Bayesian Methods for Data Analysis in Software Engineering

Bayesian Classification and Regression

Mohan Sridharan[1] Akbar Siami Namin[2]

[1]Stochastic Estimation and Autonomous Robotics (SEAR) Lab
Department of Computer Science
Texas Tech University

[2]AdVanced Empirical Software Testing and Analysis (AVESTA) Research Group
Department of Computer Science
Texas Tech University

May 3, 2010; Cape Town

Session 2: Classification, Regression and Inference

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Session 2: Bayesian Classification

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Classification Basics

- Broad categories: supervised (labeled samples); unsupervised (no labeled samples).
- Group data based on similarity measures.
- Several sophisticated techniques exist:
  - Supervised: decision trees, support vector machines, naive Bayes.
  - Unsupervised: nearest neighbors, clustering.
- Choice of classifier based on data and application.
- *Probabilistic methods explicitly model the noise in input data!*

Classification Basics

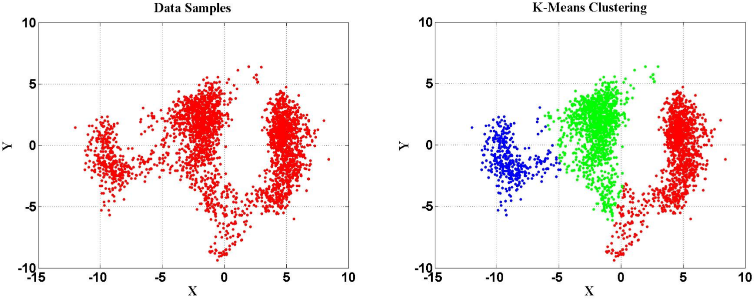
- Broad categories: supervised (labeled samples); unsupervised (no labeled samples).
- Group data based on similarity measures.
- Several sophisticated techniques exist:
  - Supervised: decision trees, support vector machines, naive Bayes.
  - Unsupervised: nearest neighbors, clustering.
- Choice of classifier based on data and application.
- *Probabilistic methods explicitly model the noise in input data!*

Classification Basics

- Broad categories: supervised (labeled samples); unsupervised (no labeled samples).
- Group data based on similarity measures.
- Several sophisticated techniques exist:
  - Supervised: decision trees, support vector machines, naive Bayes.
  - Unsupervised: nearest neighbors, clustering.
- Choice of classifier based on data and application.
- *Probabilistic methods explicitly model the noise in input data!*

Clustering Data Samples

- K-Means clustering of input data samples.
- Data grouped into three clusters.



Bayesian Classification

- Bayes' rule (once again):

$$p(x, y) = p(x|y) \cdot p(y) = p(y|x) \cdot p(x) \quad (1)$$

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)} = \frac{\text{likelihood . prior}}{\text{normalizer}}$$

- Classify based on Bayes *decision rule*:

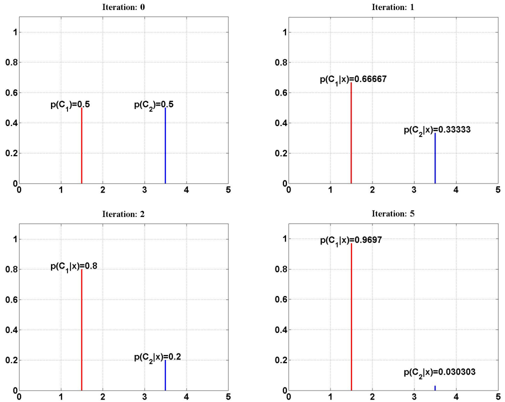
$$p(w_1|x) > p(w_2|x) \implies \text{choose } w_1\text{; else choose } w_2 \quad (2)$$

- Decision rule extends to multiple classes:

$$p(w_i|x) > p(w_j|x) \;\; \forall j \neq i \implies \text{choose } w_i \quad (3)$$

Illustrative Example 1

- C_1 : *fault*; C_2 : ¬*fault*; x : *data*.
- $p(C_1) = p(C_2) = 0.5;\ p(x|C_1) = 0.6;\ p(x|C_2) = 0.3$



Multi-Class Extension

- Model *likelihoods* and *priors* based on training samples.
- Update belief incrementally based on evidence.
- Use multi-class Decision rule:

$$p(w_i|x) > p(w_j|x) \ \ \forall j \neq i \implies \text{choose } w_i \tag{4}$$

- Question: *what representation to use to model the likelihoods?*
- Answer: *Typically, functions with well-understood properties are used* – e.g. Gaussians.

