CS 307. Logistic Regression, Kalman Filtering
Today’s Goal

• Recall of Naïve Bayes Classifier
  • Variable independence assumption
  • Test is straightforward
  • Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption

• Logistic regression

• Kalman filtering
Multivariate analysis

• Machine learning models
  • Linear regression
  • Logistic regression
  • Poisson regression
  • Loglinear model
  • Discriminant analysis
  • ......

• Choice of the tool according to the objectives, the study, and the variables
Simple linear regression

<table>
<thead>
<tr>
<th>Age</th>
<th>SBP</th>
<th>Age</th>
<th>SBP</th>
<th>Age</th>
<th>SBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>131</td>
<td>41</td>
<td>139</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>128</td>
<td>41</td>
<td>171</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>116</td>
<td>46</td>
<td>137</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>106</td>
<td>47</td>
<td>111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>114</td>
<td>48</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>123</td>
<td>49</td>
<td>133</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>117</td>
<td>49</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>122</td>
<td>50</td>
<td>183</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>99</td>
<td>51</td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>121</td>
<td>51</td>
<td>133</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>147</td>
<td>51</td>
<td>144</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1  Age and systolic blood pressure (SBP) among 33 adult women

<table>
<thead>
<tr>
<th>Age</th>
<th>SBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>128</td>
</tr>
<tr>
<td>54</td>
<td>105</td>
</tr>
<tr>
<td>56</td>
<td>145</td>
</tr>
<tr>
<td>57</td>
<td>141</td>
</tr>
<tr>
<td>58</td>
<td>153</td>
</tr>
<tr>
<td>59</td>
<td>157</td>
</tr>
<tr>
<td>63</td>
<td>155</td>
</tr>
<tr>
<td>67</td>
<td>176</td>
</tr>
<tr>
<td>71</td>
<td>172</td>
</tr>
<tr>
<td>77</td>
<td>178</td>
</tr>
<tr>
<td>81</td>
<td>217</td>
</tr>
</tbody>
</table>
SBP (mm Hg)

SBP = 81.54 + 1.222 \cdot \text{Age}

Simple linear regression

- Relation between 2 continuous variables (SBP and age)

- Regression coefficient $\beta_1$
  - Measures association between $y$ and $x$
  - Amount by which $y$ changes on average when $x$ changes by one unit
  - Least squares method

$$y = \alpha + \beta_1 x_1$$
Multiple linear regression

• Relation between a continuous variable and a set of i continuous variables

\[
y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i
\]

• Partial regression coefficients \( \beta_i \)
  • Amount by which \( y \) changes on average when \( x_i \) changes by one unit and all the other \( x \)s remain constant
  • Measures association between \( x_i \) and \( y \) adjusted for all other \( x \)

• Example
  • SBP versus age, weight, height, etc
Multiple linear regression

\[ y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i \]

- Predicted
- Response variable
- Outcome variable
- Dependent

- Predictor variables
- Explanatory variables
- Covariates
- Independent variables
Logistic regression (1)

Table 2  Age and signs of coronary heart disease (CD)

<table>
<thead>
<tr>
<th>Age</th>
<th>CD</th>
<th>Age</th>
<th>CD</th>
<th>Age</th>
<th>CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>41</td>
<td>1</td>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>58</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>47</td>
<td>0</td>
<td>60</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>62</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>65</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>67</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>71</td>
<td>1</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>51</td>
<td>1</td>
<td>77</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>52</td>
<td>0</td>
<td>81</td>
<td>1</td>
</tr>
</tbody>
</table>
How can we analyse these data?

