# 02450 - F24
# Machine Learning And Data Mining
# Full Course Notes

With Prof Bjørn Sand Jensen

These notes thoroughly cover 100% of content found in quizzes and ~98% of total content. Please do share!

Josiah Plett

# 02450 - Machine Learning - Notes (josiah plett - https://plett.dev)

I'm excited cause I just made a new site!

★ = Should do a practice problem for this

## 1 Data Feature Extraction & Visualization

- Predictive → Supervised Learning
- Descriptive → Unsupervised Learning
  - Clustering
  - Association Rule Discovery
  - Anomaly Detection

Classification: Discrete predictions
Regression: Continuous predictions

## 2 Data Features, PCA

### Dataset Manipulations

Sampling: selecting subset of data
Feature Extraction / Transformation: New features from existing attributes
Feature Subset Selection: A B C → A C
Dimensionality Reduction: Project data down dimensions, using PCA

### Types of Attributes ★

Discrete: countable set
Continuous: uncountable

Nominal: defined categories (Qualitative)
Ordinal: rankable (eg. <, >) (Qualitative)
Interval: Nominal + Ordinal
Ratio: zero means absent (Quantitative)

### Common Feature Transformations ★

Standardization: $X_{ij} = (X_{ij} - \hat{\mu}_j)/\hat{\sigma}_j$
Detrending
Binarization

One-out-of-K

| Value | A | B | C |
|---|---|---|---|
| A | 1 | 0 | 0 |
| B | 0 | 1 | 0 |
| C | 0 | 0 | 1 |

Bag of Words Representation
"a long sentence with stemming applied" →

| | |
|---|---|
| long | 1 |
| sentence | 1 |
| stem | 1 |
| unused | 0 |
| apply | 1 |

PCA - Principle Component Analysis ★ (BAD / GOOD)
Projection onto lower-dimensional plane that maximizes variance

## 3 Similarity, Probabilities

$s(x,y)$ → Similarity, 0-1
$d(x,y)$ → Dissimilarity, >0

Dissimilarity Measurements (M = total values)

| One-Norm | Euclidean |
|---|---|
| $d_1(x,y) = \sum_{j=1}^{M} \lvert x_j - y_j \rvert$ | $d_2(x,y) = \sqrt{\sum_{j=1}^{M} (x_j - y_j)^2}$ |

GMAD $d_p = \sum_{j=1}^{M} |x_j - y_j|^p$

Similarity Measurements (K = total values) ★

| SMC | Jaccard | Cosine | Extended Jaccard |
|---|---|---|---|
| $\frac{f_{00}+f_{11}}{K}$ | $\frac{f_{11}}{K - f_{00}}$ | $\frac{x^T y}{\lVert x \rVert_2 \lVert y \rVert_2}$ | $EJ(x,y) = \frac{x^T y}{\lVert x \rVert_2^2 + \lVert y \rVert_2^2 - x^T y}$ |

Transformations
Standardization: $\tilde{x}_{ik} = \frac{x_{ik} - \hat{\mu}_k}{\hat{\sigma}_k}$
weighting
Detrending
Combining heterogeneous attributes

## Probabilities ★

Sum: $P(A|B) + P(\bar{A}|B) = 1$
Product: $P(AB|C) = P(B|AC)P(A|C)$
Bayes: $P(A|BC) = \frac{P(B|AC)P(A|C)}{P(B|AC)P(A|C) + P(B|\bar{A}C)P(\bar{A}|C)}$
Given: $P(A|B) = \frac{P(A \cdot B)}{P(B)}$

Independence: $P(AB) = P(A)P(B)$
Expectation: $E[f] = \sum_{i=1}^{N} f(x_i) P(x_i)$
Variance: $Var[x] = \sum_{i=1}^{N} (x_i - E[x_i])^2 P(x_i)$
Mean: $E[X] = \sum_{i=1}^{N} x_i P(x_i)$
Bernoulli: $P(b|\theta) = \theta^b (1-\theta)^b$

## 4 Data Visualizations + Probability Densities

Probability ∀ 8 Density

Events are Intervals
Probability is the Integral
Density is the Slope

## Statistics ★

Mean: $\bar{x} = E[x]$
Expected: $E[g] = \int_{-\infty}^{\infty} g(x) P(x)\, dx$
Covariance: $cov(x,y) = E[(x-\bar{x})(y-\bar{y})]$
Variance: $var(x) = cov(x,x) = E[(x-\bar{x})]$
Standard Deviation: $std(x) = \sqrt{var(x)}$
$\mu$ is mean vector, $\Sigma$ is covariance matrix
Mahalanobis Distance: $mahalanobis(x,y)^2 = (x-y)^T \Sigma^{-1} (x-y)$

## Visualizations ★

Line plot
Scatterplot
Bar plot

### Single Attribute

Histogram
Box Plot (Attr1, Attr2; 25%, 75%; Min, Median, Max)

### Relation Between Attributes

2D Histogram
Scatterplots (Eg All Pairs of Attrs.)