Multi-Class Extension

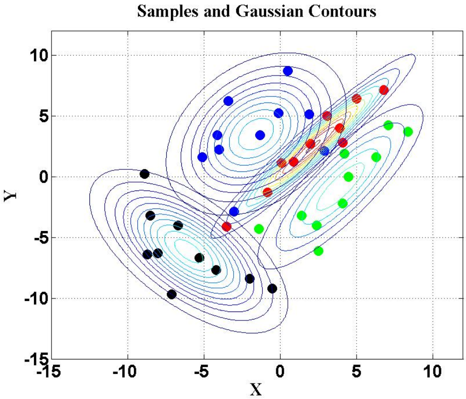
- Model *likelihoods* and *priors* based on training samples.
- Update belief incrementally based on evidence.
- Use multi-class Decision rule:

$$p(w_i|x) > p(w_j|x) \;\; \forall j \neq i \implies \text{choose } w_i \tag{4}$$

- Question: *what representation to use to model the likelihoods?*
- Answer: *Typically, functions with well-understood properties are used* – e.g. Gaussians.

Illustrative Example 2

- Four-class problem; ten training data samples per class.
- Model individual class likelihoods as Gaussians.



Illustrative Example 2: Modeling

- Compute Gaussian means and covariances:

$$\mu_1 = [2.16, 2.49]; \quad \mu_2 = [3.95, -0.84] \tag{5}$$
$$\mu_3 = [-1.57, 3.5]; \quad \mu_4 = [-6, -6.14]$$

$$\Sigma_1 = \begin{pmatrix} 9.32 & 10.12 \\ 10.12 & 11.85 \end{pmatrix}$$

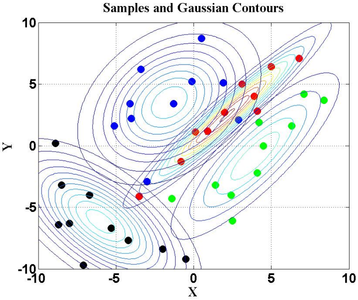
$$\Sigma_2 = \begin{pmatrix} 8.36 & 8.87 \\ 8.87 & 13.02 \end{pmatrix}$$

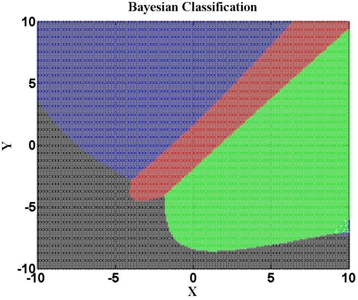
$$\Sigma_3 = \begin{pmatrix} 7.63 & 2.98 \\ 2.98 & 9.78 \end{pmatrix}$$

$$\Sigma_4 = \begin{pmatrix} 8.62 & -5.71 \\ -5.71 & 9.26 \end{pmatrix}$$

Illustrative Example 2: Classification

- Decision boundaries for all four classes:





Summary

- Elegant belief update and decision rule for classification.
- Bayes error: *minimum classification error that cannot be eliminated.*
- Little or no tuning of arbitrary thresholds.
- Challenge 1: *what functional form and parameters to use for modeling likelihoods and priors?*
- Challenge 2: *how to obtain enough data to model the likelihoods and priors?*
- Demo: *Matlab-based comparison with other classifiers.*