• Compare mean age of diseased and non-diseased
  
  • Non-diseased: 38.6 years
  • Diseased: 58.7 years (p<0.0001)

• Linear regression?
Dot-plot: Data from Table 2

<table>
<thead>
<tr>
<th>AGE (years)</th>
<th>Signs of coronary disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>20</td>
<td>No</td>
</tr>
<tr>
<td>40</td>
<td>No</td>
</tr>
<tr>
<td>60</td>
<td>No</td>
</tr>
<tr>
<td>80</td>
<td>Yes</td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Logistic regression (2)

Table 3  Prevalence (%) of signs of CD according to age group

<table>
<thead>
<tr>
<th>Age group</th>
<th># in group</th>
<th>#</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 - 29</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30 - 39</td>
<td>6</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>40 - 49</td>
<td>7</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>50 - 59</td>
<td>7</td>
<td>4</td>
<td>57</td>
</tr>
<tr>
<td>60 - 69</td>
<td>5</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>70 - 79</td>
<td>2</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>80 - 89</td>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>
Dot-plot: Data from Table 3
Logistic function (1)

$$P(y|x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}}$$
Transformation

\[ P(y|x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}} \]

\[ \frac{P(y|x)}{1 - P(y|x)} \]

\[ \ln \left[ \frac{P(y|x)}{1 - P(y|x)} \right] = \alpha + \beta x \]

\( \alpha = \text{log odds of disease in unexposed} \)

\( \beta = \text{log odds ratio associated with being exposed} \)

\( e^\beta = \text{odds ratio} \)
Fitting the data

• Linear regression: Least squares
• Logistic regression: Maximum likelihood
• Likelihood function
  • Estimates parameters $\alpha$ and $\beta$
  • Practically easier to work with log-likelihood

$$L(B) = \ln[l(B)] = \sum_{i=1}^{n} \left\{ y_i \ln[\pi(x_i)] + (1 - y_i) \ln[1 - \pi(x_i)] \right\}$$
Maximum likelihood

• Iterative computing
  • Choice of an arbitrary value for the coefficients (usually 0)
  • Computing of log-likelihood
  • Variation of coefficients’ values
  • Reiteration until maximisation (plateau)

• Results
  • Maximum Likelihood Estimates (MLE) for $\alpha$ and $\beta$
  • Estimates of $P(y)$ for a given value of $x$
Multiple logistic regression

• More than one independent variable
  • Dichotomous, ordinal, nominal, continuous ...

\[
\ln \left( \frac{P}{1-P} \right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i
\]

• Interpretation of \( \beta_i \)
  • Increase in log-odds for a one unit increase in \( x_i \) with all the other \( x_i \)'s constant
  • Measures association between \( x_i \) and log-odds adjusted for all other \( x_i \)
The estimated logistic regression curve (red solid) is far away from the correct one (blue dashed) due to the existence of just one outlier (red circle)
Robust Linear Regression Against Data Poisoning Attack

Main Ideas: a two-phase solution

• Phase 1: Rely on dimension reduction (PCA) to prune non-principal noise in the training data

• Phase 2: In the low-dimensional space, learn a linear model (i.e., PCR)
Main Challenges

• Both of the two phases can be the target of the training data poisoning adversary

• Have no assumption on the ground truth distribution
  • ... except assuming they lie in a low-dimensional manifold
What Can Be Achieved

• Prove a **sufficient and necessary** condition on the **exact sub-space recovery** problem
  • Provides a criteria that the PCA process cannot be poisoned

• A **bound** on the **expected test error** when the training data is poisoned up to **$\gamma$ poisoning rate**
  • i.e., inject up to $\gamma N$ poisoning samples into the pristine training data of $N$ samples
Which line fits the data better?
Answer: democracy!
What about now?
Observation 1: When $\gamma \geq 1$, it is impossible to distinguish the poisoning samples from the pristine ones.
What is the mean of the data distribution?
How can a data poisoning adversary efficiently fool the mean estimator?
Answer: leveraging the pristine data!
Answer: leveraging the pristine data
Answer: leveraging the pristine data
Observation 2: the data poisoning adversary can fool a machine learning algorithm if and only if there is a portion of the pristine data that he can leverage
Sub-space Recovery Problem

• Problem Definition 1 (Subspace Recovery). Design an algorithm $L_{recovery}$, which takes as input $X$, and returns a set of vectors $B$ that form the basis of $X_*$

• Notation:
  • $X$: observed (poisoned) feature matrix
  • $X_*$: the pristine feature matrix
  • $X_0$: the pristine feature matrix with noise
    • $X_0 = X_* + noise$
Noise Residual and sub-matrix Residual

