

```

38 int Move(string& board, int dir){
39     int row, col;
40     LocBlank(board, row, col);
41     if ((dir==1 || dir == -3) && row > 0){
42         Swap(board, row, col, row-1, col);
43     }
44     else if ((dir == 2 || dir == -4) && col < MAXDIM-1{
45         Swap(board, row, col, row, col+1);
46     }
47     else if ((dir == 3 || dir == -1) && row < MAXDIM-1{
48         Swap(board, row, col, row+1, col);
49     }
50     else if ((dir == 4 || dir == -2) && col > 0 ){
51         Swap(board, row, col, row, col-1);
52     }
53     else {
54         return -1;
55     }
56     return 0;

```



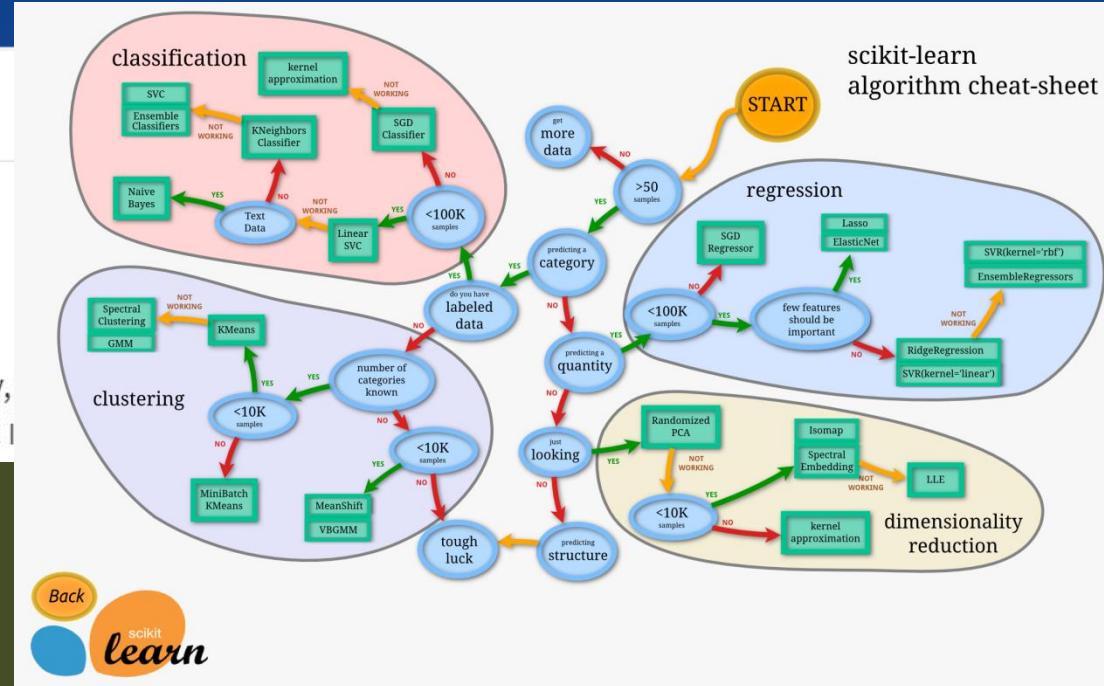
david ruby

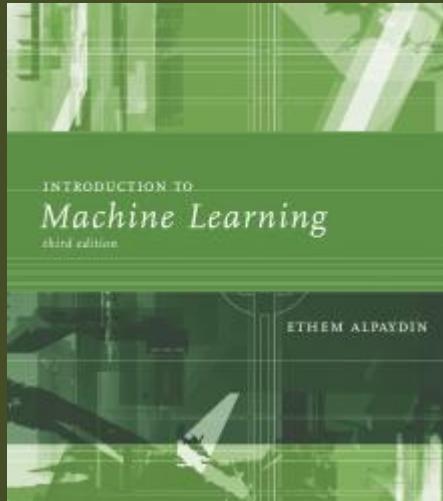
Lecturer at California State University,
California State University, Fresno • UCI

WELCOME: MACHINE LEARNING

Spring, 2018

Department of Computer Science COLLEGE OF SCIENCE AND MATHEMATICS





Primary Textbook:

INTRODUCTION TO MACHINE LEARNING

3RD EDITION

ETHEM ALPAYDIN
© The MIT Press, 2014

alpaydin@boun.edu.tr
<http://www.cmpe.boun.edu.tr/~ethem/i2ml3e>

Textbook Author: Ethem Alpaydin

Ethem Alpaydin

Ethem ALPAYDIN

[Türkçe](#)

Professor in the
[Department of Computer Engineering](#)
[Bogaziçi University](#)

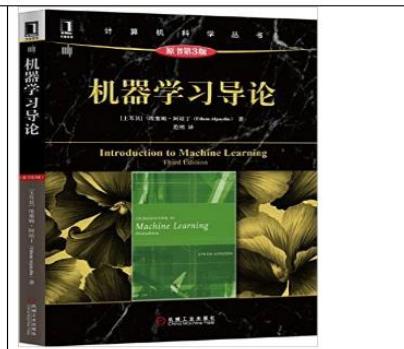
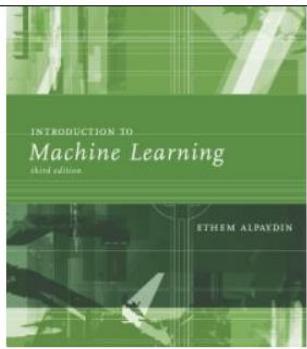
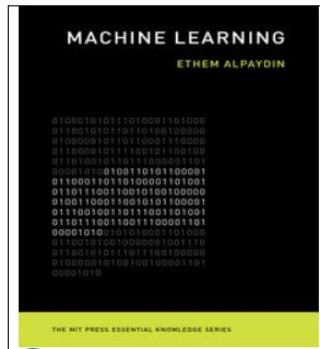
Member of [The Science Academy, Turkey](#)

[Google Scholar](#), [Semantic Scholar](#), [DBLP](#), [MS Academic Search](#), [ACM Digital Library](#), [Academia.edu](#), [ResearchGate](#), [LinkedIn](#).

[Publications](#), [Talks](#), [Other Writing](#), [Courses](#), [Short Biography](#), [Details](#)



+ Follow

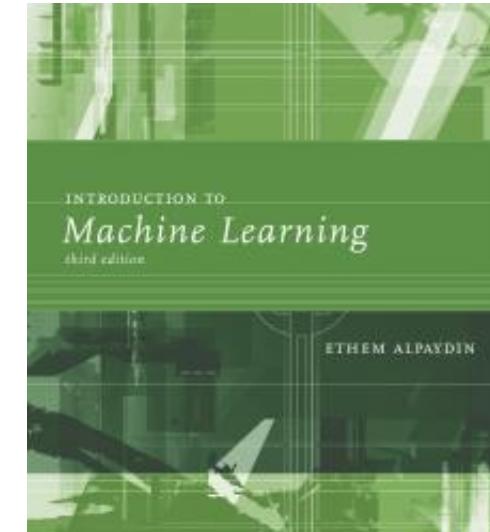


Ethem ALPAYDIN is Professor in the Department of Computer Engineering, Bogazici University, Istanbul Turkey and is a member of the Science Academy, Istanbul. He received his PhD from the Ecole Polytechnique Fédérale de Lausanne, Switzerland in 1990 and was a postdoc at the International Computer Science Institute, Berkeley in 1991. He was a Fulbright scholar in 1997. He was a visiting researcher at MIT, USA in 1994, IDIAP, Switzerland in 1998 and TU Delft, The Netherlands in 2014.

Textbook: Introduction To Machine Learning

4

- Need an Academic Roadmap
 - ▣ What do Researchers in Field currently view component keys.
- Need References to Key Underlying Concepts
 - ▣ How are these key components derived.
- Need to avoid being swamped.
 - ▣ Too much technical details prevents progress.



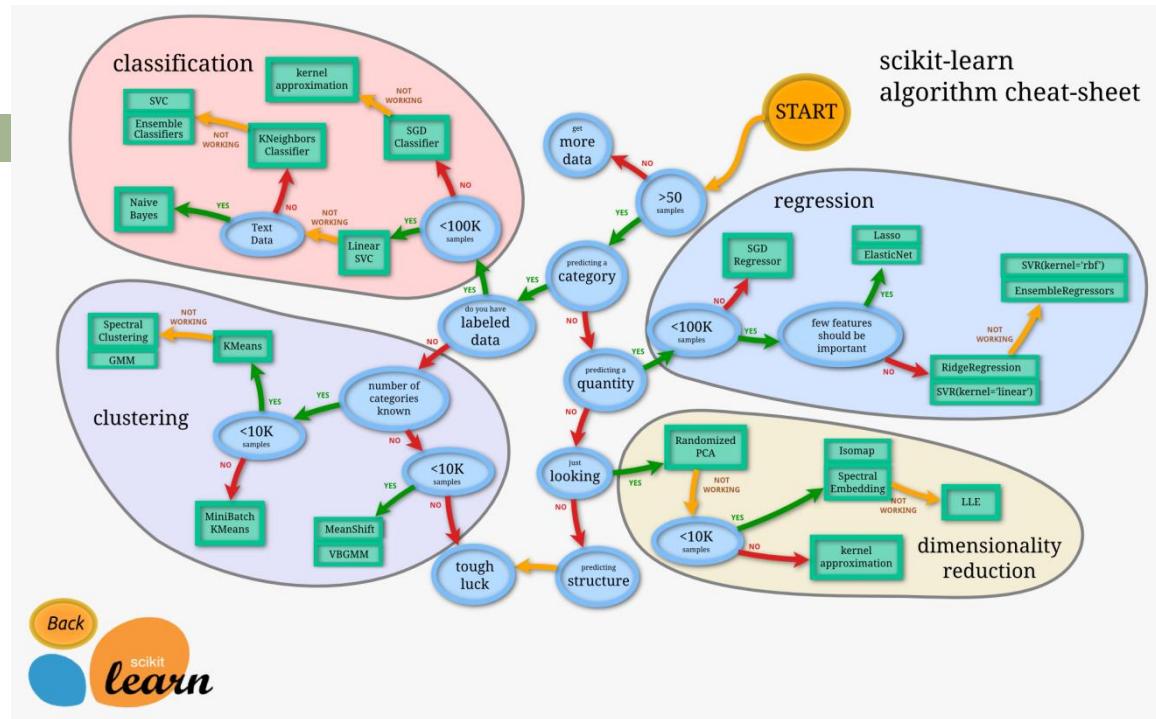
Machine Learning: Getting Going

5

- Theory alone is difficult to engage.
- New practitioners need access routes.
- Gaining Intuition needs algorithm exploration.

Machine Learning Libraries

6



- Need to combine theory with practice.
- Will be using Scikit-Learn w/ Python
- Need Help Here...!

Hands-On
Machine Learning
with Scikit-Learn
& TensorFlow

CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



Aurélien Géron

Hands-On Machine Learning with Scikit-Learn and TensorFlow

Concepts, Tools, and Techniques to Build Intelligent Systems

By [Aurélien Géron](#)

Publisher: [O'Reilly Media](#)

Release Date: March 2017

Pages: 576

Graphics in this book are printed in black and white.