Summary

- Elegant belief update and decision rule for classification.
- Bayes error: *minimum classification error that cannot be eliminated.*
- Little or no tuning of arbitrary thresholds.
- Challenge 1: *what functional form and parameters to use for modeling likelihoods and priors?*
- Challenge 2: *how to obtain enough data to model the likelihoods and priors?*
- Demo: *Matlab-based comparison with other classifiers.*

Summary

- Elegant belief update and decision rule for classification.
- Bayes error: *minimum classification error that cannot be eliminated.*
- Little or no tuning of arbitrary thresholds.
- Challenge 1: *what functional form and parameters to use for modeling likelihoods and priors?*
- Challenge 2: *how to obtain enough data to model the likelihoods and priors?*
- Demo: *Matlab-based comparison with other classifiers.*

For more information

- C. Bishop. *Pattern Recognition and Machine Learning*. Springer publishing house, 2007.
- D. Stork and E. Yom-Tov. *Computer Manual in MATLAB to accompany Pattern Classification*. Wiley-Interscience, 2004.
- R. Duda and P. Hart and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.
- *Weka 3: Data Mining Software in Java*, 2010. http://www.cs.waikato.ac.nz/ml/weka/.
- *Matlab Statistics Toolbox 7.3*, 2010. http://www.mathworks.com/products/statistics/

Session 2: Bayesian Regression

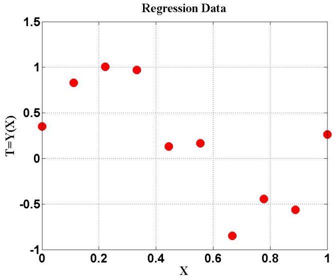
- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

Regression Basics

- Consider polynomial curve fitting of target variable t:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \quad (6)$$

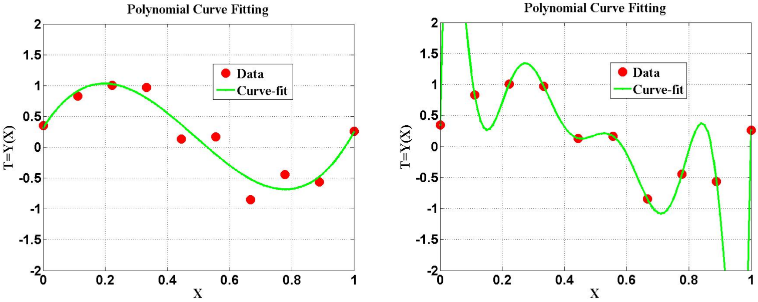
- Consider data sampled from a sinusoidal waveform:



- Can use polynomials of different degrees.

Illustrative Example 1

- Polynomial curve fitting of data: best performance for degree $= 3$.



- However *over-fitting* can lead to problems.

Regularization

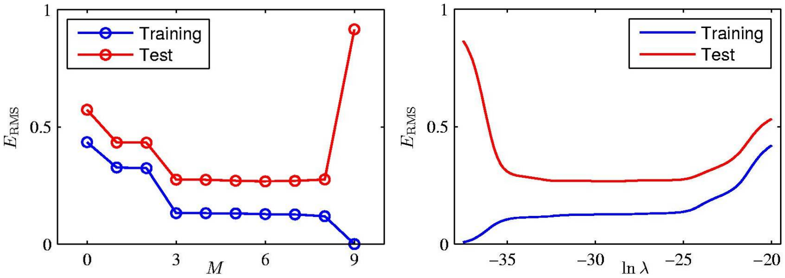
- Regularization in sum-of-squares error function:

$$
\begin{aligned}
E(\boldsymbol{w}) &= E_D(\boldsymbol{w}) + \lambda E_w(\boldsymbol{w}) \\
&= \frac{1}{2}\sum_{n=1}^{N}\{t_n - y(x_n, \boldsymbol{w})\}^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2
\end{aligned} \tag{7}
$$

- λ is the regularization co-efficient. Models cost of over-fitting.
- Demo: Matlab-based curve-fitting toolbox.

RMS Errors

- Standard vs. regularized performance:



Regularization Parameter Tuning

- Polynomial co-efficients as a function of the regularization parameter:

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

- $ln(\lambda) = -\infty$: no regularization.