• Noise residual $NR(X_0)$ optimizes
  \[
  \min_{X'} ||X_0 - X'|| \\
  \text{s.t. } \text{rank}(X') \leq k
  \]

• Sub-matrix residual $SR(X_0)$ optimizes
  \[
  \min_{I,B,U} ||X_0^I - U\overline{B}|| \\
  \text{s.t. } \text{rank}(\overline{B}) = k, \overline{B}\overline{B}^T = I_k, X_*\overline{B}^T\overline{B} \neq X_* \\
  I \subseteq \{1, 2, \ldots, n\}, |I| = (1 - \gamma)N
Sufficient and Necessary Condition

• Theorem. If $SR(X_0) \leq NR(X_0)$, then no algorithm solves problem 1 with a probability greater than $1/2$.

• If $SR(X_0) > NR(X_0)$, then Algorithm 2 solves problem 1.

---

**Algorithm 2** Exact recovery algorithm for Problem 1

Solve the following optimization problem and get $\mathcal{I}$.

$$
\min_{\mathcal{I}, L} ||X^{\mathcal{I}} - L|| \\
\text{s.t. } \text{rank}(L) \leq k, \mathcal{I} \subseteq \{1, \ldots, n + n_1\}, |\mathcal{I}| = n
$$

(3)

**return** a basis of $X^{\mathcal{I}}$. 

---
Sub-space recover experiments (synthetic data)
Takeaways

*Message 1.* The poisoning attacker can leverage pristine data distribution to construct strong attacks

*Message 2.* When the poisoning ratio is not sufficiently large, we can bound the loss on the computed estimator.
Kalman Filtering

• Follow a point
• Follow a template
• Follow a changing template
• Follow all the elements of a moving person, fit a model to it.
• What are the dynamics of the thing being tracked?
• How is it observed?

\[
\hat{X}_{n|n} = E(X_n|Y_1,Y_2,\ldots,Y_n), \quad \hat{X}_{n+1|n} = E(X_{n+1}|Y_1,Y_2,\ldots,Y_n)
\]

\[
\hat{X}_{n|n} \xrightarrow{\text{Prediction}} \hat{X}_{n+1|n} \xrightarrow{\text{Correction}} \hat{X}_{n+1|n+1}
\]
Three main issues in tracking

- **Prediction:** we have seen $y_0, \ldots, y_{i-1}$ — what state does this set of measurements predict for the $i$'th frame? to solve this problem, we need to obtain a representation of $P(X_i | Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$.

- **Data association:** Some of the measurements obtained from the $i$-th frame may tell us about the object’s state. Typically, we use $P(X_i | Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$ to identify these measurements.

- **Correction:** now that we have $y_i$ — the relevant measurements — we need to compute a representation of $P(X_i | Y_0 = y_0, \ldots, Y_i = y_i)$. 
Simplifying Assumptions

- **Only the immediate past matters:** formally, we require
  \[ P(X_i|X_1,\ldots,X_{i-1}) = P(X_i|X_{i-1}) \]

  This assumption hugely simplifies the design of algorithms, as we shall see; furthermore, it isn’t terribly restrictive if we’re clever about interpreting \(X_i\) as we shall show in the next section.

- **Measurements depend only on the current state:** we assume that \(Y_i\) is conditionally independent of all other measurements given \(X_i\). This means that
  \[ P(Y_i,Y_j,\ldots,Y_k|X_i) = P(Y_i|X_i)P(Y_j,\ldots,Y_k|X_i) \]

  Again, this isn’t a particularly restrictive or controversial assumption, but it yields important simplifications.
Kalman filter graphical model and corresponding factorized joint probability

\[ P(x_1, x_2, x_3, y_1, y_2, y_3) = \\
\quad P(x_1)P(y_1 \mid x_1)P(x_2 \mid x_1)P(y_2 \mid x_2)P(x_3 \mid x_2)P(y_3 \mid x_3) \]
Tracking as induction