Through a series of recent breakthroughs, deep learning has boosted the entire field of machine learning. Now, even programmers who know close to nothing about this technology can use simple, efficient tools to implement programs capable of learning from data. This practical book shows you how.

By using concrete examples, minimal theory, and two production-ready Python frameworks—scikit-learn and TensorFlow—author Aurélien Géron helps you gain an intuitive understanding of the concepts and tools for building intelligent systems. You'll learn a range of techniques, starting with simple linear regression and progressing to deep neural networks. With exercises in each chapter to help you apply what you've learned, all you need is programming experience to get started.

- Explore the machine learning landscape, particularly neural nets
- Use scikit-learn to track an example machine-learning project end-to-end
- Explore several training models, including support vector machines, decision trees, random forests, and ensemble methods
- Use the TensorFlow library to build and train neural nets
- Dive into neural net architectures, including convolutional nets, recurrent nets, and deep reinforcement learning
- Learn techniques for training and scaling deep neural nets
- Apply practical code examples without acquiring excessive machine learning theory or algorithm details

SECOND BOOK

Practical Handbook

Second Book: Author: Aurélien Geron

8



Aurélien Geron
ageron

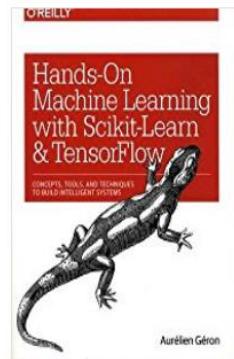
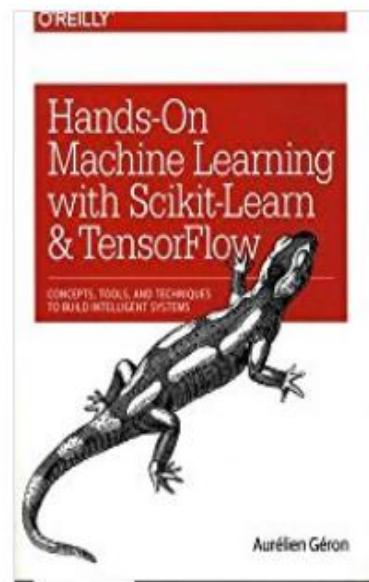
Machine Learning consultant, former PM of YouTube video classification and founder & CTO of telco operator

Follow

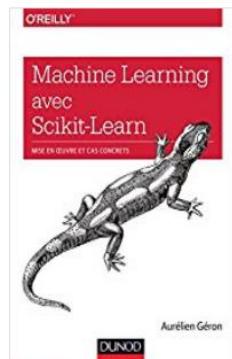
Aurélien Géron



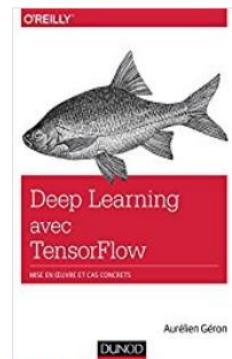
+ Follow



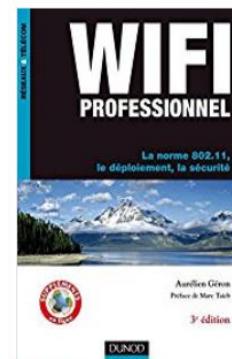
£23.94
Paperback



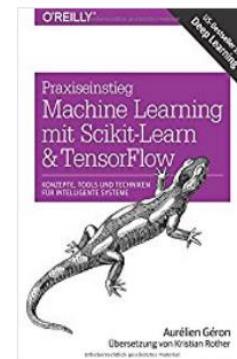
£27.99
Kindle Edition



£31.99
Kindle Edition



£32.99
Kindle Edition



Aurélien Géron
Übersetzung von Kristian Rother
Paperback

Hands-On Machine Learning...

9

- We'll walk through second book along with theory material from Textbook.

Machine Learning Notebooks

This project aims at teaching you the fundamentals of Machine Learning in python. It contains the example code and solutions to the exercises in my O'Reilly book [Hands-on Machine Learning with Scikit-Learn and TensorFlow](#):



Simply open the [Jupyter](#) notebooks you are interested in:

- Using [jupyter.org's notebook viewer](#)
 - note: [github.com's notebook viewer](#) also works but it is slower and the math formulas are not displayed correctly,
- or by cloning this repository and running Jupyter locally. This option lets you play around with the code. In this case, follow the installation instructions below.

Machine Learning Practice

11

- Utilize Teams!
- Classes require computer
 - ▣ Computers used for quizzes and tests.
- Follow second book Python Configuration
- Explore Machine Learning with Python in Jupyter Notebooks on GitHub.



Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

 Pick a username

Email

 you@example.com

Password

 Create a password

Use at least one letter, one numeral, and seven characters.

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

GitHub

13

- Everyone will use/create GitHub accounts.
- Teams will create Jupyter Notebooks to illustrate Machine Learning in Action.
- Datasets with Scikit Learn
- Datasets with Kaggle

Kaggle

14

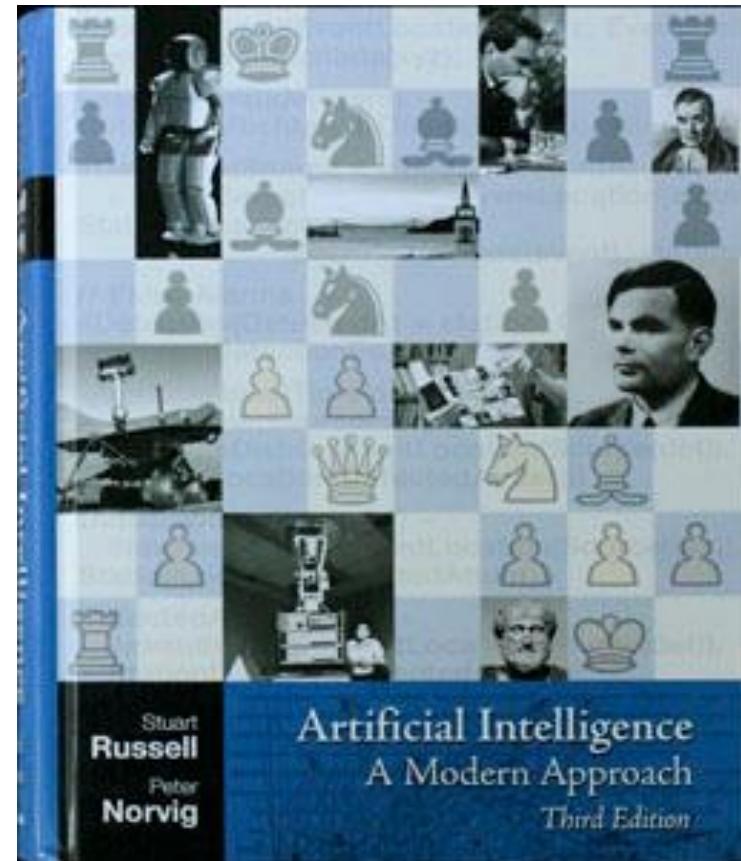
- Great resource for data science community

The screenshot shows the top navigation bar of the Kaggle website, featuring the 'kaggle' logo, a search bar, and links for Competitions, Datasets, Kernels, Discussion, Jobs, and Sign In. Below the header, the main title 'The Home of Data Science & Machine Learning' is displayed in large white text. A subtitle 'Kaggle helps you learn, work, and play' follows. To the right, there is a cartoon illustration of an astronaut in space with the text 'jobs board >' above it. At the bottom left, there is a 'What's Kaggle?' illustration showing various icons related to data science and machine learning.

Additional Textbook:

15

- Artificial Intelligence: A Modern Approach
 - Stuart Russell & Peter Norvig



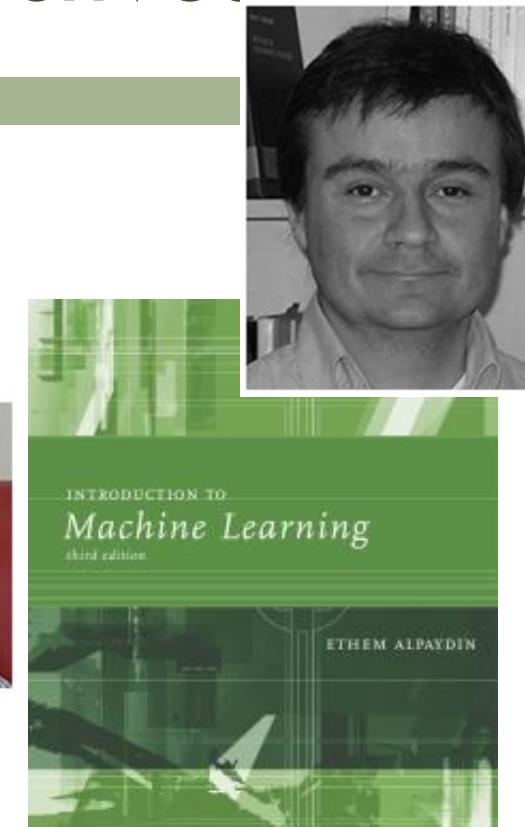
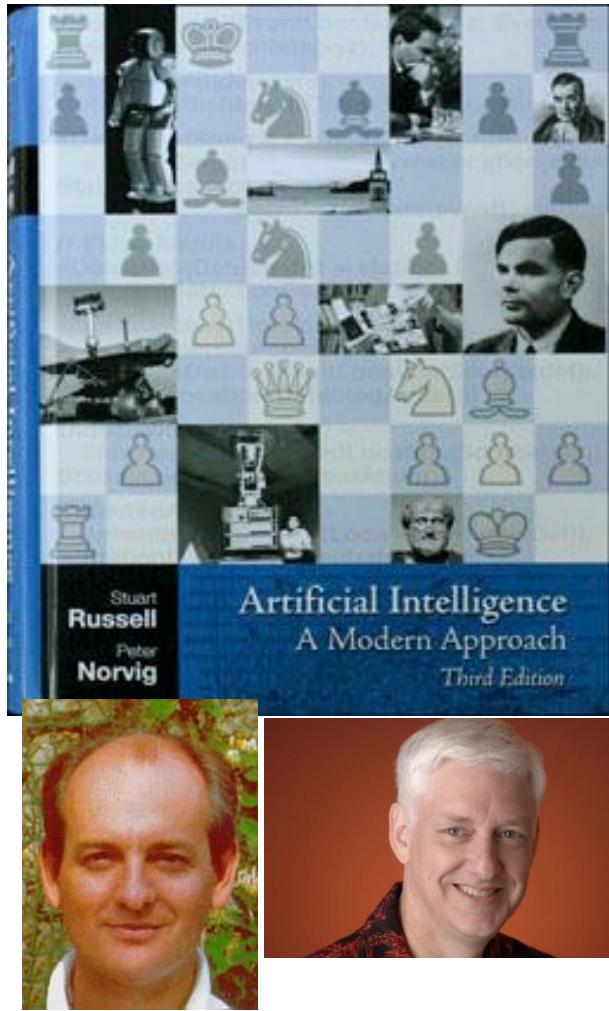
Artificial Intelligence: A Modern Approach