Basis Functions

- Model curve fitting using basis functions:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \Phi(\boldsymbol{x}) \tag{8}$$

- The $\phi_j(\boldsymbol{x})$ are the *basis functions*.
- Normally $\phi_0(\boldsymbol{x}) = 1$ i.e. w_0 is the *bias*.
- Polynomial functions: $\phi_d(\boldsymbol{x}) = x^d$

Basic Bayesian Approach

- Assume a zero-mean Gaussian noise model:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon \tag{9}$$
$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), 1/\beta)$$

- Extension to data set with N samples: $X = \{x_1, \dots, x_N\}$ with target values: $t_1, \dots, t_N$:

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_i|\boldsymbol{w}^T \phi(x_i), \frac{1}{\beta}) \tag{10}$$

Maximum Likelihood Estimation

- Compute the log likelihood:

$$ln\, p(\mathbf{t}|\mathbf{w}, \beta) = \frac{N}{2} ln(\beta) - \frac{N}{2} ln(2\pi) - \beta E_D(\mathbf{w}) \tag{11}$$

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \phi(x_i)\}^2$$

- Partial differentials of the log likelihood provides maximum likelihood estimates of the parameters: $\mathbf{w}_{ML}$, β_{ML}
- Extends to multiple outputs, incremental updates and regularized least squares.

Bayesian vs. Frequentist

- Consider curve-fitting with observed data $\mathcal{D} = \{t_1, \ldots, t_N\}$ and parameter values $\boldsymbol{w}$.
- Frequentist and Bayesian: estimate $p(\mathcal{D}|\boldsymbol{w})$.
- Frequentist approach (MLE): $\boldsymbol{w}$ is chosen to maximize $p(\mathcal{D}|\boldsymbol{w})$. Error bars obtained by considering distribution of data sets $\mathcal{D}$.
- Bayesian approach: only one data set $\mathcal{D}$ available. Uncertainty in parameters expressed using probability distribution of $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\mathcal{D})} \tag{12}$$

- Avoids over-fitting, uses training data for model selection.

Bayesian vs. Frequentist

- Consider curve-fitting with observed data $\mathcal{D} = \{t_1, \dots, t_N\}$ and parameter values $\boldsymbol{w}$.
- Frequentist and Bayesian: estimate $p(\mathcal{D}|\boldsymbol{w})$.
- Frequentist approach (MLE): $\boldsymbol{w}$ is chosen to maximize $p(\mathcal{D}|\boldsymbol{w})$. Error bars obtained by considering distribution of data sets $\mathcal{D}$.
- Bayesian approach: only one data set $\mathcal{D}$ available. Uncertainty in parameters expressed using probability distribution of $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\mathcal{D})} \tag{12}$$

- Avoids over-fitting, uses training data for model selection.

Bayesian vs. Frequentist

- Consider curve-fitting with observed data $\mathcal{D} = \{t_1, \ldots, t_N\}$ and parameter values $\boldsymbol{w}$.
- Frequentist and Bayesian: estimate $p(\mathcal{D}|\boldsymbol{w})$.
- Frequentist approach (MLE): $\boldsymbol{w}$ is chosen to maximize $p(\mathcal{D}|\boldsymbol{w})$. Error bars obtained by considering distribution of data sets $\mathcal{D}$.
- Bayesian approach: only one data set $\mathcal{D}$ available. Uncertainty in parameters expressed using probability distribution of $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\mathcal{D})} \tag{12}$$

- Avoids over-fitting, uses training data for model selection.

Bayesian vs. Frequentist

- Consider curve-fitting with observed data $\mathcal{D} = \{t_1, \ldots, t_N\}$ and parameter values $\boldsymbol{w}$.
- Frequentist and Bayesian: estimate $p(\mathcal{D}|\boldsymbol{w})$.
- Frequentist approach (MLE): $\boldsymbol{w}$ is chosen to maximize $p(\mathcal{D}|\boldsymbol{w})$. Error bars obtained by considering distribution of data sets $\mathcal{D}$.
- Bayesian approach: only one data set $\mathcal{D}$ available. Uncertainty in parameters expressed using probability distribution of $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\mathcal{D})} \tag{12}$$

- Avoids over-fitting, uses training data for model selection.