• Make a measurement starting in the 0th frame
• Then: assume you have an estimate at the ith frame, after the measurement step.
• Show that you can do prediction for the i+1th frame, and measurement for the i+1th frame.
Prediction step

**Prediction**

Prediction involves representing

\[ P(X_i | y_0, \ldots, y_{i-1}) \]

given

\[ P(X_{i-1} | y_0, \ldots, y_{i-1}). \]

Our independence assumptions make it possible to write

\[
P(X_i | y_0, \ldots, y_{i-1}) = \int P(X_i, X_{i-1} | y_0, \ldots, y_{i-1}) dX_{i-1}
\]

\[
= \int P(X_i | X_{i-1}, y_0, \ldots, y_{i-1}) P(X_{i-1} | y_0, \ldots, y_{i-1}) dX_{i-1}
\]

\[
= \int P(X_i | X_{i-1}) P(X_{i-1} | y_0, \ldots, y_{i-1}) dX_{i-1}
\]
Correction step

Correction involves obtaining a representation of

\[ P(X_i|y_0, \ldots, y_i) \]

given

\[ P(X_i|y_0, \ldots, y_{i-1}) \]

Our independence assumptions make it possible to write

\[
P(X_i|y_0, \ldots, y_i) = \frac{P(X_i, y_0, \ldots, y_i)}{P(y_0, \ldots, y_i)} \\
= \frac{P(y_i|X_i, y_0, \ldots, y_{i-1})P(X_i, y_0, \ldots, y_{i-1})P(y_0, \ldots, y_{i-1})}{P(y_0, \ldots, y_i)} \\
= P(y_i|X_i)P(X_i|y_0, \ldots, y_{i-1}) \frac{P(y_0, \ldots, y_{i-1})}{P(y_0, \ldots, y_i)} \\
= \frac{P(y_i|X_i)P(X_i|y_0, \ldots, y_{i-1})}{\int P(y_i|X_i)P(X_i|y_0, \ldots, y_{i-1})dX_i}
\]
The Kalman Filter

• Key ideas:
  • Linear models interact uniquely well with Gaussian noise - make the prior Gaussian, everything else Gaussian and the calculations are easy
  • Gaussians are really easy to represent --- once you know the mean and covariance, you’re done
Recall the three main issues in tracking

- **Prediction:** we have seen $y_0, \ldots, y_{i-1}$ — what state does this set of measurements predict for the $i$’th frame? to solve this problem, we need to obtain a representation of $P(X_i | Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$.

- **Data association:** Some of the measurements obtained from the $i$-th frame may tell us about the object’s state. Typically, we use $P(X_i | Y_0 = y_0, \ldots, Y_{i-1} = y_{i-1})$ to identify these measurements.

- **Correction:** now that we have $y_i$ — the relevant measurements — we need to compute a representation of $P(X_i | Y_0 = y_0, \ldots, Y_i = y_i)$. 

The Kalman Filter
Example: The Kalman Filter in 1D

• Dynamic Model

\[ x_i \sim N(d_i x_{i-1}, \sigma_{d_i}^2) \]

\[ y_i \sim N(m_i x_i, \sigma_{m_i}^2) \]

• Notation

Predicted mean

Corrected mean

mean of \( P(X_i | y_0, \ldots, y_{i-1}) \) as \( \overline{X}_i^- \)

mean of \( P(X_i | y_0, \ldots, y_i) \) as \( \overline{X}_i^+ \)

the standard deviation of \( P(X_i | y_0, \ldots, y_{i-1}) \) as \( \sigma_i^- \)

of \( P(X_i | y_0, \ldots, y_i) \) as \( \sigma_i^+ \)
The Kalman Filter
Prediction for 1D Kalman filter

• The new state is obtained by
  • multiplying old state by known constant
  • adding zero-mean noise

\[ x_i \sim N(d_i x_{i-1}, \sigma_{d_i}^2) \]

• Therefore, predicted mean for new state is
  • constant times mean for old state

• Old variance is normal random variable
  • variance is multiplied by square of constant
  • and variance of noise is added.