### Higher Dimensions

MatrixPlots
Parallel Coor

## 5 Decision Trees, Regression

### Unsupervised Learning

Types:
- Regression: Continuous output
- Classification: Discrete output

Data: inputs+outputs
Model: $f(\text{input}) = \text{output}$ (Same as Unsupervised)
Cost Function: $d(\text{output}, f(\text{input}))$

### Classification

"Classification Trees", "Regression Trees"

#### Decision Trees

Ask a series of questions until a conclusion is reached. Hunt's Algorithm starts at the root and repeatedly asks "good questions," making leaf nodes if all surviving data points match.

Purity Gain ★

$\Delta = I(r) - \sum_{k=1}^{K=2} \frac{N(V_k)}{N(r)} I(V_k)$

$p(c|V) = \frac{\{\#\ \in \text{class } C \in \text{branch } V\}}{N(V)}$

$\text{Entropy}(v) = -\sum_{c=1}^{C} p(c|v) \log_2 p(c|v)$

$\text{Gini}(v) = 1 - \sum_{c=1}^{C} p(c|v)^2$

$\text{ClassError}(v) = 1 - \max_c p(c|v)$.

Note there are diff types of splits:
Binary, Nominal, Ordinal, Continuous

Information Gain
$I = -\log p_i$

### Evaluating a Classifier

Confusion Matrix

| Actual \ Predicted | | |
|---|---|---|
| | 3 | 2 |
| | 0 | 5 |

Accuracy $\frac{3+5}{3+2+0+5} = 0.8$
Error $\frac{2}{10} = 0.2$

### Evaluating a Regression Model

Average Loss per Observation
$E = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))$
$L_1 = |y_i - \hat{y}_i|$ $L_2 = (y_i - \hat{y}_i)^2$

### Logistic Regression "Assume $y_i$ is binary"

$\theta_i = \sigma(f(x,w))$ where $\sigma(z) = \frac{1}{1+e^{-z}}$ ★

$E(w) = \frac{-1}{N} \sum_{i=1}^{N} \left[ y_i \log(\theta_i) + (1-y_i) \log(1-\theta_i) \right]$, $\theta_i = \sigma(x_i^T w)$

### General Linear Model (GLM)

$d$ = cost function, $g$ = link function
$E(w) = \frac{1}{N} \sum_{i=1}^{N} d(y_i, g(\tilde{x}_i^T w))$

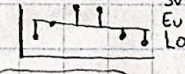
## Supervised Learning

### Linear Regression ★

K-dimensional inputs

$f(x) = w_0 + w_1 x_1 + \ldots + w_K x_K = \sum_{i=0}^{K} w_i x_i = X^T W$ $(x_0 = 1)$

Residual Error



sum of Euclidean Loss ($L_2$)

### Learning

Maxima A Posteriori (MAP) Learning
Data: $X = x_1 \cdots x_N$ $Y = y_1 \cdots y_N$
Parameters: $w_i$ relates $x_i \to y_i$
Maximizing $w^* = \max_w p(w|X,y)$ *is equivalent to*
$w^* = \min_w E(w)$ where

$E(w) = \frac{1}{N} \left[ -\sum_{i=1}^{N} \log p(y_i|x_i,w) - \log p(w) \right]$ ★

### MATRIX MULTIPLICATION

$\underset{3\times3}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \underset{3\times1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

$x_{3,2} = 1 \cdot x + 2 \cdot b + 3 \cdot c$

### NORMS

Euclidian $\lVert x \rVert_2 = \sqrt{x^T x}$
Frobenius $\lVert X \rVert_F^2 = \sum_{i,j}^{n,m} X_{ij}^2$

# 02450 - Machine Learning - Notes 2

(Josiah Plett https://plett.dev) ★ = Should do a practice problem for this

## 6 Overfitting, Cross-Validation, Nearest-Neighbour

### Overfitting

To solve, try to control "model complexity" eg. K in linear regression, or "pruning" in d. trees.

### Errors

**Training Error**

$$E_{M_k}^{train} = \frac{1}{N^{train}} \sum_{i \in D^{train}} (y_i - f_{M_k}(x_i, w))^2$$

**Test Error**

Same as train but for test data.

