} \right)$
What are some of the limitations of k-NN?
Linear Classifier

- Finds a *linear function* to separate the classes:

\[ f(x) = \text{sgn}(w \cdot x + b) \]
Algorithm 1.3 The Perceptron

Perceptron($X, Y$) \{reads stream of observations $(x_i, y_i)$\}

Initialize $w = 0$ and $b = 0$

while There exists some $(x_i, y_i)$ with $y_i(\langle w, x_i \rangle + b) \leq 0$ do

\[ w \leftarrow w + y_i x_i \] and \[ b \leftarrow b + y_i \]

end while
Dot product

\[ \mathbf{a} = (1, 4, -2) \]
\[ \mathbf{b} = (-2, 1, 7) \]
\[ \mathbf{a} \cdot \mathbf{b} = 1 \cdot (-2) + 4 \cdot 1 + (-2) \cdot 7 \]
\[ = -2 + 4 - 14 = -12 \]
Fig. 1.22. The Perceptron without bias. Left: at time $t$ we have a weight vector $w_t$ denoted by the dashed arrow with corresponding separating plane (also dashed). For reference we include the linear separator $w^*$ and its separating plane (both denoted by a solid line). As a new observation $x_t$ arrives which happens to be mis-classified by the current weight vector $w_t$ we perform an update. Also note the margin between the point $x_t$ and the separating hyperplane defined by $w^*$. Right: This leads to the weight vector $w_{t+1}$ which is more aligned with $w^*$. 
Linear Classifier

• How do we decide which line is the best?
Linear Support Vector Machine
Linear Support Vector Machine
Linear Support Vector Machine

This is called the margin.
Linear Support Vector Machine

...but what happens when the data cannot be linearly separated?

This is called the *margin*
Nonlinear Support Vector Machine

[http://www.imtech.res.in/raghava/rbpred/svm.jpg]
Nonlinear Support Vector Machine

Linearly separable:

Not linearly separable:
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \varphi(x) \rightarrow \langle x_1, x_2 \rangle \]
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \phi(x) \rightarrow \langle x_1, x_2 \rangle \]

In other words, both \( x_1 \) and \( x_2 \) need to be function of \( x \).
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \phi(x) \rightarrow \langle x, x \rangle \]

Example: both \( x_1 \) and \( x_2 \) are set to \( x \)
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\( \phi(x) \rightarrow <x,|x|> \)

Example: \( x_1 = x \) and \( x_2 = |x| \)
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \phi(x) \rightarrow \langle x_1, x_2 \rangle \]

In other words, both \(x_1\) and \(x_2\) need to be function of \(x\).
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \phi(x) \rightarrow <x, x^2> \]
Can we construct a mapping function from 1D to 2D such that the data in the 2D space is linearly separable?

\[ \varphi(x) \rightarrow \langle x, x^2 \rangle \]
Nonlinear Support Vector Machine

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(x)$, define a kernel function $K$ such that

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

(to be valid, the kernel function must satisfy *Mercer’s condition*)

- Intuitively, the kernel function should encode a measure of similarity between $x_i$ and $x_j$
Nonlinear Support Vector Machine

Consider the mapping $\varphi(x) = (x, x^2)$

\[ \varphi(x) \cdot \varphi(y) = (x, x^2) \cdot (y, y^2) = xy + x^2y^2 \]

\[ K(x, y) = xy + x^2y^2 \]
Nonlinear Support Vector Machine

- **Polynomial Kernel:**
  \[ K_{poly}(x_i, x_j) = (x_i^T x_j + 1.0)^p \]

- **Histogram kernel function:**
  \[ K_{hist}(x_i, x_j) = e^{-\rho d_{a,b}(x_i, x_j)} \]

  \[ d_{a,b}(x_i, x_j) = \sum_k |x_{ik}^a - x_{jk}^a|^b \]
Nonlinear Support Vector Machine

- linear: $K(x_i, x_j) = x_i^T x_j$.
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$.
- radial basis function (RBF): $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$, $\gamma > 0$.
- sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.
Nonlinear Support Vector Machine

- **Support Vector Machine**: a discriminative learning algorithm

1. Finds maximum margin hyperplane that separates two classes
2. Uses Kernel function to map data points into a feature space in which such a hyperplane exists

[Input Space]  [Feature Space]

[http://www.imtech.res.in/raghava/rbpred/svm.jpg]
There are many other classifiers out there...
Decision Trees

Feed-Forward Neural Networks
Deep Learning Methods

L0 (Input) 512x512
L1 256x256
L2 128x128
L3 64x64
L4 32x32
F5
F6 (Output)
Deep Learning Methods

There are many ways to combine classifiers...
Classifier Ensembles

Train Set

Sampling

$S_1 \xrightarrow{\text{train}} C_1$
$S_2 \xrightarrow{\text{train}} C_2$
$S_3 \xrightarrow{\text{train}} C_3$
$\cdots$
$S_m \xrightarrow{\text{train}} C_m$

Collection of Classifiers

Combine Outputs

Predictions
Sequences of classifiers that grows in complexity of classifier

input patterns → 1 (T) → 2 (T) → 3 (T) → further processing

- REJECTED
Concept Diagram of Stacking

Training data → classifier

Training data → Level 0 classifier

Level 0 output value → Level 1 classifier

Level 1 output value → output value

Discussion

What are some problems faced by our service robots which could benefit from a machine learning solution?

What are some common things in the environment that the robot could learn to classify?

Can a classifier be used for prediction?
Take-home message

“The decision to use machine learning is more important than the choice of a particular learning method.”

- James Hays, Brown University
Resources

• WEKA Machine Learning Library (in Java): http://www.cs.waikato.ac.nz/ml/weka/
• Support Vector Machine example using OpenCV: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
THE END