\[ \bar{X}_i = d_i \bar{X}_{i-1} \quad \quad (\sigma_i^-)^2 = \sigma_{d_i}^2 + (d_i \sigma_{i-1}^+)^2 \]
The Kalman Filter

Time Update ("Predict")

Measurement Update ("Correct")
Measurement update for 1D Kalman filter

\[ x_i^+ = \left( \frac{\bar{x}_i \sigma_{m_i}^2 + m_i y_i (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right) \]

\[ \sigma_i^+ = \sqrt{\left( \frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right) } \]

Notice:
- if measurement noise is small, we rely mainly on the measurement;
- if it’s large, mainly on the prediction
- \( \sigma \) does not depend on \( y \)
Dynamic Model:

\[ x_i \sim N(d_i x_{i-1}, \sigma_{d_i}) \]

\[ y_i \sim N(m_i x_i, \sigma_{m_i}) \]

Start Assumptions: \( \bar{x}_0^- \) and \( \sigma_0^- \) are known

Update Equations: Prediction

\[ \bar{x}_i^- = d_i \bar{x}_{i-1}^+ \]

\[ \sigma_i^- = \sqrt{\sigma_{d_i}^2 + (d_i \sigma_{i-1}^+)^2} \]

Update Equations: Correction

\[ x_i^+ = \left( \frac{\bar{x}_i^- \sigma_{m_i}^2 + m_i y_i (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right) \]

\[ \sigma_i^+ = \sqrt{\left( \frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2} \right)} \]
What happens if the x dynamics are given a non-zero variance?
Dynamic Model:

\[ x_i \sim N(d_i x_{i-1}, \sigma_{d_i}) \]

\[ y_i \sim N(m_i x_i, \sigma_{m_i}) \]

Start Assumptions: \( \bar{x}_0^- \) and \( \sigma_0^- \) are known

Initial conditions:

\[ \bar{x}_0^- = 0 \quad \sigma_0^- = \infty \]

Update Equations: Prediction

\[ \bar{x}_i^- = d_i \bar{x}_{i-1}^+ \]

\[ \sigma_i^- = \sqrt{\sigma_{d_i}^2 + (d_i \sigma_{d_{i-1}}^+)^2} \]

Iteration 0 1 2

<table>
<thead>
<tr>
<th>( \bar{x}_i^- )</th>
<th>( y_0 )</th>
<th>( \frac{y_0 + 2y_1}{3} )</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>( \bar{x}_i^+ )</th>
<th>( y_0 )</th>
<th>( \frac{y_0 + 2y_1}{3} )</th>
<th>( \frac{y_0 + 2y_1 + 5y_2}{8} )</th>
</tr>
</thead>
</table>

Update Equations: Correction

\[ x_i^+ = \left( \frac{\sigma_{m_i}^2 + m_i y_i \sigma_i^-}{\sigma_{m_i}^2 + m_i^2 \sigma_i^-} \right) \]

\[ \sigma_i^- = \sqrt{\frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2}} \]

\[ \sigma_i^+ = \sqrt{\frac{\sigma_{m_i}^2 (\sigma_i^-)^2}{\sigma_{m_i}^2 + m_i^2 (\sigma_i^-)^2}} \]

\[ \sigma_i^- \quad \infty \quad \sqrt{2} \quad \frac{\sqrt{5}}{\sqrt{3}} \]

\[ \sigma_i^+ \quad 1 \quad \frac{\sqrt{2}}{\sqrt{3}} \quad \frac{\sqrt{5}}{\sqrt{8}} \]
Linear dynamic models

• A linear dynamic model has the form

\[ x_i = \mathcal{N}(D_{i-1}x_{i-1}; \Sigma_{d_i}) \]

\[ y_i = \mathcal{N}(M_ix_i; \Sigma_{m_i}) \]

• This is much, much more general than it looks, and extremely powerful
Examples of linear state space models

• Drifting points
  • assume that the new position of the point is the old one, plus noise

\[ D = \text{Identity} \]

\[
x_i = N(D_{i-1} x_{i-1}; \Sigma_{d_i})
\]

\[
y_i = N(M_i x_i; \Sigma_{m_i})
\]
Constant velocity

- We have
  \[ u_i = u_{i-1} + \Delta t v_{i-1} + \epsilon_i \]
  \[ v_i = v_{i-1} + \zeta_i \]

