Machine Learning

Introduction
Part of “Data mining”

Data mining is the process of exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules.  
[Mastering Data Mining by Berry and Linoff, 2000]
Why Data mining is important NOW?

• Massive Data Sources
• Advanced computer technology — fast access to massive data source, capability in using computationally intensive algorithm and statistical methods
• Knowledge in subject matter — ask important questions, understand/verify discovered knowledge
Problem identification

Data preparation

Exploring data to gain understanding
(suspected associations, unanticipated characteristics, obvious anomalies)

Data cleaning and transformation

Data division

Train data

Validation data

Test data

Build and fit model

Tune and refine model

Evaluate model

Tune and refine model

Choose model

One may consider several models

Prediction

Make decision

Accuracy requirement met?

no

Consider alternative model

yes

Collect more data

Choose model

Build and fit model

Tune and refine model

Evaluate model
Learning : Supervised and Unsupervised learning

• Supervised Learning
  – Learning with a teacher (feedback)
  – Classification

• Unsupervised Learning
  – Learning without a teacher (optimizing)
  – Clustering
Unsupervised Learning

• Clustering methods – hierarchical, k-means
• Dimension reduction: Factor Analysis, Principal Component Analysis, Singular Value Decomposition
• Linear Algebra based methods
• Self-organizing maps – artificial neural networks
• Distance, metrics.
Supervised learning

- Classification: output is a categorical variable
- Prediction (Regression): output is a (numerical, continuous) quantitative variable
- Steps
  - Producing an answer $\hat{y}_i$ to a given input $x_i$
  - the correct answer $y_i$ is given as a feedback --- an error for student’s answer: $|\hat{y}_i - y_i|$ or $(\hat{y}_i - y_i)^2$ etc. (depends on the choice of the loss-risk-error functions)
  - result characterized by a loss function: $L(y; \hat{y})$
  - The predictive model is based on the function approximation: $y = f(x; \varepsilon)$
Supervised learning – Some Basic Methods

• Boosting - family of methods
  – iteratively learning with weak classifiers and adding them to a final strong classifier
  – Each time another weak learner is added, the data is reweighted: examples that are misclassified gain weight.
  – Adaboost : minimizing an exponential loss function, historically important.

• Decision trees - Principle of Maximum Entropy (always choose the probability distribution with the largest entropy) Entropy: \( E[-\ln(P(X))] \)

• Backpropagation on Neural Networks – updating weights, delta method (gradient descent learning rule in a perceptron)

• Gradient Descent – always step towards the negative gradient, minimize the loss function in that way

• Support Vector Machine – hyper-planes separating data subsets, classification/regression method

• Cross-validation: leave-one-out, repeat that are classified correctly lose weight

• Bootstrap sampling – resampling ting sub-samples, k-subsamples
Supervised Learning - Regression

- Linear methods based on LSE estimation are the most typical.
- Generally: everything is regression when we estimate the relationship between a dependent and an independent variable using functions and we estimate the parameters of these functions.
- Classification (logistic regression) is based on the very same model.
- MANY types of regression.
Supervised Learning – Linear Regression

• $Z_y = R \cdot Z_x$, correlation
• Regression effect
Supervised Learning – Logistic Regression

\[ P = \frac{e^{a+bX}}{1 + e^{a+bX}} \]
Supervised Learning – Logistic Regression

$$odds = \frac{P}{1-P}$$
Supervised Learning - Bayes Classifiers

- The class probability of $\pi_i$ is the Prior: $P(\pi_i)$
- Conditional probability is the Likelihood Function: $P(x|\pi_i)$
- Unconditional probability is the normalizing constant in Bayes’ Theorem: $P(x) = \sum_{j=1}^{K} P(x|\pi_i)P(\pi_i)$
- Posterior probability = Likelihood x Prior / Normalizing constant
  $$P(\pi_i|x) = \frac{P(x|\pi_i)P(\pi_i)}{P(x)}$$
- Bayes classifier maximizes posterior probability.