16

- Everyone should have a copy from 164
- Chapter 18: Machine Learning
- Chapter 20: Learning Probabilistic Models
- Chapter 21: Reinforcement Learning

3 Sources --- 3 Perspectives

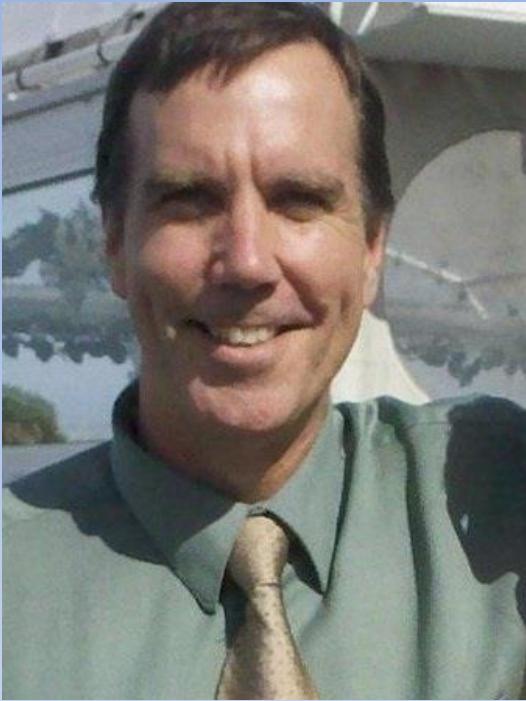
17



Grading:

18

- Participation
 - Teams Formed
 - 1 team member must attend each class
 - Team reports in class
- Assignments
 - Jupyter Notebooks
 - Peer Reviews
- Quizzes
 - In Class – Computer Based
- Midterm
 - In Class – Computer Based
- Final
 - In Class – Computer Based



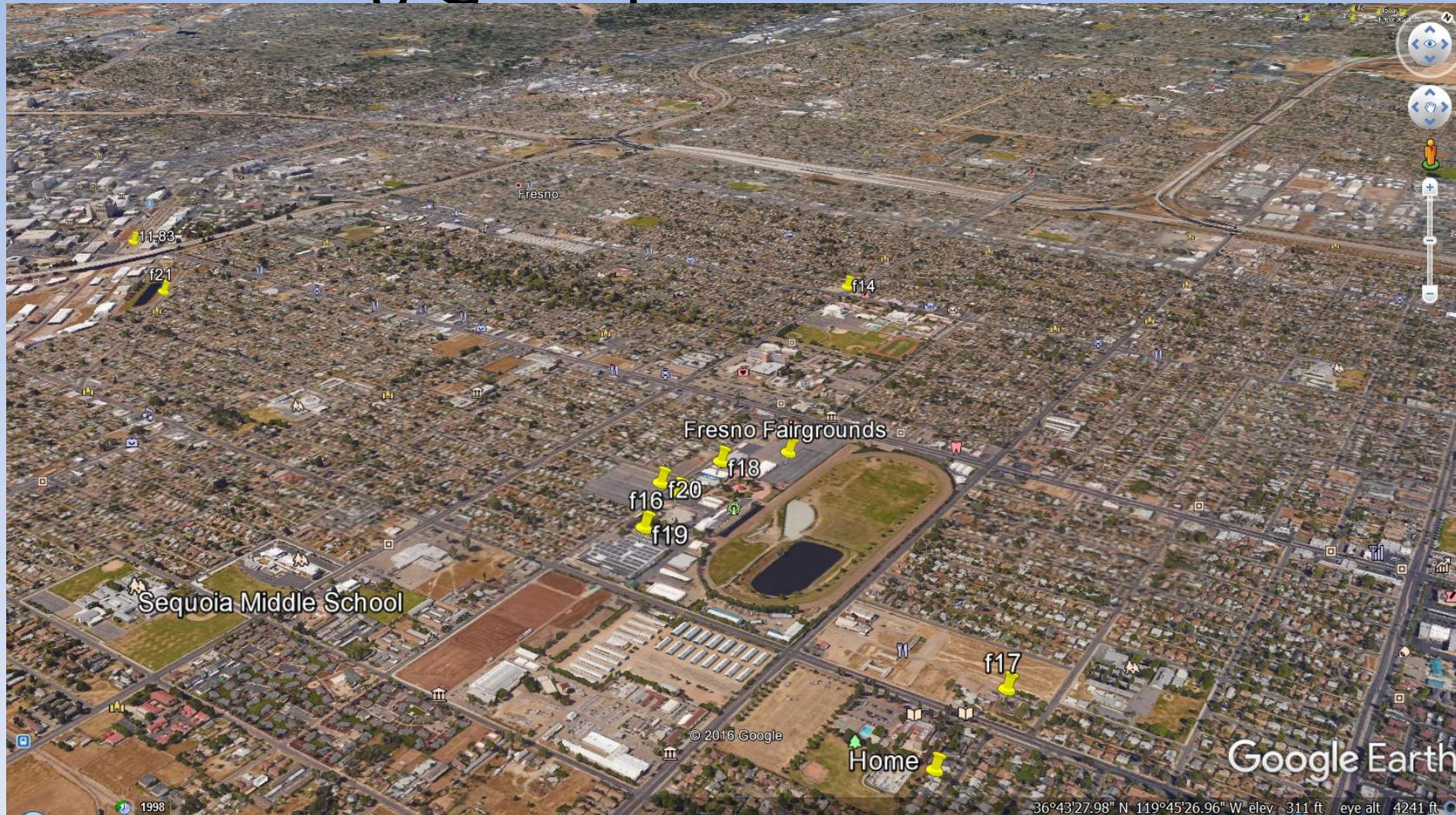
David Ruby

Class Instructor

- Office
 - Science II – 273
- Email:
 - druby@csufresno.edu

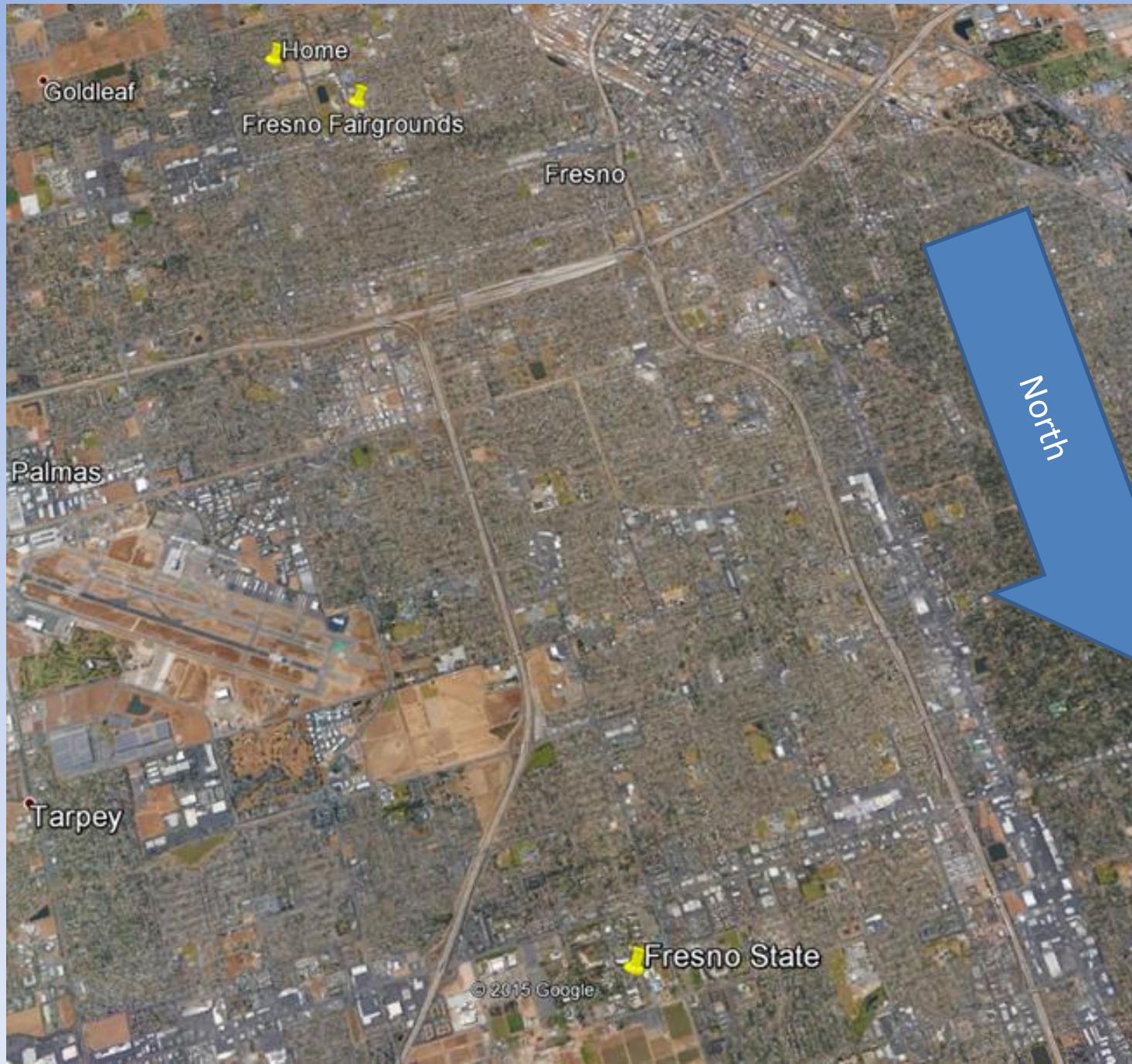
- First-Generation College Student
- How PhD?

Father Floyd Fresno Career Custodian Ending @ Sequoia Middle School



Google Earth

36°43'27.98" N 119°45'26.96" W elev 311 ft eye alt 4241 ft



Compute Science Focus: Jobs/Degrees

- Students want...
 - Jobs!
 - Advanced Degrees!!
- How???