Bayesian vs. Frequentist

- Consider curve-fitting with observed data $\mathcal{D} = \{t_1, \ldots, t_N\}$ and parameter values $\boldsymbol{w}$.
- Frequentist and Bayesian: estimate $p(\mathcal{D}|\boldsymbol{w})$.
- Frequentist approach (MLE): $\boldsymbol{w}$ is chosen to maximize $p(\mathcal{D}|\boldsymbol{w})$. Error bars obtained by considering distribution of data sets $\mathcal{D}$.
- Bayesian approach: only one data set $\mathcal{D}$ available. Uncertainty in parameters expressed using probability distribution of $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})\, p(\boldsymbol{w})}{p(\mathcal{D})} \tag{12}$$

- Avoids over-fitting, uses training data for model selection.

References

- C. Bishop. *Pattern Recognition and Machine Learning*. Springer publishing house, 2007.
- R. Duda and P. Hart and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.
- *Matlab Statistics Toolbox 7.3*, 2010. `http://www.mathworks.com/products/statistics/`

Session 2: Bayesian Inference

- **9.00–10.30**:
  - Introduction.
  - Statistical analysis; hypothesis testing.
  - Basic probability, Bayes' rule.
- **11.00–12.30**:
  - Bayesian classification.
  - Bayesian regression.
  - Bayesian inference.
- **14.00–15.30**:
  - Information theory.
  - Stochastic sampling.
- **16.00–17.30**:
  - Markov decision processes.
  - Partially observable Markov decision processes.
  - Discussion.

The Framework

- Inputs:
  - Stream of observations z and actions u: $\{u_1, z_1, \ldots, u_t, z_t\}$
  - Sensor model: $p(z|x)$
  - Action model: $p(x'|u, x)$
  - Prior probability of system state: $p(x)$
- Outputs:
  - Estimate the state $\boldsymbol{x}$ of a *dynamical system*.
  - Posterior of state, called the belief:

$$bel(x_t) = p(x_t|u_1, z_1, \ldots, u_t, z_t) \tag{13}$$

Markov Assumption

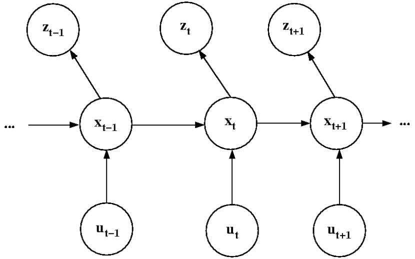
- First-order Markov assumption:

$$p(x_t | x_0, \ldots, x_{t-1}) = p(x_t | x_{t-1}) \tag{14}$$

- Bayesian filtering:

$$p(z_t | x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t | x_t) \tag{15}$$

$$p(x_t | x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t | x_{t-1}, u_t)$$



Bayes Filters 1

- Bayes rule:

$$\begin{aligned} bel(x_t) &= p(x_t | u_{1:t}, z_{1:t}) \\ &\propto p(z_t | x_t, u_1, z_1, \ldots, u_t)\ p(x_t | u_1, z_1, \ldots, u_t) \end{aligned} \tag{16}$$

- Markov assumption:

$$\begin{aligned} bel(x_t) &\propto p(z_t | x_t, u_1, z_1, \ldots, u_t)\ p(x_t | u_1, z_1, \ldots, u_t) \\ &= p(z_t | x_t)\ p(x_t | u_1, z_1, \ldots, u_t) \end{aligned} \tag{17}$$

Bayes Filters 1

- Bayes rule:

$$\begin{aligned} bel(x_t) &= p(x_t|u_{1:t}, z_{1:t}) \\ &\propto p(z_t|x_t, u_1, z_1, \ldots, u_t)\ p(x_t|u_1, z_1, \ldots, u_t) \end{aligned} \tag{16}$$

