Naïve Bayes

Lecture 5:

Naïve Bayes Networks

ECE/CS 498 DS

Professor Ravi K. Iyer

Department of Electrical and Computer Engineering

University of Illinois

Announcements

- **Completed ICA 1 worksheet due today by end of lecture!**
- MP 1 Checkpoint 1 due tomorrow 2/6!
  - One submission per group. Include
    - A single .ipynb file for both Task 0 and Task 1
    - A single .pdf file with answers to the questions in Task 0 and Task 1 (template has been released)
- Final Project information will be released soon!
  - For 4 credit-hour students, initial project ideas will be due next week
- Tomorrow is the last day to turn in HW 0 for late credit (after 3 late days, assignment won't be graded)
- Discussion section on Friday, Feb 7 will be an additional office hour
  - Discussion section next Friday, Feb 14 will be extra practice with Bayesian Networks!

Naïve Bayes- Intuitive Example: Infer the Dog-Breed

- Assume two possible dog breeds (*classes*)
  - C_1= German Shepherd (GS)
  - C_2 = Dalmatian (D)
- Suppose we label dogs with a height of at least 40 cm are ***tall***
- **Problem**:
  - We have a dog whose breed we do not know, say d
  - We know that the dog is ***tall***
  - Based on the above information and earlier *training data*, can we infer the breed of d?
- **Equivalent Question**: Based on the training data, is it more probable that a ***tall*** dog is a German Shepherd or a Dalmatian?
  - Which is greater: $p(GS|tall)$ or $p(D|tall)$?

German Shepherd

Dalmatian

Think about Bayes Theorem

Likelihood: What is the probability of being tall given Dalmatian?

Prior: What is the probability of being a Dalmatian?

$$p(D|tall) = \frac{p(tall|D)\ p(D)}{p(tall)}$$

Posterior: What is the probability of being Dalmatian given tall?

Evidence: What is the probability of being tall?

- Fortunately, we have a database with dog heights and breed
- We use the database to calculate the posterior probability for both D and GS

Height	Breed
Short	GS
Tall	D
Short	D
Tall	D
Tall	GS
Short	D
Short	D
Short	GS

Guess the dog breed: calculations

Suppose C_k = Class "k" (either D or GS)

$$p(C_k|tall) = \frac{p(tall|C_k)\,p(C_k)}{p(tall)}$$

$$p(GS|tall) = \frac{p(tall|GS)\,p(GS)}{p(tall)} = \frac{\frac{1}{3} * \frac{3}{8}}{3/8} = \frac{\mathbf{0.125}}{3/8}$$

$$p(D|tall) = \frac{p(tall|D)\,p(D)}{p(tall)} = \frac{\frac{2}{5} * \frac{5}{8}}{3/8} = \frac{\mathbf{0.250}}{3/8}$$

$p(D|tall) > p(GS|tall)$, therefore, dog d is more likely to be **Dalmatian**.

Height	Breed
Short	GS
Tall	D
Short	D
Tall	D
Tall	GS
Short	D
Short	D
Short	GS

Multiple features

- In our example we had just one *feature* (height)
- What if we have several *features* (*e.g.*, height, weight, color of eyes)?
- Say that the dog d is tall, heavy and has brown eyes
- Assuming that **features are independent given the breed,** we can write

$$p(d|C_k) = p(tall, heavy, brown\ eyes|C_k)$$
$$= p(tall|C_k)\ p(heavy|C_k)\ p(brown\ eyes|C_k)$$

- Generally, suppose $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ are a series of n features that are observed. Then,

$$p(\boldsymbol{x}|C_k) = p(x_1|C_k) * p(x_2|C_k) * \cdots * p(x_n|C_k)$$

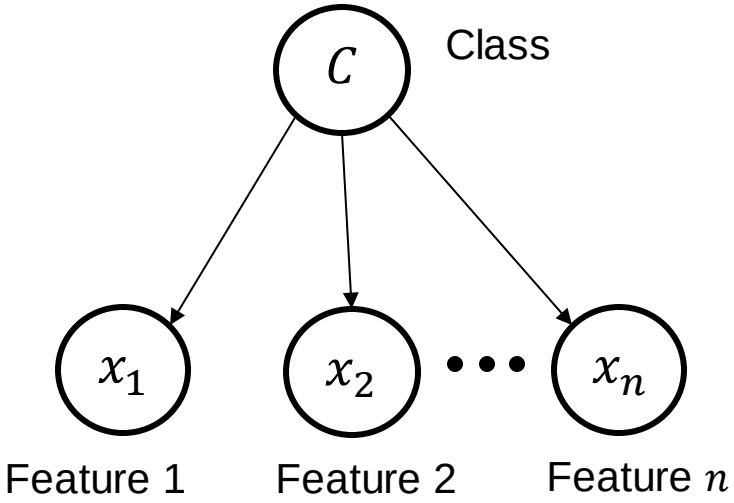
Probability of class C_k generating instance $\boldsymbol{x}$