Dr. Joy Goto, Professor Biochemistry, Fresno State

- https://youtu.be/FXUiEPrK_II

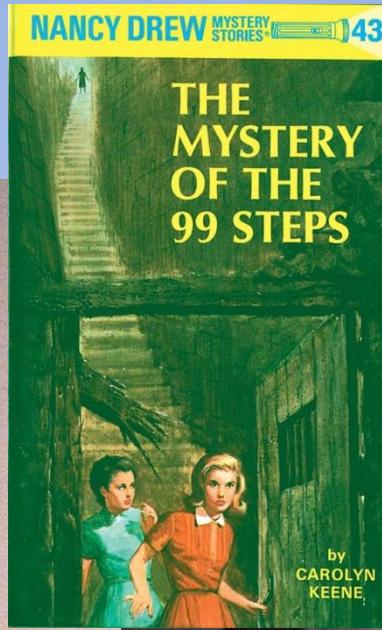
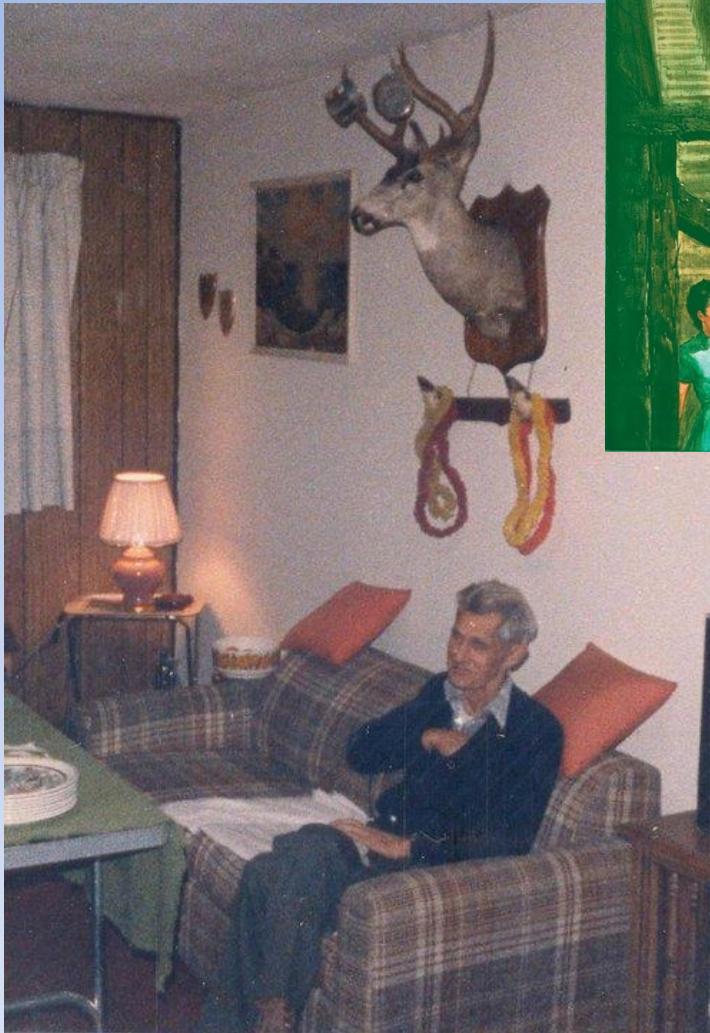


Dr. Joy Goto, Professor Biochemistry, Fresno State



- https://youtu.be/FXUiEPrK_II
- Engaging story of discovering joy in science growing up here in the central valley.

My Story



- Family Memories



Interest
In
Puzzles

Memories.. eXciting Puzzles !

- Home Hedges Maze Crawwwwwl !
- Also – First time w/ Sliding Tile Puzzle



Start State

1	2	3
4		6
7	5	8



1	2	3
4	5	6
7		8



1	2	3
4	5	6
7	8	

Goal State

Thesis: Tile-Sliding Puzzle



Artificial Intelligence

Branch: master ▾ [ics1293](#) / [Papers](#) / [Ruby.AAAI91-082.pdf](#) [Find file](#) [Copy path](#)

 **everestso** paper 017b894 4 days ago

1 contributor

1000 KB [Raw](#) [History](#)  

From: AAAI-91 Proceedings. Copyright ©1991, AAAI (www.aaai.org). All rights reserved.

SteppingStone: An Empirical and Analytical Evaluation*

David Ruby and Dennis Kibler
Department of Information & Computer Science
University of California, Irvine
Irvine, CA 92717 U.S.A.
druby@ics.uci.edu

Abstract

Decomposing a difficult problem into simpler subproblems is a classic problem solving technique. Unfortunately, the most difficult subproblems can be as difficult, if not more difficult, than the original problem. This is not an obstacle to problem solving if the difficult subproblems recur in other problems. If the difficult subproblems recur often, then its solution need only be learned once and reused. SteppingStone is a learning problem solver that decomposes a problem into simple and difficult-but-recurring subproblems. It solves the

SteppingStone operates on problems defined with a state space representation consisting of a set of goals, a set of operators, and an initial state. The goal orderer takes as input a set of goals. It orders these goals so that the constrained search method will likely solve them. It does this by ordering them so as to reduce the likelihood of subgoal interactions using a domain independent heuristic we call *openness* [Ruby and Kibler, 1989]. It produces an ordered set of subgoals as output.

The constrained search component takes as input an

My Idea...

Memories are constructed.. Not stored complete!

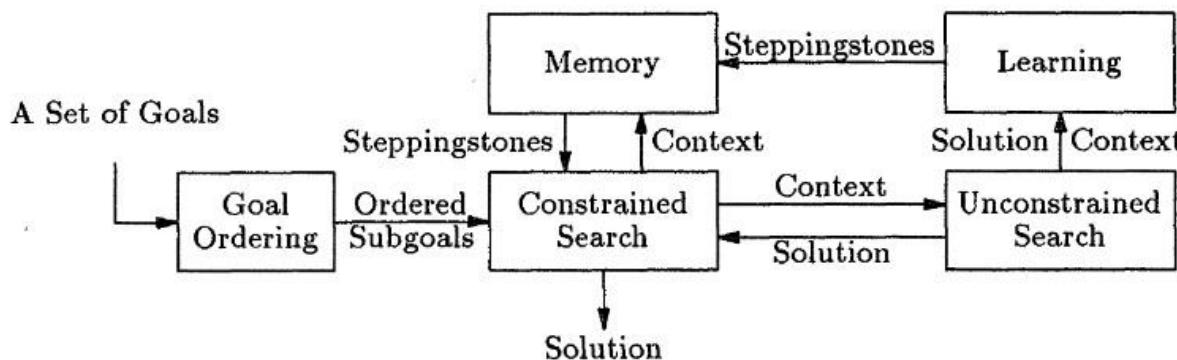


Figure 1: Overview of SteppingStone

the original impasse state.

When memory fails to return any useful steppingstones the constrained search component calls the unconstrained search component. The unconstrained search component takes as input a context, just as the memory component did. Unconstrained search relaxes the protection on the solved subgoals in its search for a solution. If it resolves the impasse, it returns the sequence of moves found to the constrained search component. The unconstrained search component also sends its impasse solution, along with the context, to the learner.

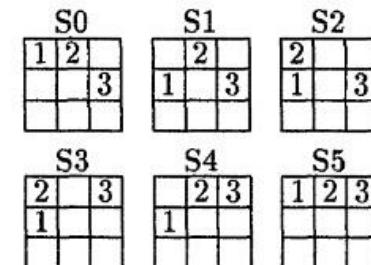
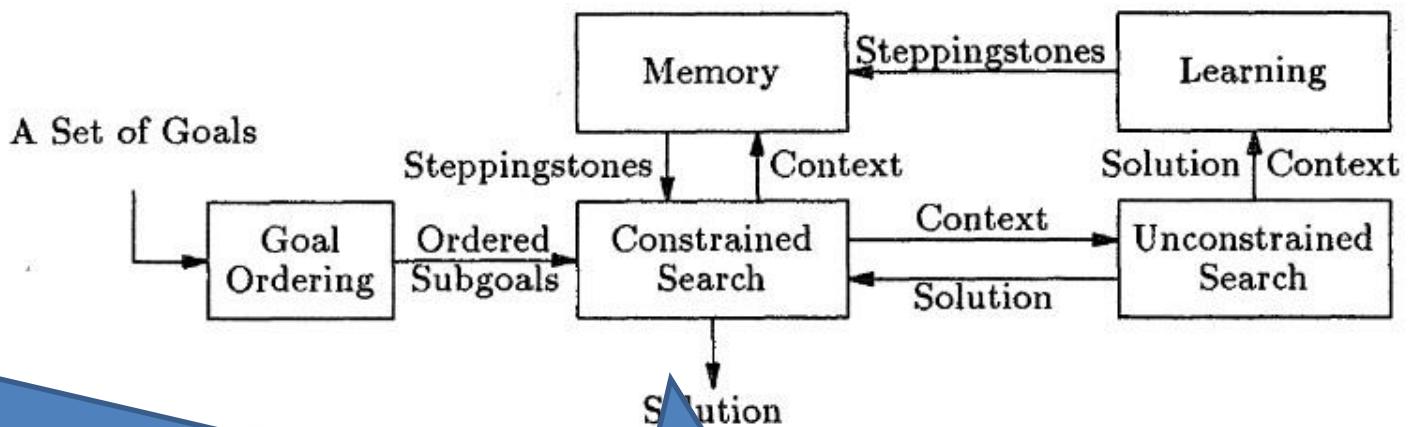


Figure 2: Steppingstones from Memory

Thesis: Tile Sliding Domain



1: Overview of SteppingStone

When stones the constrained search component takes as input a context from the memory component did. Unconstrained search removes the protection on the solved subgoals in its search for a solution. If it resolves the impasse, it returns the sequence of moves found to the constrained search component. The unconstrained search component also sends its impasse solution, along with the context, to the learner.