- Markov assumption:

$$\begin{aligned} bel(x_t) &\propto p(z_t|x_t, u_1, z_1, \ldots, u_t)\ p(x_t|u_1, z_1, \ldots, u_t) \\ &= p(z_t|x_t)\ p(x_t|u_1, z_1, \ldots, u_t) \end{aligned} \tag{17}$$

Bayes Filters 2

- Probability expansion:

$$bel(x_t) \propto p(z_t|x_t)\, p(x_t|u_1, z_1, \ldots, u_t) \quad (18)$$
$$= p(z_t|x_t) \int p(x_t|u_{1:t}, z_{1:t-1}, x_{t-1}) p(x_{t-1}|u_{1:t}, z_{1:t-1})\, dx_{t-1}$$

- Markov assumption:

$$bel(x_t) \propto p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, u_t)\, dx_{t-1} \quad (19)$$

Bayes Filters 2

- Probability expansion:

$$\begin{aligned} bel(x_t) &\propto p(z_t|x_t)\, p(x_t|u_1, z_1, \ldots, u_t) \\ &= p(z_t|x_t) \int p(x_t|u_{1:t}, z_{1:t-1}, x_{t-1}) p(x_{t-1}|u_{1:t}, z_{1:t-1})\, dx_{t-1} \end{aligned} \tag{18}$$

- Markov assumption:

$$bel(x_t) \propto p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, u_t)\, dx_{t-1} \tag{19}$$

Bayes Filters 3

- Markov assumption:

$$bel(x_t) \propto p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, u_t)\, dx_{t-1} \tag{20}$$

$$= p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, z_{t-1})\, dx_{t-1}$$

- Recursion:

$$bel(x_t) = \eta\, p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1} \tag{21}$$

Bayes Filters 3

- Markov assumption:

$$bel(x_t) \propto p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, u_t)\, dx_{t-1} \tag{20}$$

$$= p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, z_{t-1})\, dx_{t-1}$$

- Recursion:

$$bel(x_t) = \eta\, p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1} \tag{21}$$

Bayes Filters Summary

- Recursive belief update based on Markov assumption:

$$
\begin{aligned}
bel(x_t) &= p(x_t|u_{1:t}, z_{1:t}) \qquad (22)\\
&\propto p(z_t|x_t, u_1, z_1, \ldots, u_t)\, p(x_t|u_1, z_1, \ldots, u_t)\\
&= p(z_t|x_t)\, p(x_t|u_1, z_1, \ldots, u_t)\\
&= p(z_t|x_t) \int p(x_t|u_{1:t}, z_{1:t-1}, x_{t-1}) p(x_{t-1}|u_{1:t}, z_{1:t-1})\, dx_{t-1}\\
&= p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, u_t)\, dx_{t-1}\\
&= p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_1, z_1, \ldots, z_{t-1})\, dx_{t-1}
\end{aligned}
$$

$$
bel(x_t) = \eta\, p(z_t|x_t) \int p(x_t|u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}
$$

Bayes Inference

- Bayes prediction and correction:

$$\forall x_t : \; bel(x_t) = \eta \; p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}$$

$$\forall k : \; p_{k,t} = \eta \; p(z_t|X_t = x_k) \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1}$$

- Bayes filter:

$$\forall x_t : \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1} \tag{23}$$

$$bel(x_t) = \eta \; p(z_t|x_t) \, \overline{bel}(x_t)$$

- Discrete Bayes filter:

$$\forall k : \overline{p}_{k,j} = \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1} \tag{24}$$

$$p_{k,j} = \eta \; p(z_t|X_t = x_k) \, \overline{p}_{k,j}$$

Bayes Inference

- Bayes prediction and correction:

$$\forall x_t : \; bel(x_t) = \eta \; p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}$$

$$\forall k : \; p_{k,t} = \eta \; p(z_t|X_t = x_k) \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1}$$