- Stack \((u, v)\) into a single state vector

\[
\begin{pmatrix}
  u \\
v
\end{pmatrix}_i =
\begin{pmatrix}
  1 & \Delta t \\
  0 & 1
\end{pmatrix}
\begin{pmatrix}
  u \\
v
\end{pmatrix}_{i-1} + \text{noise}
\]

- which is the form we had above
Constant Velocity Model
Constant acceleration

- We have
  \[ u_i = u_{i-1} + \Delta t v_{i-1} + \epsilon_i \]
  \[ v_i = v_{i-1} + \Delta t a_{i-1} + \zeta_i \]
  \[ a_i = a_{i-1} + \xi_i \]

  (the Greek letters denote noise terms)

- Stack \((u, v)\) into a single state vector

\[
\begin{pmatrix} u \\ v \\ a_i \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ a_{i-1} \end{pmatrix} + \text{noise}
\]

- which is the form we had above

\[ \mathbf{x}_i = N(\mathbf{D}_{i-1}\mathbf{x}_{i-1}; \Sigma_{d_i}) \]
\[ \mathbf{y}_i = N(\mathbf{M}_i\mathbf{x}_i; \Sigma_{m_i}) \]
Constant Acceleration Model
Periodic motion

\[ x_i = N(D_{i-1}x_{i-1}; \Sigma_{d_i}) \]
\[ y_i = N(M_i x_i; \Sigma_{m_i}) \]

Assume we have a point, moving on a line with a periodic movement defined with a differential eq:

\[ \frac{d^2p}{dt^2} = -p \]

can be defined as

\[ \frac{du}{dt} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} u = Su \]

with state defined as stacked position and velocity \( u=(p, v) \)
Periodic motion

\[ x_i = N(D_{i-1}x_{i-1}; \Sigma_d) \]

\[ y_i = N(M_ix_i; \Sigma_m) \]

\[
\frac{du}{dt} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} u = Su
\]

Take discrete approximation....(e.g., forward Euler integration with \( \Delta t \) stepsize.)

\[
u_i = u_{i-1} + \Delta t \frac{du}{dt}
\]

\[
= u_{i-1} + \Delta t Su_{i-1}
\]

\[
= \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} u_{i-1}
\]
n-D Prediction

Generalization to n-D is straightforward but more complex.

Prediction:
• Multiply estimate at prior time with forward model:
\[
\bar{x}_i^- = D_i \bar{x}_i^{+1}
\]
• Propagate covariance through model and add new noise:
\[
\Sigma_i^- = \Sigma_i + D_i \sigma_{i-1} \sigma_i^+ D_i
\]
n-D Correction

Generalization to n-D is straightforward but more complex.

Correction:
• Update *a priori* estimate with measurement to form *a posteriori*
n-D correction

Find linear filter on innovations

$$\bar{x}_i^+ = \bar{x}_i^- + K_i \left[ y_i - M_i \bar{x}_i^- \right]$$

which minimizes a posteriori error covariance:

$$E \left[ (x - \bar{x}^+)^T (x - \bar{x}^+) \right]$$

K is the Kalman Gain matrix. A solution is

$$K_i = \Sigma_i^- M_i^T \left[ M_i \Sigma_i^- M_i^T + \Sigma_m \right]^{-1}$$
Kalman Gain Matrix

\[
\bar{x}_i^+ = \bar{x}_i^- + K_i \left[ y_i - M_i \bar{x}_i^- \right]
\]

\[
K_i = \Sigma_i^- M_i^T \left[ M_i \Sigma_i^- M_i^T + \Sigma_m \right]^{-1}
\]

As measurement becomes more reliable, K weights residual more heavily,

\[
\lim_{\Sigma_i \to 0} K_i = M^{-1}
\]