**Generalization Error**

G. error is test error over an infinitely large test set.

If we had many test sets $D_1, \dots, D_K$, then:

$$E_M^{Gen} \approx \frac{1}{K} \sum_{k=1}^{K} E_{M, D_k}^{test}$$

### Cross-Validation ★

Purpose: Estimate generalization Error.

Use:
- Holdout
- K-fold
- Leave-1-out

### 1-layer Cross Validation

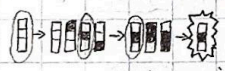
Purpose: Select best of S models,

Run basic cross-validation, and simply select the best one!

### Forward Selection ★

or, "Sequential Feature Selection"

1. Start w/ no features.
2. Compute cross-validation error for all +1-feature subsets.
3. Choose the best one.
4. Repeat until no change.



Backward Selection is in reverse.

### 2-Layer Cross-Validation ★

Purpose: Select optimal model AND estimate its generalization error.

Outer Loop is for comparing models via basic cross-validation

Inner Loop is for estimating model parameters via 1-layer CV

### Nearest Neighbours

Can use euclidian distance or a specified distance function.

**KNN with Leave-1-Out CV** ★

For each observation $x_i$:
1. Temporarily remove $x_i$.
2. Find K nearest neighbours around $x_i$.
3. Determine if $x_i$ is classified correctly based on its neighbourhood, $c_i = 0, 1$

### Bayesian Classification

EQUATIONS ★

$$p(y|x_1 \dots x_M) = \frac{p(x_1 \dots x_M | y) p(y)}{\sum_{k=0}^{K-1} p(x_1 \dots x_M | y=k) p(y=k)}$$

$$p(x_i \dots x_M | y=k) = \frac{\#\text{ of times } y=k \text{ AND we get } x_1 \dots x_M}{\#\text{ of times } y=k}$$

Naïve Bayes' Assumption: $p(x_1, x_2, \dots x_M | y) = p(x_1|y) p(x_2|y) \cdots p(x_M|y)$

## 7 Performance Evaluation, (Naïve) Bayes

### Performance Evaluation

Outline
- What's our overall Objective?
- What tools are available?
- What specific test should we use?

### Hypothesis Testing

Let $z$ (perhaps $z = E_A^{Gen} - E_B^{Gen}$) be a quantity of interest.

Test: $H_0: z = 0$ vs $H_1: z \neq 0$

1. Estimate test error somehow.
2. Calculate p-value for $H_1$.

Estimated result: $\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} z_i$

### Statistical Tools

*Too technical for me, will not be writing down.*

### How to Evaluate ★

Training on D, we obtain prediction rules $f_{D,A}: x \to y$ and $f_{D,B}: x \to y$

then we compare generalization error:

$$E_{D,A}^{gen} = \int p(x,y) L(f_{D,A}(x), y) \, dx \, dy$$

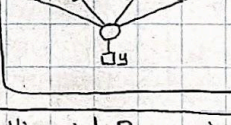
probability; Loss function; Setup I

But this is only for dataset D. To generalize, use many $D_i$ and compute the (weighted) average. This we'll call Setup II

Setup II is ideal but requires lots of CV so go with Setup I if you only have a single train-test split.

### General Linear Model

- Data: $\{x_n, y_n\}_{n=1}^{N}$
- Model: $f(x) = g_{link}(x^T w)$
- Cost function: $d(y, f(x))$
- Parameters: $w = \arg\min_w \sum_{n=1}^{N} d(y_n, f(x_n))$




### Artificial Neural Networks ★

All same, but now model is:

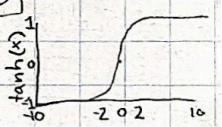
- Model: $f(x) = h^{(2)}\left(V_{10} + \sum_{j=2}^{H} V_{1j} h^{(1)}(\tilde{x}^T w_j)\right)$

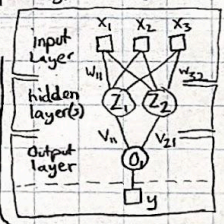
Common Choices

$h^{(1)}(x) = \tanh(x)$

$h^{(2)}(x) = x$

$d(y, f(x)) = (y - f(x))^2$



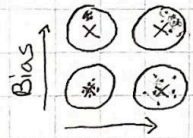

$Z^{(1)}$: "hidden layer"

$Z_j^{(1)}$: "node"