- Bayes filter:

$$\forall x_t : \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1} \qquad (23)$$

$$bel(x_t) = \eta \; p(z_t|x_t) \, \overline{bel}(x_t)$$

- Discrete Bayes filter:

$$\forall k : \overline{p}_{k,j} = \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1} \qquad (24)$$

$$p_{k,j} = \eta \; p(z_t|X_t = x_k) \, \overline{p}_{k,j}$$

Bayes Inference

- Bayes prediction and correction:

$$\forall x_t: \;\; bel(x_t) = \eta \; p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}$$

$$\forall k: \;\; p_{k,t} = \eta \; p(z_t|X_t = x_k) \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1}$$

- Bayes filter:

$$\forall x_t: \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1} \tag{23}$$

$$bel(x_t) = \eta \; p(z_t|x_t) \, \overline{bel}(x_t)$$

- Discrete Bayes filter:

$$\forall k: \overline{p}_{k,j} = \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i) \, p_{i,t-1} \tag{24}$$

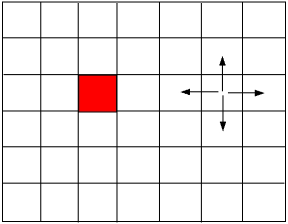
$$p_{k,j} = \eta \; p(z_t|X_t = x_k) \, \overline{p}_{k,j}$$

Examples

- Pictorial representation of discrete Bayes:

$$\forall k : \overline{p}_{k,j} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i)\, p_{i,t-1} \qquad (25)$$

$$p_{k,j} = \eta\, p(z_t | X_t = x_k)\, \overline{p}_{k,j}$$



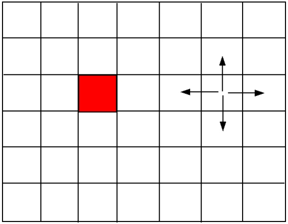
- Kalman filters, Particle filters, Bayesian Networks, Partially Observable Markov Decision Processes (POMDPs), Hidden Markov Models (HMMs) and many more!

Examples

- Pictorial representation of discrete Bayes:

$$\forall k : \overline{p}_{k,j} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i)\, p_{i,t-1} \tag{25}$$

$$p_{k,j} = \eta \; p(z_t | X_t = x_k)\, \overline{p}_{k,j}$$



- Kalman filters, Particle filters, Bayesian Networks, Partially Observable Markov Decision Processes (POMDPs), Hidden Markov Models (HMMs) and many more!

Summary 1

- Pattern classification is a necessary task in several application domains.
- Bayesian formulation for classification results in incremental probabilistic updates.
- Regression is a widely-used predictive scheme in several domains.
- Bayesian formulation for regression better models the prediction noise.

Summary 1

- Pattern classification is a necessary task in several application domains.
- Bayesian formulation for classification results in incremental probabilistic updates.
- Regression is a widely-used predictive scheme in several domains.
- Bayesian formulation for regression better models the prediction noise.

Summary 2

- Bayesian inference is a general framework for probabilistic state estimation.
- Markov assumption, though not always true, allows for elegant belief updates.
- Incorporates changes in system dynamics independent of the observations of the system.
- Applications: *computer vision, robotics, adaptive testing, fault localization*.

Summary 2

- Bayesian inference is a general framework for probabilistic state estimation.
- Markov assumption, though not always true, allows for elegant belief updates.
- Incorporates changes in system dynamics independent of the observations of the system.
- Applications: *computer vision, robotics, adaptive testing, fault localization*.

Summary 2

- Bayesian inference is a general framework for probabilistic state estimation.
- Markov assumption, though not always true, allows for elegant belief updates.
- Incorporates changes in system dynamics independent of the observations of the system.
- Applications: *computer vision, robotics, adaptive testing, fault localization.*

Summary 2

- Bayesian inference is a general framework for probabilistic state estimation.
- Markov assumption, though not always true, allows for elegant belief updates.
- Incorporates changes in system dynamics independent of the observations of the system.
- Applications: *computer vision, robotics, adaptive testing, fault localization*.