- Theorem:
  - The Bayes estimator $T$ of parameter $\theta$ under the square loss function $L(T, \theta) = [T - \theta]^2$ is the mean of the posterior distribution. (Expected value)
  - The Bayes estimator $T$ of parameter $\theta$ under the square loss function $L(T, \theta) = |T - \theta|$ is the median of the posterior distribution. (50th percentile.)
Supervised Learning - Bayesian Posterior Predictive Distributions

- Posterior distribution – loss function tells which estimator is of interest
- Monte Carlo sampling
- Metropolis Hastings Algorithms (MCMC sampling in high dimensions when it is difficult to sample from the original distribution)
- Posterior predictive distribution – distribution of $y$

$$p(y_{\text{observable}} | y) = \int p(y_{\text{observable}} | \theta) p(\theta | y) d\theta$$
Kalman Filter – Linear Quadratic Estimation

- Bayesian
- Linear method
- Supervised learning
- Recursive
- Let’s see in details...
Rudolf (Rudy) Emil Kálmán, born in Budapest May 19, 1930, electro-engineer 1960: "A new approach to linear filtering and prediction problems"

The Kalman filter is:

- a set of mathematical equations
- that provides an efficient computational (recursive) means to estimate the state of a process,
- in a way that minimizes the mean of the squared error.
What is Kalman Filtering?

- Generates **optimal** estimate of desired quantities.
- **Recursive** data processing algorithm.
- Optimal?
  - For linear system and white Gaussian errors, Kalman filter is "best" estimate based on all previous measurements.
  - For non-linear systems given only the mean and standard deviation of noise, the Kalman filter is the best *linear estimator*. *Non-linear estimators may be better.*
- Recursive?
  - Doesn’t need to store all previous measurements and reprocess all data each time step
  - Real time processing!
Mean of Squared Error - MSE

T is an estimator of parameter \( \theta \).
\( \alpha \) is the loss function.
Several loss functions are possible.
In case an estimator \( T \) has finite variance, we might use the squared error as loss function:
\[
\alpha(T, \theta) = (T - \theta)^2
\]
The risk is the mean (expected value) of the loss function:
\[
E_\theta [\alpha(T, \theta)] = E_\theta [(T - \theta)^2] \equiv MSE_\theta(T)
\]
Mean of Squared Error - MSE

\[ E_\theta[\alpha(T, \theta)] = E_\theta[(T - \theta)^2] \equiv MSE_\theta(T) \]

\[ MSE_\theta(T) = E_\theta[(T - \theta)^2] \]

\[ = E_\theta[(T - E_\theta(T) + E_\theta(T) - \theta)^2] \]

\[ = E_\theta[(T - E_\theta(T)]^2 + (E_\theta(T) - \theta)^2 \]

\[ = Variance_{\theta}(T) + [Bias_{\theta}(T)]^2 \]

The MSE of an estimator is the sum of its variance and the square of its bias.
Expectation, Mean, Expected value in high dimension

• Let $x$ be a random variable.

• The expected value in continuous and discrete case:

$$E[x] = \int x \cdot p(x) \, dx$$

$$\bar{x} = \frac{1}{N} \sum_{1}^{N} x_i$$

• Expected value of a vector $\mathbf{x}$ is calculated by components:

$$E[\mathbf{x}] = \overline{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n]^T$$
Variance and Covariance in higher dimensions

• The variance is by definition:
  \[ E_\theta [(x - E_\theta (x))^2] \]

• Covariance matrix is calculated by:
  \[ E_\theta [(x - E_\theta (x)) (x - E_\theta (x))'] \]

  – The diagonal elements are the Variances, the off-diagonals are the Covariances.
  – Generally speaking: element in the i, j position is the covariance between the i th and j th elements of a random vector
Assumptions behind Kalman Filter