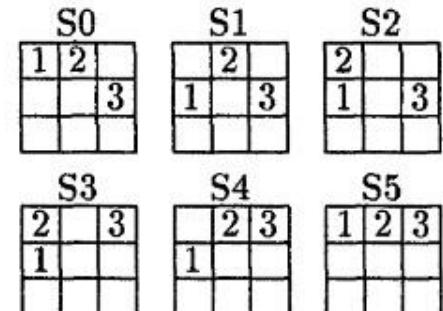
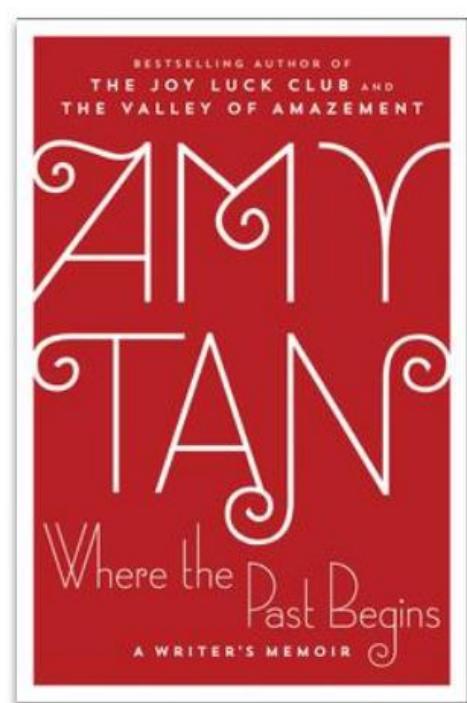
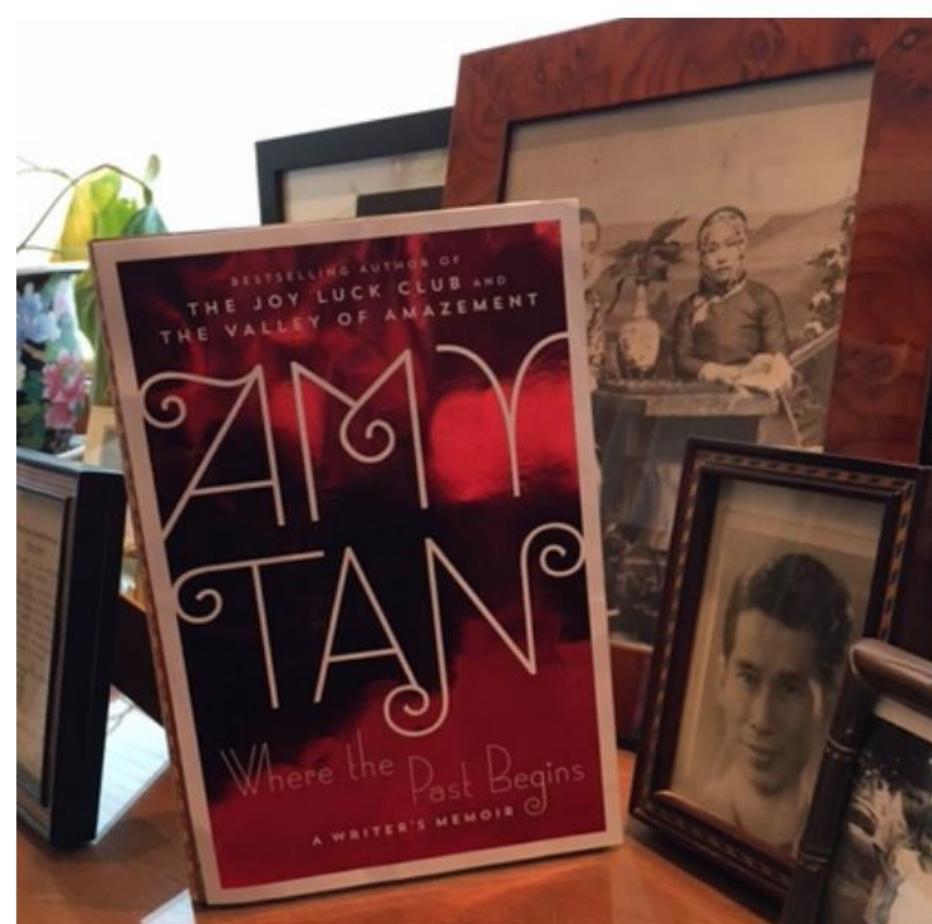


Figure 2: Steppingstones from Memory

Emotional Memory: Process of a Writer



coming
Oct 17, 2017
[Pre-order](#)

Malleable Memory (Gaps)

Learning & Memory w/ Elizabeth Loftus



CORRUPTED MEMORY

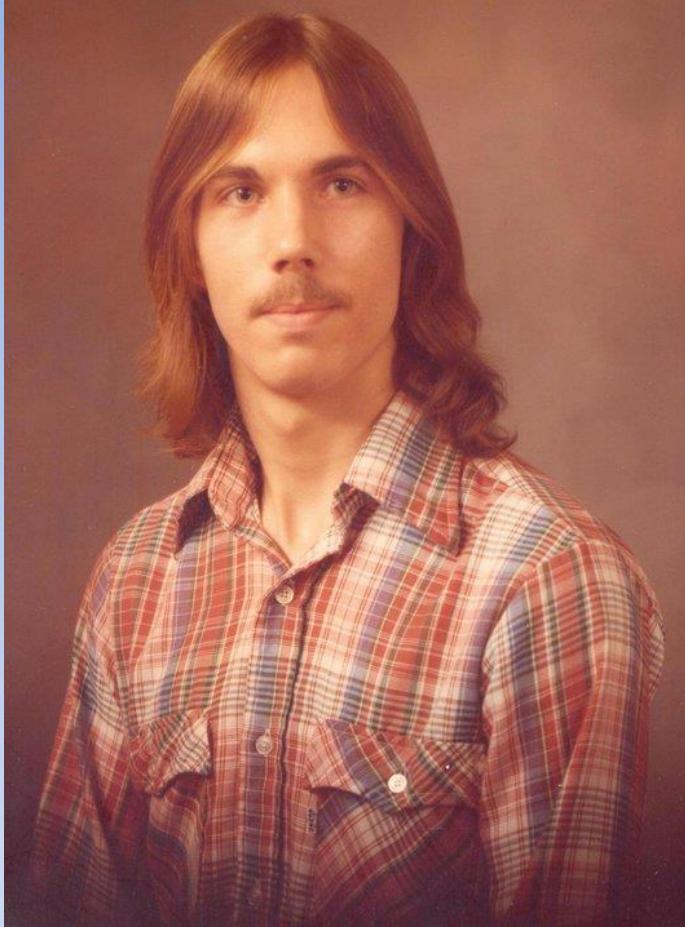
Elizabeth Loftus has spent decades exposing flaws in eyewitness testimony. Her ideas are gaining fresh traction in the US legal system.

BY MOHEB COSTANDI



Elizabeth Loftus is a cognitive psychologist at the University of California Irvine.

eXciting Mazes Memories!



- ME:
 - Do you remember the FUN maze?
- NEIGHBOR:
 - Do YOU remember this other HORRIBLE thing??
- ME:
 - Hmm .. I guess not.
- Language influencing memory ??

Memories & Learning

JOURNAL OF VERBAL LEARNING AND VERBAL BEHAVIOR 13, 585-589 (1974)

Reconstruction of Automobile Destruction: An Example of the Interaction Between Language and Memory'

ELIZABETH F. LOFTUS AND JOHN C. PALMER

University of Washington

Two experiments are reported in which subjects viewed films of automobile accidents and then answered questions about events occurring in the films. The question, "About how fast were the cars going when they smashed into each other?" elicited higher estimates of speed than questions which used the verbs *collided*, *bumped*, *contacted*, or *hit* in place of *smashed*. On a retest one week later, those subjects who received the verb *smashed* were more likely to say "yes" to the question, "Did you see any broken glass?", even though broken glass was not present in the film. These results are consistent with the view that the questions asked subsequent to an event can cause a reconstruction in one's memory of that event.

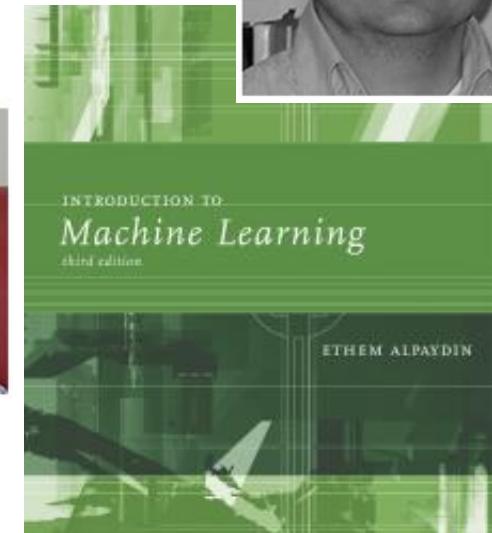
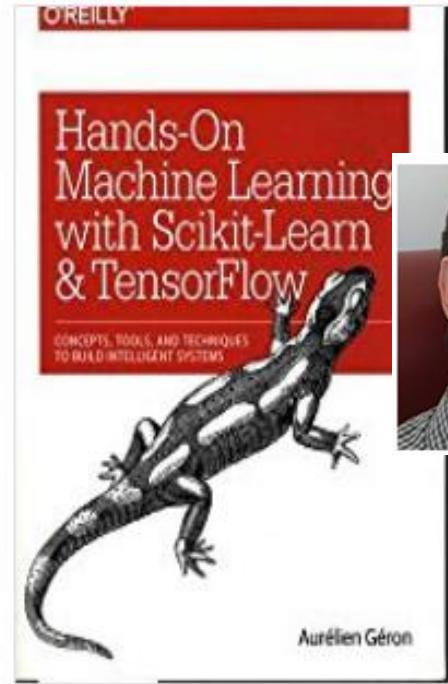
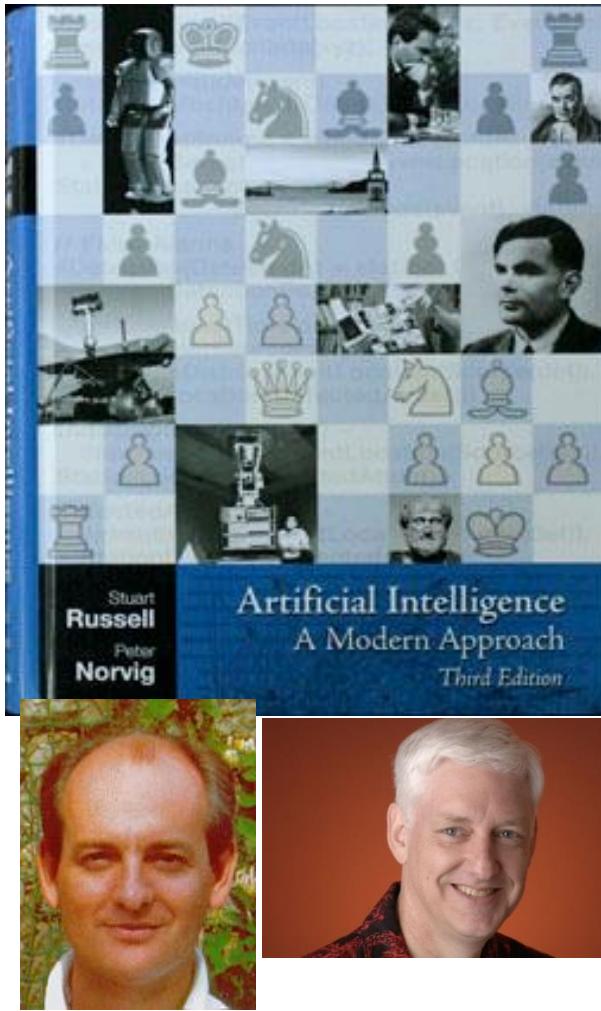
Computer Science / Memories

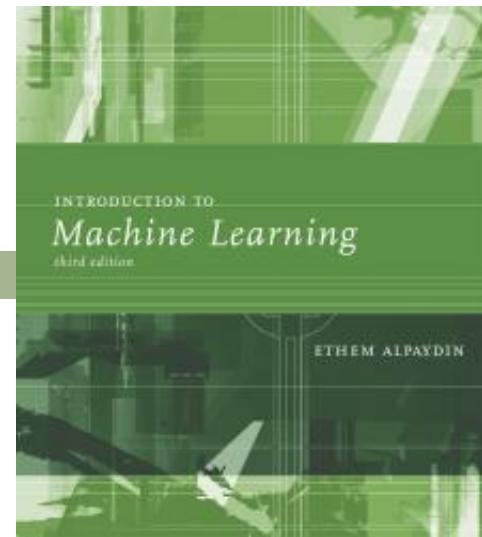
- Puzzles
- Abstractions
- Memories

3 Sources --- 3 Perspectives

Chapter Outlines

36



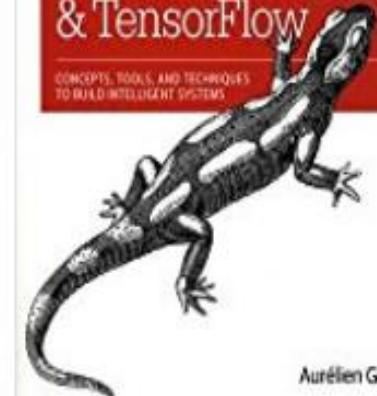


1 *Introduction* 1

1.1	What Is Machine Learning?	1
1.2	Examples of Machine Learning Applications	4
1.2.1	Learning Associations	4
1.2.2	Classification	5
1.2.3	Regression	9
1.2.4	Unsupervised Learning	11
1.2.5	Reinforcement Learning	13
1.3	Notes	14
1.4	Relevant Resources	17
1.5	Exercises	18
1.6	References	20

Hands-On Machine Learning with Scikit-Learn & TensorFlow

CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



Aurélien Géron

1. The Machine Learning Landscape

What Is Machine Learning?