As prior covariance approaches 0, measurements are ignored:

\[
\lim_{\Sigma_m \to 0} K_i = 0
\]
Dynamic Model:

\[ x_i \sim N(D_i x_{i-1}, \Sigma_{d_i}) \]

\[ y_i \sim N(M_i x_i, \Sigma_{m_i}) \]

Start Assumptions: \( \mathbf{\Sigma_0} \) and \( \mathbf{\Sigma_0} \) are known

Update Equations: Prediction

\[ \mathbf{\Sigma_{i-1}} = \mathbf{\Sigma_{d_i}} + \mathbf{D}_i \mathbf{\Sigma_{i-1}} \mathbf{D}_i^T \]

Update Equations: Correction

\[ \mathbf{\Sigma_{i-1}} = \mathbf{\Sigma_{i-1}} \mathbf{M}_i^T \left[ \mathbf{M}_i \mathbf{\Sigma_{i-1}} \mathbf{M}_i^T + \mathbf{\Sigma}_{m_i} \right]^{-1} \]

\[ \mathbf{\Sigma_{i-1}}^+ = \mathbf{\Sigma_{i-1}} + \mathbf{\Sigma_{i-1}} \mathbf{K}_i \mathbf{[y_i - M_i x_i]} \]

\[ \mathbf{\Sigma_{i-1}}^+ = (\mathbf{I}_d - \mathbf{K}_i \mathbf{M}_i) \mathbf{\Sigma_{i-1}} \]
Constant Velocity Model
Smoothing

- Idea
  - We don’t have the best estimate of state - what about the future?
  - Run two filters, one moving forward, the other backward in time.
  - Now combine state estimates
    - The crucial point here is that we can obtain a smoothed estimate by viewing the backward filter’s prediction as yet another measurement for the forward filter.
Forward estimates.

The o-s give state, x-s measurement.

The *-s give \( \overline{x}_i^- \), + -s give \( \overline{x}_i^+ \), vertical bars are 3 standard deviation bars
Backward estimates.

The o-s give state, x-s measurement.

The *-s give $\bar{x}_{i}^{-}$, +-s give $\bar{x}_{i}^{+}$, vertical bars are 3 standard deviation bars.
Combined forward-backward estimates.

The o-s give state, x-s measurement.

The *-s give $\overline{x}_i^-$, + -s give $\overline{x}_i^+$, vertical bars are 3 standard deviation bars.
Multiple model filters

Test several models of assumed dynamics
MM estimate

Two models: Position (P), Position+Velocity (PV)
Non-toy image representation

Phase of a steerable quadrature pair (G2, H2). Steered to 4 different orientations, at 2 scales.
Representing Distributions using Weighted Samples

Rather than a parametric form, use a set of samples to represent a density:
Representing Distributions using Weighted Samples

Rather than a parametric form, use a set of samples to represent a density:

\[ \{(u^i, w^i)\} \]

Sample positions | Probability mass at each sample

This gives us two knobs to adjust when representing a probability density by samples: the locations of the samples, and the probability weight on each sample.
Representing distributions using weighted samples, another picture
Sampled representation of a probability distribution

Represent a probability distribution

\[ p_f(X) = \frac{f(X)}{\int f(U) \, dU} \]

by a set of \( N \) weighted samples

\[ \{(u^i, w^i)\} \]

where \( u^i \sim s(u) \) and \( w^i = f(u^i) / s(u^i) \).

You can also think of this as a sum of dirac delta functions, each of weight \( w \):

\[ p_f(x) = \sum_i w^i \delta(x - u^i) \]
Tracking, in particle filter representation

\[ P(x_n | y_1 ... y_n) = k \ P(y_n | x_n) \int dx_{n-1} \ P(x_n | x_{n-1}) \ P(x_{n-1} | y_1 ... y_{n-1}) \]

\[ p_f(x) = \sum_i w^i \delta(x - u^i) \]
Applications

Tracking
  • hands  
  • bodies  
  • Leaves

What might we expect?
  Reliable, robust, slow
Contour tracking

[Isard 1998]
Head tracking

Picture of the states represented by the top weighted particles

The mean state

[Isard 1998]
Leaf tracking

[Isard 1998]