• The model you use to predict the ‘state’ needs to be a LINEAR function of the measurement.
• The model error and the measurement error (noise) must be Gaussian with zero mean.
Overview of the process

1. Initial conditions are the previous corrected prediction and its variance ($\hat{x}_{k-1}$ and $p_{k-1}$)
2. Make prediction ($\hat{x}_k^-, p_k^-$) before correction – using the initial conditions/model/parameters
3. Take a new measurement ($z_k$)
4. Perform Correction to obtain corrected prediction with corrected variance ($\hat{x}_k$ and $p_k$)
   1. Correct prediction by using the residuals
      \[
      \text{Optimal estimate } ((x) = \text{Prediction } + \text{(Kalman Gain) } \times \text{(Measurement } - \text{Prediction)}
      \]
   2. Obtain a smaller variance
      \[
      \text{Variance of estimate } = \text{Variance of prediction } \times (1 - \text{Kalman Gain})
      \]
Estimates and Errors

- $\hat{x}_k \in \mathbb{R}^n$ is the estimated state at time-step $k$.
- $\hat{x}_k^- \in \mathbb{R}^n$ after prediction, before observation.
- Errors:
  - $e_k^- = x_k - \hat{x}_k^-$
  - $e_k = x_k - \hat{x}_k$
- Error covariance matrices:
  - $P_k^- = E[e_k^- e_k^-^T]$
  - $P_k = E[e_k e_k^T]$
- Kalman Filter’s task is to update $\hat{x}_k$ and $P_k$
The set of Kalman Filtering Equations in Detail

**Prediction (Time Update)**
1. Project the state ahead
   \[ \hat{X}_k = AX_{k-1} + Bu_k \]
2. Project the error covariance ahead
   \[ \hat{P}^-_k = AP_{k-1}A^T + Q \]

**Correction (Measurement Update)**
1. Compute the Kalman Gain
   \[ K = \hat{P}^-_k H^T (H \hat{P}^-_k H^T + R)^{-1} \]
2. Update estimate with measurement \( z_k \)
   \[ \hat{X}_k = \hat{X}^-_k + K(z_k - H\hat{X}^-_k) \]
3. Update Error Covariance
   \[ P_k = (I - KH) \hat{P}^-_k \]

**Equations**
- nxn matrix describes how the state evolves from step t-1 to t.
- nxl matrix describes how the control \( u_i \) changes the state from t-1 to t.
- kxn matrix describes how to map the state \( x_i \) to an observation \( z_i \).

Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance \( R_i \) and \( Q_i \) respectively.
Computational approach: the components of Kalman filter

A: nxn matrix describes how the state evolves from step $t-1$ to $t$.

B: nxl matrix describes how the control $u_t$ changes the state from $t-1$ to $t$.

C: kxn matrix describes how to map the state $x_t$ to an observation $z_t$.

R,Q: Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance $R_t$ and $Q_t$ respectively.
Kalman filter algorithm

- **Input:** $A, B, C, Q, R, \hat{x}^{-}_{k}, \ P^{-}_{k}, \ u_{k}$
- **Prediction:**
  \[
  \hat{x}^{'}_{k} = Ax^{-}_{k-1} + Bu_{k} \\
  \hat{P}^{-}_{k} = AP^{-}_{k-1}A^{T} + Q
  \]
- **Correction:**
  \[
  K = \hat{P}^{-}_{k} H^{T} (H \hat{P}^{-}_{k} H^{T} + R)^{-1} \\
  \hat{x}^{-}_{k} = \hat{x}^{'}_{k} + K(z_{k} - H \hat{x}^{-}_{k}) \\
  P_{k} = (I - KH) \hat{P}^{-}_{k}
  \]

$u \in \mathbb{R}^{l} \text{motor}$

$x \in \mathbb{R}^{n} \text{state}$

$z \in \mathbb{R}^{m} \text{sense}$
Discrete Kalman Filter

• Estimate the state \( x \in \mathbb{R}^n \) of a linear stochastic difference equation

\[
x_k = Ax_{k-1} + Bu_k + w_{k-1}
\]

– process noise \( w \) is drawn from \( N(0,Q) \), with covariance matrix \( Q \).

