

General Notes

CS, ML and Stats

Patrick Daly

Contents

1	Computer Science	4
1.1	Algorithms	4
1.2	Data Structures	7
1.3	Linux	7
2	Machine Learning	7
2.1	Supervised	7
2.1.1	Ordinary Least Squares (OLS)	7
2.1.2	Generalized Linear Model(GLM)	7
2.1.3	Logistic Regression	7
2.1.4	Linear Discriminant Analysis	7
2.1.5	Support Vector Machines	7
2.1.6	K-Nearest Neighbors	7
2.1.7	Gaussian Process	7
2.1.8	K-Nearest Neighbors	7
2.1.9	Decision Trees	7
2.1.10	Random Forest	7
2.1.11	Gaussian Process	7
2.1.12	Naive Bayes	7
2.1.13	Kalman Filter? dunno where this should go yet... maybe estimation section?	7
2.2	Unsupervised	7
2.2.1	Gaussian Mixture Models	7
2.2.2	K-Means	7
2.2.3	Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	7
2.2.4	Spectral Clustering	7
2.2.5	Hierarchical Clustering	7
2.2.6	Factor Analysis	7
2.2.7	Independent Component Analysis	7
2.2.8	Principal Component Analysis	7
2.2.9	Non-Negative Matrix Factorization (NMF)	7
2.2.10	Latent Dirichlet Allocation (LDA)	7
2.2.11	Outlier Detection?	7
3	Deep Learning	7
3.1	Convolutional Networks	7
3.2	Recurrent Networks	7

3.3	Long Short-Term Memory (LSTM)	7
3.4	Autoencoders	7
3.5	Reinforcement Learning	7
4	Linear Algebra	7
4.1	Norms	7
4.1.1	Euclidean / Frobenius	7
4.1.2	Manhattan	7
4.1.3	Infinity	7
4.1.4	Nuclear	7
4.1.5	Spectral	7
4.1.6	Symmetric	7
4.1.7	Positive Definite	7
4.1.8	Positive Semi-Definite	7
4.1.9	Negative Definite	7
4.1.10	Negative Semi-Definite	7
4.2	Eigendecomposition	7
4.3	Singular Value Decomposition (SVD)	7
4.4	Principal Component Analysis (PCA)	7
4.5	Independent Component Analysis (ICA)	7
4.6	Canonical Component Analysis (CCA)	7
4.7	Factor Analysis	7
5	Statistics	7
5.1	Probability Theory	7
5.2	Distributions	7
5.3	Combinatorics	7

1 Computer Science

1.1 Algorithms

DFS Time: $O(n)$, Space: $O(n)$
Solution exists far away.

Recursive

```
1 def dfs(node):
2     if node:
3         # do stuff if pre-order
4         if node.left:
5             dfs(node.left)
6         # do stuff if in-order
7         if node.right:
8             dfs(node.right)
9         # do stuff if post-order
```

Iterative

```
1 def dfs(node): # if bst, may need to swap search left/right
2     visited = set()
3     stack = [node]
4     while stack:
5         current = stack.pop(-1)
6         print(current.val)
7         if current not in visited:
8             visited.add(current)
9             if current.left and current.left not in visited:
10                stack.append(current.left)
11            if current.right and current.right not in visited:
12                stack.append(current.right)
13    return visited
```

BFS Time: $O(n)$, Space: $O(n)$
Iterative

```
1 def bfs(node):
2     stack = [node]
3     while stack:
```

```

4         current = stack.pop(-1)
5         if current.left:
6             stack.append(current.left)
7         if current.right:
8             stack.append(current.right)

```

Mergesort Time: $O(n \log n)$, Space: $O(n)$

```

1 def mergesort(array, start, end):
2     if start < end:
3         mid = (start+end) // 2
4         mergesort(array, start, mid)
5         mergesort(array, mid+1, end)
6         merge(array, start, mid, end)

1 def merge(array, start, mid, end):
2     left = array[start: mid+1]
3     right = array[mid+1: end+1]
4     i, j, k = 0, 0, start
5     while i < len(left) and j < len(right):
6         if left[i] < right[j]:
7             array[k] = left[i]
8             i += 1
9         else:
10            array[k] = right[j]
11            j += 1
12            k += 1
13    if j == len(right):
14        array[k: end+1] = left[i:]

```


1.2 Data Structures

1.3 Linux

2 Machine Learning

2.1 Supervised

2.1.1 Ordinary Least Squares (OLS)

2.1.2 Generalized Linear Model (GLM)

2.1.3 Logistic Regression

2.1.4 Linear Discriminant Analysis

2.1.5 Support Vector Machines

2.1.6 K-Nearest Neighbors

2.1.7 Gaussian Process

2.1.8 K-Nearest Neighbors

2.1.9 Decision Trees

2.1.10 Random Forest

2.1.11 Gaussian Process

2.1.12 Naive Bayes

2.1.13 Kalman Filter? dunno where this should go yet... maybe estimation section?

2.2 Unsupervised

2.2.1 Gaussian Mixture Models

2.2.2 K-Means

2.2.3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

2.2.4 Spectral Clustering

2.2.5 Hierarchical Clustering

2.2.6 Factor Analysis

2.2.7 Independent Component Analysis

2.2.8 Principal Component Analysis

2.2.9 Non-Negative Matrix Factorization (NMF)

2.2.10 Latent Dirichlet Allocation (LDA)

2.2.11 Outlier Detection?

3 Deep Learning