Why Use Machine Learning?

Types of Machine Learning Systems

- Supervised/Unsupervised Learning

- Batch and Online Learning

- Instance-Based Versus Model-Based Learning

Main Challenges of Machine Learning

- Insufficient Quantity of Training Data

- Nonrepresentative Training Data

- Poor-Quality Data

- Irrelevant Features

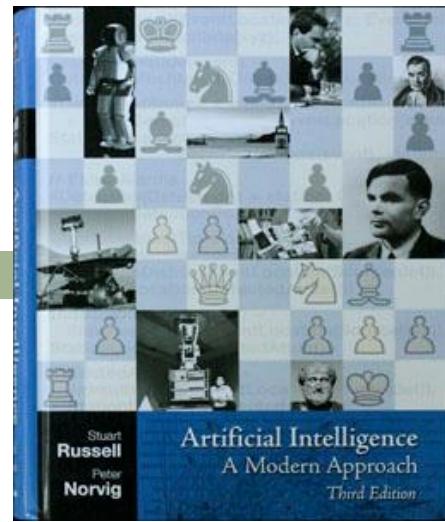
- Overfitting the Training Data

- Underfitting the Training Data

- Stepping Back

Testing and Validating

Exercises



18 Learning from Examples	693
18.1 Forms of Learning	693
18.2 Supervised Learning	695
18.3 Learning Decision Trees	697
18.4 Evaluating and Choosing the Best Hypothesis	708
18.5 The Theory of Learning	713
18.6 Regression and Classification with Linear Models	717
18.7 Artificial Neural Networks	727
18.8 Nonparametric Models	737
18.9 Support Vector Machines	744
18.10 Ensemble Learning	748
18.11 Practical Machine Learning	753
18.12 Summary, Bibliographical and Historical Notes, Exercises	757

2. End-to-End Machine Learning Project

Working with Real Data

Look at the Big Picture

Frame the Problem

Select a Performance Measure

Check the Assumptions

Get the Data

Create the Workspace

Download the Data

Take a Quick Look at the Data Structure

Create a Test Set

Discover and Visualize the Data to Gain Insights

Visualizing Geographical Data

Looking for Correlations

Experimenting with Attribute Combinations

Prepare the Data for Machine Learning Algorithms

Data Cleaning

Handling Text and Categorical Attributes

Custom Transformers

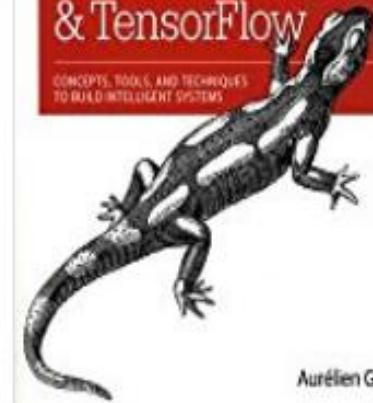
Feature Scaling

Transformation Pipelines

O'REILLY

Hands-On Machine Learning with Scikit-Learn & TensorFlow

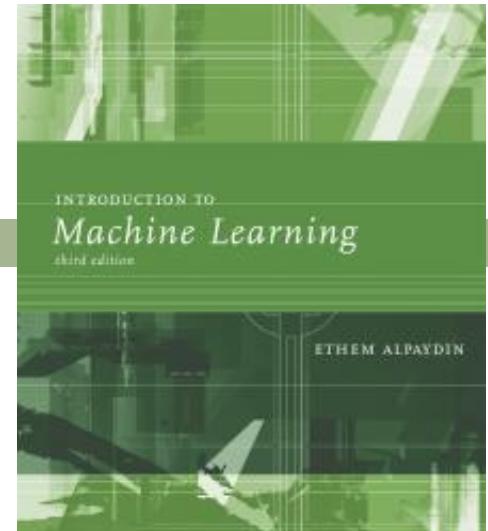
CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



Aurélien Géron

2 *Supervised Learning* 21

2.1	Learning a Class from Examples	21
2.2	Vapnik-Chervonenkis Dimension	27
2.3	Probably Approximately Correct Learning	29
2.4	Noise	30
2.5	Learning Multiple Classes	32
2.6	Regression	34
2.7	Model Selection and Generalization	37
2.8	Dimensions of a Supervised Machine Learning Algorithm	41
2.9	Notes	42



3. Classification

MNIST

Training a Binary Classifier

Performance Measures

Measuring Accuracy Using Cross-Validation

Confusion Matrix

Precision and Recall

Precision/Recall Tradeoff

The ROC Curve

Multiclass Classification

Error Analysis

Multilabel Classification

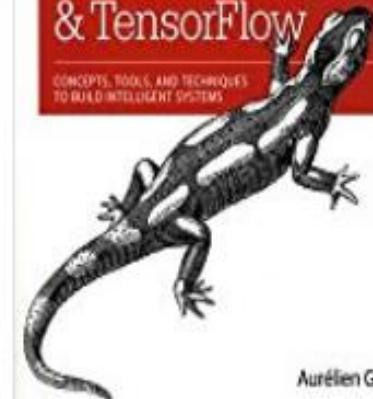
Multioutput Classification

Exercises

O'REILLY

Hands-On
Machine Learning
with Scikit-Learn
& TensorFlow

CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



Aurélien Géron

4. Training Models

Linear Regression

The Normal Equation

Computational Complexity

Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

Mini-batch Gradient Descent

Polynomial Regression

Learning Curves

Regularized Linear Models

Ridge Regression

Lasso Regression

Elastic Net

Early Stopping

Logistic Regression

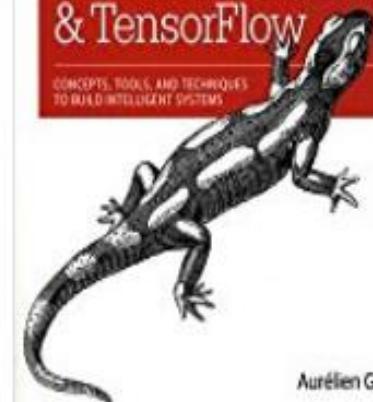
Estimating Probabilities

Training and Cost Function

O'REILLY

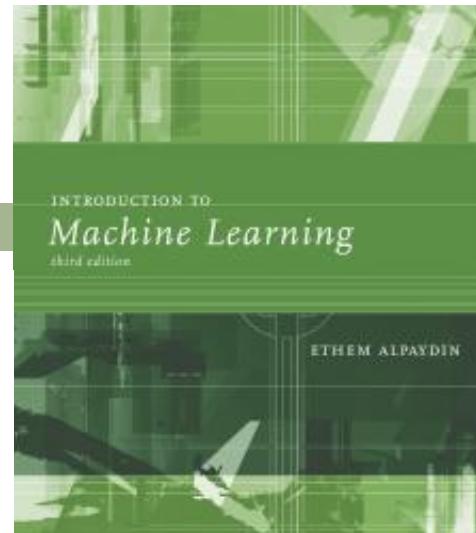
Hands-On
Machine Learning
with Scikit-Learn
& TensorFlow

CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



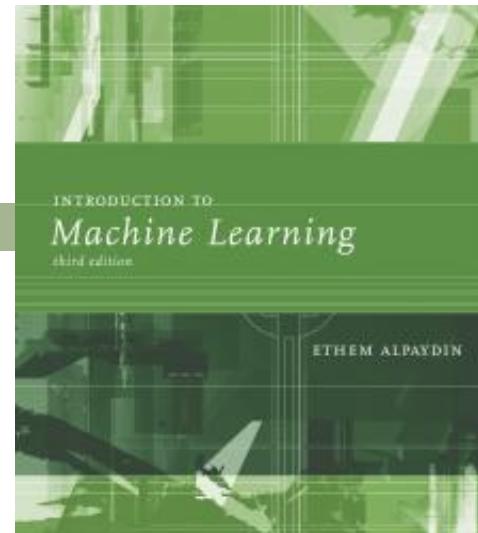
Aurélien Géron

19 Design and Analysis of Machine Learning Experiments	547
19.1 Introduction	547
19.2 Factors, Response, and Strategy of Experimentation	550
19.3 Response Surface Design	553
19.4 Randomization, Replication, and Blocking	554
19.5 Guidelines for Machine Learning Experiments	555
19.6 Cross-Validation and Resampling Methods	558
19.6.1 K-Fold Cross-Validation	559
19.6.2 5 × 2 Cross-Validation	560
19.6.3 Bootstrapping	561
19.7 Measuring Classifier Performance	561
19.8 Interval Estimation	564
19.9 Hypothesis Testing	568
19.10 Assessing a Classification Algorithm's Performance	570
19.10.1 Binomial Test	571
19.10.2 Approximate Normal Test	572
19.10.3 <i>t</i> Test	572
19.11 Comparing Two Classification Algorithms	573
19.11.1 McNemar's Test	573



9 *Decision Trees* 213

- 9.1 Introduction 213
- 9.2 Univariate Trees 215
 - 9.2.1 Classification Trees 216
 - 9.2.2 Regression Trees 220
- 9.3 Pruning 222
- 9.4 Rule Extraction from Trees 225
- 9.5 Learning Rules from Data 226
- 9.6 Multivariate Trees 230
- 9.7 Notes 232
- 9.8 Exercises 235
- 9.9 References 237



Chapter 6 Decision Trees

Training and Visualizing a Decision Tree

Making Predictions

Estimating Class Probabilities

The CART Training Algorithm

Computational Complexity

Gini Impurity or Entropy?

Regularization Hyperparameters

Regression

Chapter 6. Decision Trees

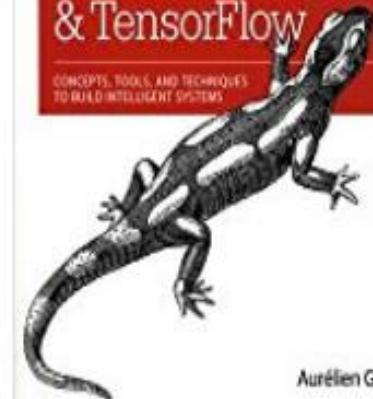
Like SVMs, *Decision Trees* are versatile Machine Learning algorithms that can perform both classification and regression tasks, and even multioutput tasks. They are very powerful algorithms, capable of fitting complex datasets. For example, in [Chapter 2](#) you trained a `DecisionTreeRegressor` model on the California housing dataset, fitting it perfectly (actually overfitting it).