Probability of class C_k generating observed value of feature 1

Probability of class C_k generating observed value of feature 2

Naïve Bayes Classifier Representation

- The assumption of independence of features given the class results in a **Naïve Bayes classifier**
- Naïve Bayes classifier can be represented as a graph



Note the direction of the arrow…

Naïve Bayes Classifier (Cont'd)

- Driving Question: Given a **previously unseen** data point $\boldsymbol{x}$, which class C_k does the data point belong to?
  - Previously unseen data point: $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$
  - K classes: $C_k, 1 \leq k \leq K$
- Intuitively, the most probable class would be the one with the maximum probability given the observations (data or evidence)
  - Note the difference between "max" and "argmax" in the below expression!

$$C^* = \underset{k \in \{1,..,K\}}{\text{argmax}}\ p(C_k|\boldsymbol{x})$$

- A Naïve Bayes classifier can be derived from three core concepts:
  - Chain rule
  - Conditional Independence
  - Maximum a posteriori (MAP) rule

Bayes Theorem and Chain Rule

From Bayes Theorem, we have

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})}$$

The above involves calculation of $p(\boldsymbol{x}|C_k) = p(x_1, x_2, \dots, x_n|C_k)$

Using ***chain rule***, we get

$$\begin{aligned} p(\boldsymbol{x}|C_k) &= p(x_1, x_2, \dots, x_n|C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2, \dots, x_n|C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k)p(x_3, \dots, x_n|C_k)\dots. \\ &\dots \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k) \dots p(x_{n-1}|x_n, C_k)p(x_n|C_k) \end{aligned}$$

Conditional Independence

- NB assumes that the features are *class conditionally independent* (independent given/conditioning on the class)
  - x_i is conditionally independent of x_j given C_k if $i \neq j, \forall\, i, j \in \{1, \dots, n\}$
- Therefore, $p(x_i|x_j, C_k) = p(x_i|C_k)$ if $i \neq j$
- By extension, $p(x_i|x_{i+1}, x_{i+2}, \dots, x_n, C_k) = p(x_i|C_k), \forall\, i \in \{1, \dots, n-1\}$
- Applying the class conditional independence to $p(\boldsymbol{x}|C_k)$ gives

$$p(\boldsymbol{x}|C_k) = p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k) \dots p(x_{n-1}|x_n, C_k)p(x_n|C_k)$$
$$= p(x_1|C_k)p(x_2|C_k) \dots p(x_n|C_k)$$
$$= \prod_{i=1}^{n} p(x_i|C_k)$$

MAP

- We started with:

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})}$$

- Let $Z = p(\boldsymbol{x})$. We get,

$$p(C_k|\boldsymbol{x}) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

- Apply the MAP rule, i.e., pick the class with maximum posterior probability

$$C^* = \underset{k\in\{1,..,K\}}{\text{argmax}}\ p(C_k|\boldsymbol{x}) = \underset{k\in\{1,..,K\}}{\text{argmax}} \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