$W^{(1)}$: weights

## 8 ANNs (Artificial Neural Networks) and Bias/Variance

### Bias-Variance Decomposition ★



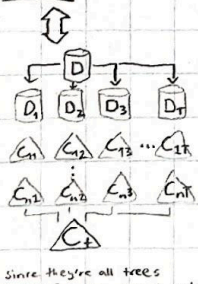

$$E_D[E^{gen}] = E_x\left[ \underbrace{Var_{y|x}[y]}_{\text{Intrinsic difficulty of the problem}} + \underbrace{(\bar{y}(x) - \bar{f}(x))^2}_{\text{Bias term; skew from true mean}} + \underbrace{Var_D[f_D(x)]}_{\text{Variance term; "wigglyness"}} \right]$$

## 9 AUC and Ensemble Methods

"Ensemble" → many weak classifiers combined into one strong classifier

"Bagging" → new datasets made by drawing randomly from pool with replacement

"Boosting" → train classifier on a bag, update bag weights and repeat,




Since they're all trees (classifiers → nodes + branches) this is called Random Forest! ^_^

### Adaboost ★

$w_i(1) = \frac{1}{N}$ for $i = 1 \dots N$

for $t = 1 \dots T$ do:

$D_t$ by Bagging

$f_t$ is classifier on $D_t$

$\epsilon_t = \sum_{i=1}^{N} w_i(t)(1 - d(f_t(x_i), y_i))$ ← weighted error

$\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}$

$w_i(t+1) = \frac{\tilde{w}_i(t+1)}{\sum_{j=1}^{N} \tilde{w}_j(t+1)}$, $\tilde{w}(t+1) = \begin{cases} w_i(t) e^{-\alpha_t} \\ w_i(t) e^{\alpha_t} \end{cases}$

### Multinomial Regression

**Cost Function** ★

($z_i$ is one-of-K encoding of $y_i$)

$$E = -\sum_{i=1}^{N} \log p(y_i | x_i)$$

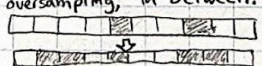
### Multi-class Neural Networks

suppose $\tilde{y}_i \dots \tilde{y}_K$ are outputs of a NN

$$E = -\sum_{i=1}^{N} \log p(y_i | \tilde{y}_i)$$

### Solving the Class Imbalance Problem

- Resampling: Undersampling oversampling, in between.



- Modify classification algo.
- consider class imbalance when measuring Performance

### Precision & Recall ★

First, the Confusion Matrix

| Actual \ Predicted | Positive | Negative |
|---|---|---|
| Positive | TP | FN |
| Negative | FP | TN |

$$\text{Precision } p = \frac{TP}{TP + FP}$$

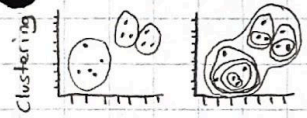
$$\text{Recall } r = \frac{TP}{TP + FN}$$

# 02450 - Machine Learning - Notes 3 (josiah plett - https://plett.dev)

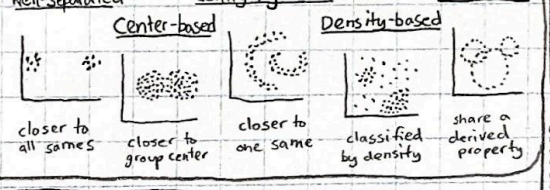
★ = should do a practice problem for this

## (10) Clustering: K-means, Heirarchical

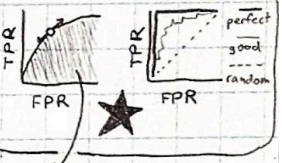
Partial vs Heirarchical clustering



### Types of Clustering ★




### Receiver Operating Characteristic




★

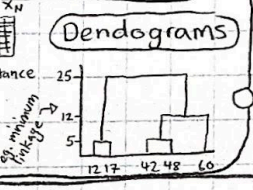
Area Under the Curve — probability of ranking a random + higher than −

### K-Means Clustering ★

1. Select K points as initial centroids.
2. Repeat:
   - Form K clusters by assigning points to closest centroids.
   - Recompute centroids.
3. Until: centroids don't change.

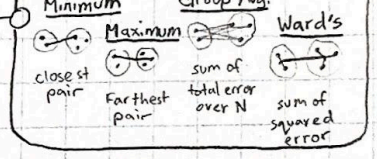
### Agglomerative Heirarchical Clustering ★

1. Initialize proximity matrix.
2. Repeat:
   - Merge the closest clusters.
   - Update proximity matrix.
3. Until: there's just 1 cluster.




### Dendograms

### Linkage Functions ★




## (11) Mixture Models (MM) and Density Estimation

### Gaussian Mixture Model (GMM)

★ A new, probability-based method of clustering.