Decision Trees are also the fundamental components of Random Forests (see [Chapter 7](#)), which are among the most powerful Machine Learning algorithms available today.

In this chapter we will start by discussing how to train, visualize, and make predictions with Decision Trees. Then we will go through the CART training algorithm used by Scikit-Learn, and we will discuss how to regularize trees and use them for regression tasks. Finally, we will discuss some of the limitations of Decision Trees.

Hands-On
Machine Learning
with Scikit-Learn
& TensorFlow

CONCEPTS, TOOLS, AND TECHNIQUES
TO BUILD INTELLIGENT SYSTEMS



Aurélien Géron

Titanic

47



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics



Kaggle · 9,693 teams · 2 years to go

[Overview](#)

[Data](#)

[Kernels](#)

[Discussion](#)

[Leaderboard](#)

[Rules](#)

[Team](#)

[My Submissions](#)

[Submit Predictions](#)

Overview

[Description](#)

[Evaluation](#)

[Frequently Asked Questions](#)

[Tutorials](#)

Start here if...

You're new to data science and machine learning, or looking for a simple intro to the Kaggle prediction competitions.

Competition Description

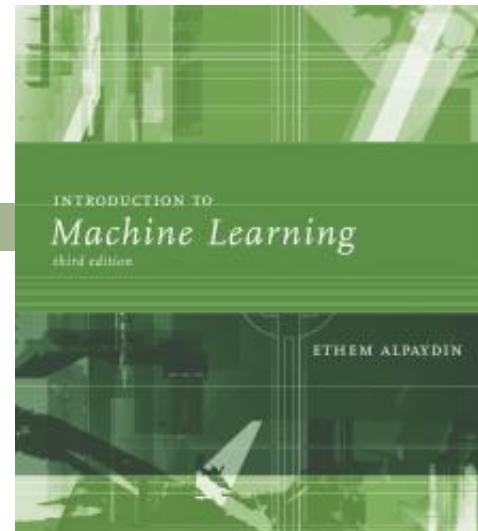
The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

Midterm

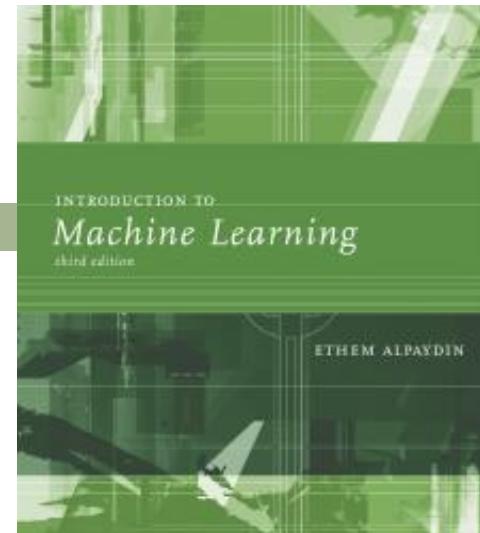
3 *Bayesian Decision Theory* 49

3.1	Introduction	49
3.2	Classification	51
3.3	Losses and Risks	53
3.4	Discriminant Functions	55
3.5	Association Rules	56
3.6	Notes	59
3.7	Exercises	60
3.8	References	64



6 Dimensionality Reduction 115

- 6.1 Introduction 115
- 6.2 Subset Selection 116
- 6.3 Principal Component Analysis 120
- 6.4 Feature Embedding 127
- 6.5 Factor Analysis 130
- 6.6 Singular Value Decomposition and Matrix Factorization 135
- 6.7 Multidimensional Scaling 136
- 6.8 Linear Discriminant Analysis 140
- 6.9 Canonical Correlation Analysis 145
- 6.10 Isomap 148
- 6.11 Locally Linear Embedding 150
- 6.12 Laplacian Eigenmaps 153
- 6.13 Notes 155
- 6.14 Exercises 157
- 6.15 References 158



Chapter 8 Dimensionality Reduction

The Curse of Dimensionality

Main Approaches for Dimensionality Reduction

PCA

Kernel PCA

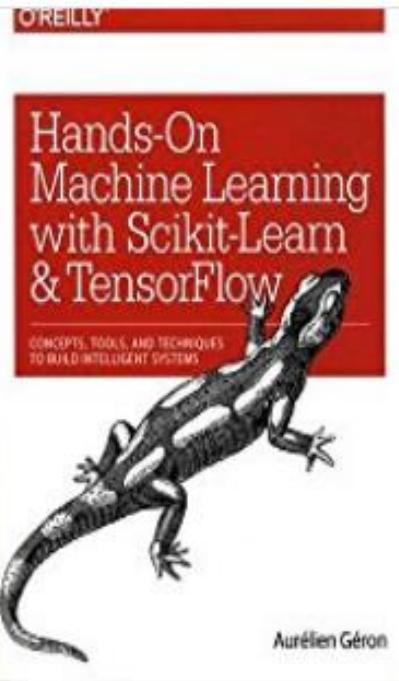
LLE

Other Dimensionality Reduction Techniques

Exercises

Chapter 8. Dimensionality Reduction

Many Machine Learning problems involve thousands or even millions of features for each training instance. Not only does this make training extremely slow, it can also make it much harder to find a good solution, as we will see. This problem is often referred to as the *curse of dimensionality*.



Aurélien Géron

Kaggle MNIST

52

Secure | <https://www.kaggle.com/c/digit-recognizer>

myfsu M email OneDrive V VitalSource mit Nitrous GitHub zy zyBooks f Stanford Lagunita csuf AOL aol LinkedIn backup

kaggle Search kaggle Competitions Datasets Kernels Discussion Jobs ... Sign In

Digit Recognizer

Learn computer vision fundamentals with the famous MNIST data
2,015 teams · 2 years to go

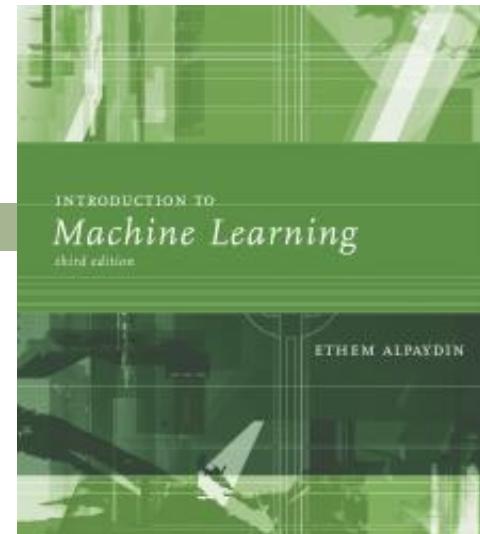
Overview Data Kernels Discussion Leaderboard Rules

Overview

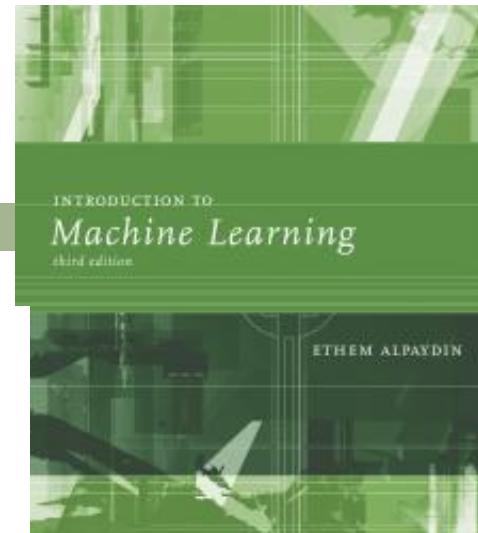
Description	Start here if... You have some experience with R or Python and machine learning basics, but you're new to computer vision. This competition is the perfect introduction to techniques like neural networks using a classic dataset including pre-extracted features.
Evaluation	
Tutorial	

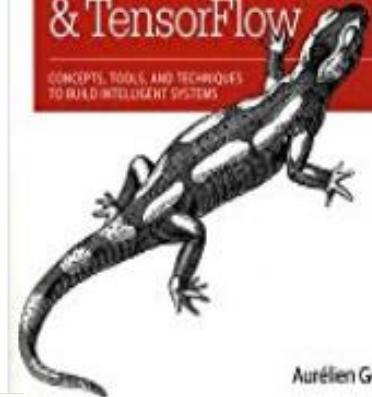
4 *Parametric Methods* 65

4.1	Introduction	65
4.2	Maximum Likelihood Estimation	66
4.2.1	Bernoulli Density	67
4.2.2	Multinomial Density	68
4.2.3	Gaussian (Normal) Density	68
4.3	Evaluating an Estimator: Bias and Variance	69
4.4	The Bayes' Estimator	70
4.5	Parametric Classification	73
4.6	Regression	77
4.7	Tuning Model Complexity: Bias/Variance Dilemma	80
4.8	Model Selection Procedures	83
4.9	Notes	87
4.10	Exercises	88
4.11	References	90



5 Multivariate Methods	93
5.1 Multivariate Data	93
5.2 Parameter Estimation	94
5.3 Estimation of Missing Values	95
5.4 Multivariate Normal Distribution	96
5.5 Multivariate Classification	100
5.6 Tuning Complexity	106
5.7 Discrete Features	108
5.8 Multivariate Regression	109
5.9 Notes	111
5.10 Exercises	112





Aurélien Géron

Geron, Chapter 5

R&N, Chapter 18

55

Chapter 5. Support Vector Machines

A *Support Vector Machine* (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. It is one of the most popular models in Machine Learning, and anyone interested in Machine Learning should have it in their toolbox. SVM are particularly well suited for classification of complex but small- or medium-sized datasets.

This chapter will explain the core concepts of SVMs, how to use them, and how they work.