• with a measurement \( z \in \mathbb{R}^m \)

\[
z_k = Hx_k + v_k
\]

– measurement noise \( v \) is drawn from \( N(0,R) \), with covariance matrix \( R \).

• \( A, Q \) are \( nxn \). \( B \) is \( nxl \). \( R \) is \( mxm \). \( H \) is \( mxn \).
Time Update (Predictor)

• Update expected value of $x$
  \[ \hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \]

• Update error covariance matrix $P$
  \[ P_k^- = AP_{k-1}A^T + Q \]

• Previous statements were simplified versions of the same idea:
  \[ \hat{x}(t_3^-) = \hat{x}(t_2) + u[t_3 - t_2] \]
  \[ \sigma^2(t_3^-) = \sigma^2(t_2) + \sigma^2_\epsilon [t_3 - t_2] \]
Measurement Update (Corrector)

• Update expected value
  \[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \]
  "innovation" is \[ z_k - H \hat{x}_k^- \]

• Update error covariance matrix
  \[ P_k = (I - K_k H) P_k^- \]

• Compare with previous form
  \[ \hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)(z_3 - \hat{x}(t_3^-)) \]
  \[ \sigma^2(t_3) = (1 - K(t_3)) \sigma^2(t_3^-) \]
The Kalman Gain

- The optimal Kalman gain $K_k$ is
  \[ K_k = P_k^{-1}H^T (HP_k^{-1}H^T + R)^{-1} \]
  \[ = \frac{P_k^{-1}H^T}{HP_k^{-1}H^T + R} \]

- Compare with previous form
  \[ K(t_3) = \frac{\sigma^2(t_3^-)}{\sigma^2(t_3^-) + \sigma_3^2} \]
Extended Kalman Filter

• Suppose the state-evolution and measurement equations are non-linear:

\[
x_k = f(x_{k-1}, u_k) + w_{k-1}
\]

\[
z_k = h(x_k) + v_k
\]

– process noise \( w \) is drawn from \( N(0, Q) \), with covariance matrix \( Q \).

– measurement noise \( v \) is drawn from \( N(0, R) \), with covariance matrix \( R \).
The Jacobian Matrix

• For a scalar function $y = f(x)$,
  \[ \Delta y = f'(x)\Delta x \]

• For a vector function $y = f(x)$,
  \[ \Delta y = J \Delta x = \begin{bmatrix} \Delta y_1 \\ \vdots \\ \Delta y_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} \]
Linearize the Non-Linear

• Let $A$ be the Jacobian of $f$ with respect to $x$.

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_{k-1}, u_k)$$

• Let $H$ be the Jacobian of $h$ with respect to $x$.

$$H_{ij} = \frac{\partial h_i}{\partial x_j}(x_k)$$

• Then the Kalman Filter equations are almost the same as before!
EKF Update Equations

• Predictor step:  
  \[ \hat{x}_k^- = f(\hat{x}_{k-1}, u_k) \]
  \[ P_k^- = AP_{k-1}A^T + Q \]

• Kalman gain:  
  \[ K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \]

• Corrector step:  
  \[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)) \]
  \[ P_k = (I - K_k H) P_k^- \]
Blending Factor

• If we are sure about measurements:
  – Measurement error covariance (R) decreases to zero
  – K decreases and weights residual more heavily than prediction

• If we are sure about prediction
  – Prediction error covariance $P^{-k}$ decreases to zero
  – K increases and weights prediction more heavily than residual
In-class exercise: answer in 5-5 sentences:

1. Why Kalman filters are often used in GPS technologies?

2. What are the weaknesses of the Kalman filtering?

3. What is Bayes’ theorem? Give the mathematical definition and discuss its components and importance in 5 sentences.