Locations: $\mu_k$
Shapes: $\Sigma_k$
Sizes: $\omega_k$

**E-step**

$$p(z_n=k|x_n) = \frac{p(z=k)p(x_n|z_n=k)}{\sum_K p(z=k)p(x_n|z_n=k)}$$

**M-step**

$$N_k = \sum_{n=1}^{N} p(z_n=k|x_n) \qquad w_k = \frac{N_k}{N}$$

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} x_n p(z_n=k|x_n)$$

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N}(x_n-\mu_k)(x_n-\mu_k)^T$$

**EM Algorithm**

1. Select some initial models.
2. Repeat:
   - Expectation: calculate $p(z_n=k|x_n)$ for all $x_n$.
   - Maximization: for each distribution, estimate parameters by max likelihood.
3. Until: params don't change.

"Mixture Models" are assessed via cross-validation, converging on a good K.

### Probability

$\mu$ = Mean ("mu" → "M")
$\sigma$ = Standard Deviation ("sigma" → "S")
$\sigma^2$ = Variance

**normal distribution**

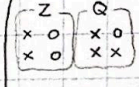
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

**multivariate normal distribution**

$$\Sigma^{(2)} = \begin{bmatrix} var(x_1) & cov(x_1,x_2) \\ cov(x_2,x_1) & var(x_2) \end{bmatrix}$$

$$N(x|\mu,\Sigma^d) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

### Comparing Partitions




How similar are Z & Q?

Z = [1 1 2 2]
Q = [1 1 1 2]

$$\mathbb{I}_{ij} = \begin{cases} 0 & o_i \neq o_j \\ 1 & o_i = o_j \end{cases}$$

$b^Z$ = [1 0 0 0 0 1]
$b^Q$ = [1 1 0 1 0 0]

S = {# of $b^Z_i = b^Q_i$ AKA total matches}
D = {# of $b^Z_i \neq b^Q_i$ AKA total mismatches}

| Rand Index | Jaccard Sim. |
|---|---|
| $\frac{S+D}{\frac{1}{2}N(N-1)}$ | $\frac{S}{\frac{1}{2}N(N-1)-D}$ |

$R(Z,Q) = \frac{3+3}{4\times3/2} = 1$ $\quad J(Z,Q) = \frac{3}{4+3/2-3} = 1$

$n^Z = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $n^Q = [3\ 1]$ $n = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$

*Information & Entropy Recap*

Information from $p_i$: $I = -\log p_i$

Avg Information is Entropy: $E[I] = -\sum_{i=1}^{N} p_i \log p_i = H[p_x]$

Mutual Info: $MI[x,y] = H[p_x] + H[p_y] - H[p_{x,y}]$

↳ Normalized: $NMI[x,y] = \frac{MI[x,y]}{\sqrt{H[p_x]}\sqrt{H[p_y]}}$

### Anomaly Detection

"Anomalies have very low density in multiple methods."

**Density-based Techniques**

- Kernel Density Estimator

$$p(x) = \sum_{n=1}^{N} \frac{1}{N} \mathcal{N}(x|x_n,\sigma^2 I)$$ ★

How to get $\sigma^2$? Cross-validation.

- Inverse distance density estimator

$$density_{X|_i}(x_i;K) = \left(\frac{1}{K}\sum_{x\in N_{X|_i}(x_i;K)} d(x_i,x')\right)^{-1}$$

Inverse average distance to K nearest neighbours

- Average Relative Density ★

$$ard_X(x_i;K) = \frac{density_{X|_i}(x_i,K)}{\frac{1}{K}\,sum\{density_{X|_j}(x_j;K)\}}$$


Thank you for using my notes!
If you want to say thanks, just share them with more people ♡ (or maybe even your prof for better reach)

## (12) Association Mining

### Definitions

**Itemset**: eg. {B,C,D}, {A,D}, {}
**Support for Itemset X**: % that have X
**Association Rule**: $X \to Y$ where $X \cup Y = \emptyset$
**Support**: (for $X \to Y$)

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N} = P(X,Y)$$ ★

**Confidence**:

$$c(X \to Y) = \frac{s(X \cup Y)}{s(X)} = P(Y|X)$$

### A-priori Algorithm ★

An algorithm for much more efficiently finding highly supported itemsets.

1. Find all 1-itemsets ($L_1$). eg $L_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ← itemsets
2. Generate k-itemsets by merging rows in $L_{k-1}$. eg $L_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$
3. Remove any rows in $L_k$ that contain any subset of length k-1 that is infrequent, AKA not in $L_{k-1}$. (k++)
4. Compute support of rows in $L_k$, remove any with insufficient support.
5. $L_1 \dots L_K$ are now all "frequent" itemsets.

*We are permitted 2 sheets of A4 paper on this exam. Use this last blank page for any personal notes as you see fit.*

*Obligatory cute red panda photo ☺️*