- $h = 0$
- $h = \pm 1$

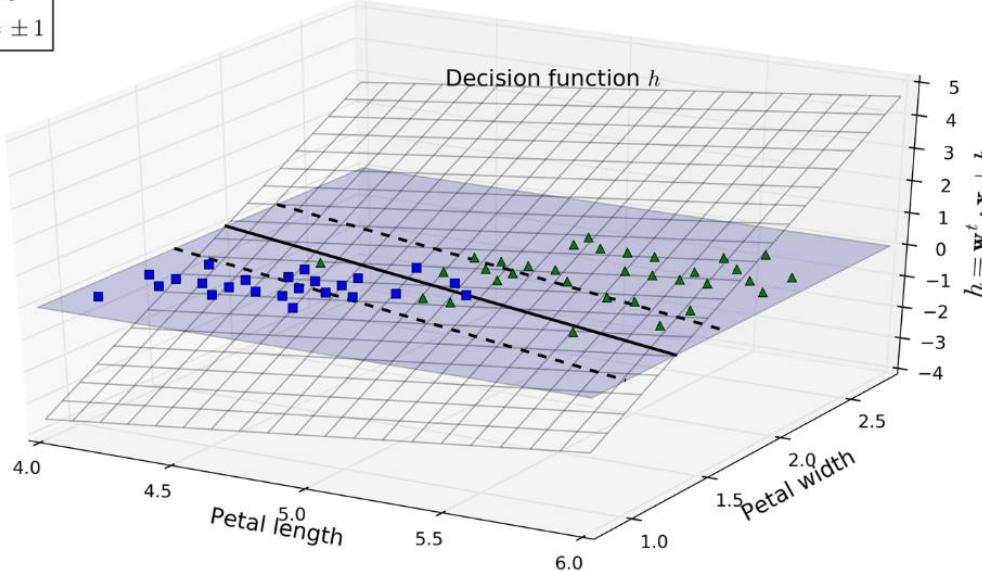


Figure 5-12. Decision function for the iris dataset

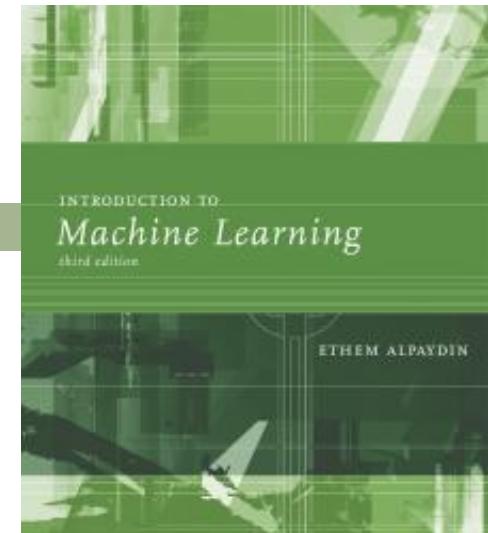
Alpaydin, Chapter 11

Neural Nets

56

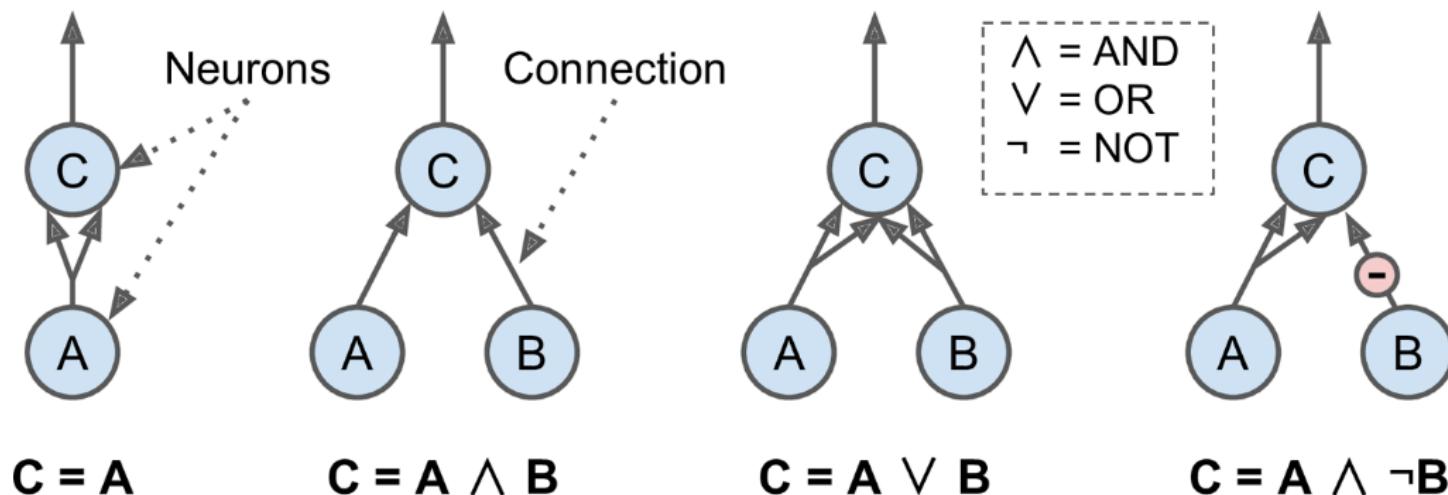
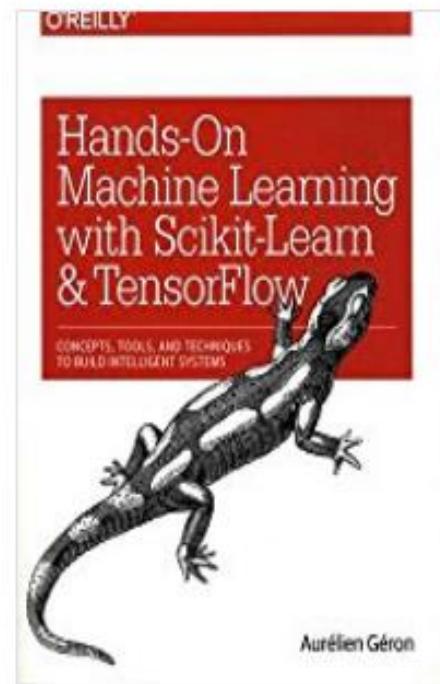
11 Multilayer Perceptrons 267

- 11.1 Introduction 267
 - 11.1.1 Understanding the Brain 268
 - 11.1.2 Neural Networks as a Paradigm for Parallel Processing 269
- 11.2 The Perceptron 271
- 11.3 Training a Perceptron 274
- 11.4 Learning Boolean Functions 277
- 11.5 Multilayer Perceptrons 279
- 11.6 MLP as a Universal Approximator 281
- 11.7 Backpropagation Algorithm 283
 - 11.7.1 Nonlinear Regression 284
 - 11.7.2 Two-Class Discrimination 286
 - 11.7.3 Multiclass Discrimination 288
 - 11.7.4 Multiple Hidden Layers 290



Chapter 10. Introduction to Artificial Neural Networks

Birds inspired us to fly, burdock plants inspired velcro, and nature has inspired many other inventions. It seems only logical, then, to look at the brain's architecture for inspiration on how to build an intelligent machine. This is the key idea that inspired *artificial neural networks* (ANNs). However, although planes were inspired by birds, they don't have to flap their wings. Similarly, ANNs have gradually become quite different from their biological cousins. Some researchers even argue that we should drop the biological analogy altogether (e.g., by saying "units" rather than "neurons"), lest we restrict our creativity to biologically plausible systems.¹



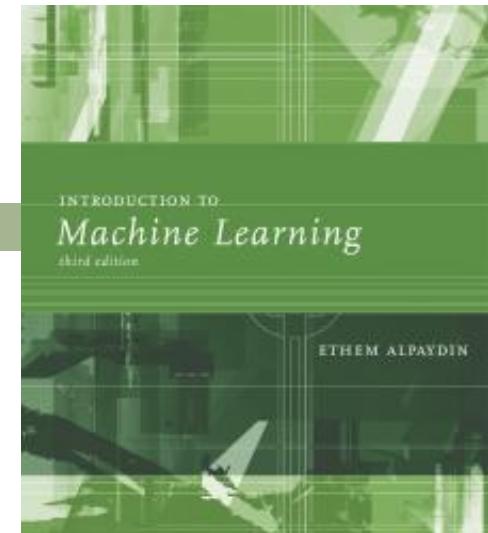
Alpaydin, Chapter 11

Neural Nets

58

11 Multilayer Perceptrons 267

- 11.1 Introduction 267
 - 11.1.1 Understanding the Brain 268
 - 11.1.2 Neural Networks as a Paradigm for Parallel Processing 269
- 11.2 The Perceptron 271
- 11.3 Training a Perceptron 274
- 11.4 Learning Boolean Functions 277
- 11.5 Multilayer Perceptrons 279
- 11.6 MLP as a Universal Approximator 281
- 11.7 Backpropagation Algorithm 283
 - 11.7.1 Nonlinear Regression 284
 - 11.7.2 Two-Class Discrimination 286
 - 11.7.3 Multiclass Discrimination 288
 - 11.7.4 Multiple Hidden Layers 290





Geron, Chapter 7

R&N, Chapter 18

59

Chapter 7. Ensemble Learning and Random Forests

Suppose you ask a complex question to thousands of random people, then aggregate their answers. In many cases you will find that this aggregated answer is better than an expert's answer. This is called the *wisdom of the crowd*. Similarly, if you aggregate the predictions of a group of predictors (such as classifiers or regressors), you will often get better predictions than with the best individual predictor. A group of predictors is called an *ensemble*; thus, this technique is called *Ensemble Learning*, and an Ensemble Learning algorithm is called an *Ensemble method*.

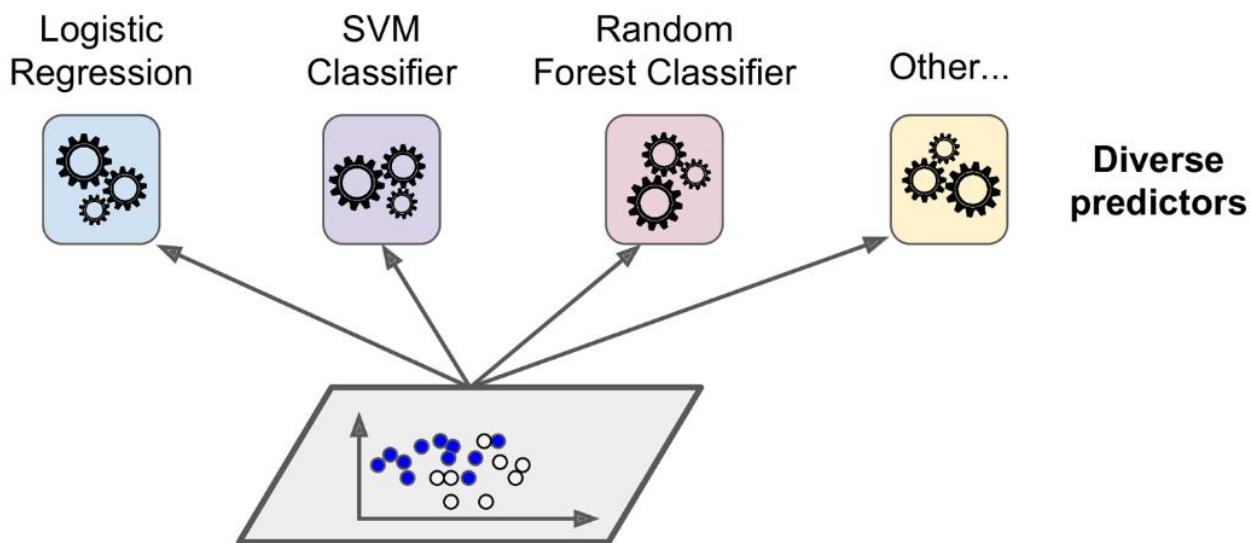


Figure 7-1. Training diverse classifiers

Final