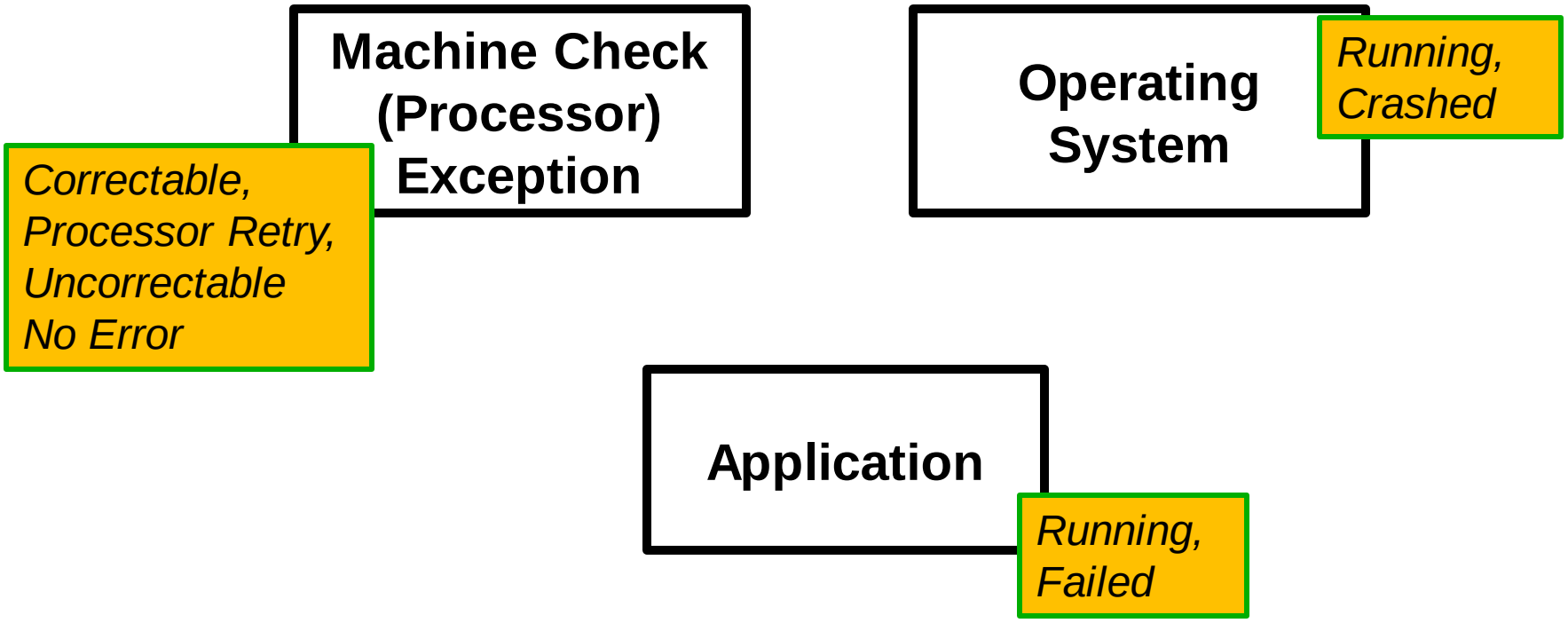
MAP

- We have computed everything except $Z = p(\boldsymbol{x})$
  - $p(x_i|C_k)$ and $p(C_k)$ are computed from the training data
- Z is dependent only on $\boldsymbol{x}$ (no dependence on k). Therefore, it is constant once $\boldsymbol{x}$ is known and can be ignored.
- A Naïve Bayes Classifier chooses a class $\mathrm{C_k}$ by evaluating:

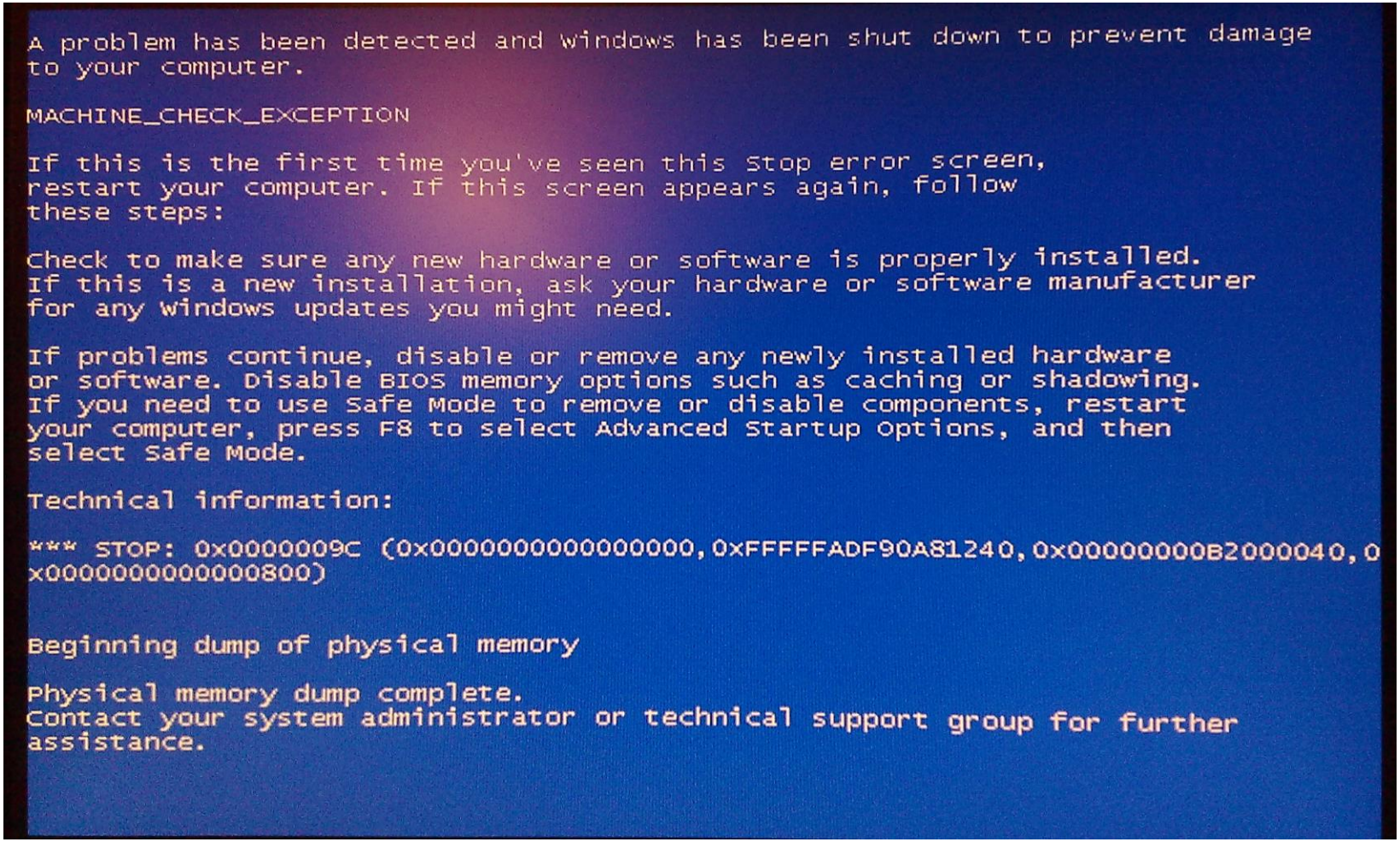
$$C^* = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Naïve Bayes Model Depicting the Resilience of your Laptop Application

- ***Task: Predict application exit status (success, failure) based on the observed failures in the system (such as MCEs, and OS errors)***

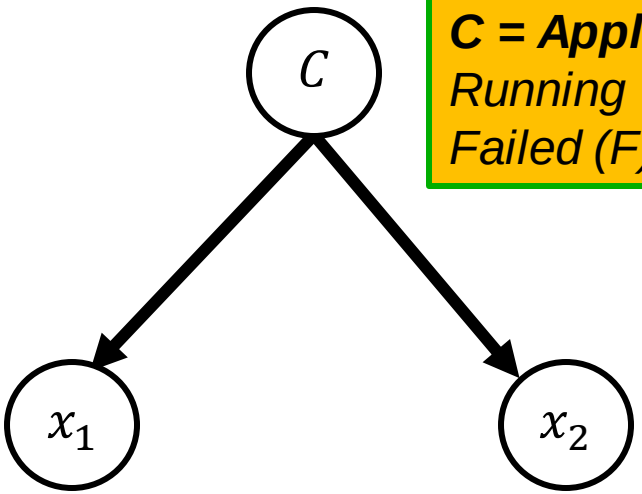


Example of Machine Check Exception (Processor Error)



Naïve Bayes

- Predicting application failures given MCE and OS-information



C = Application Status
Running (R),
Failed (F)

x_1: Machine Check Exception
Correctable (C), Processor Retry (P),
Uncorrectable (U), No Error (NE)

x_2: Operating System Status
Running (R),
Failed (F)

Toy ML for Personal Computers: Data

MacOS: sudo cat /var/log/system.log
Linux: sudo cat /var/log/syslog

1363323625 local3 6 2013-03-15T00:00:25.07 c6-3c1s1n0 xtconsole 11144 p0-20130219t183043 CPU 16: **Machine Check Exception:** 0 Bank 4: 9c624400001c017b

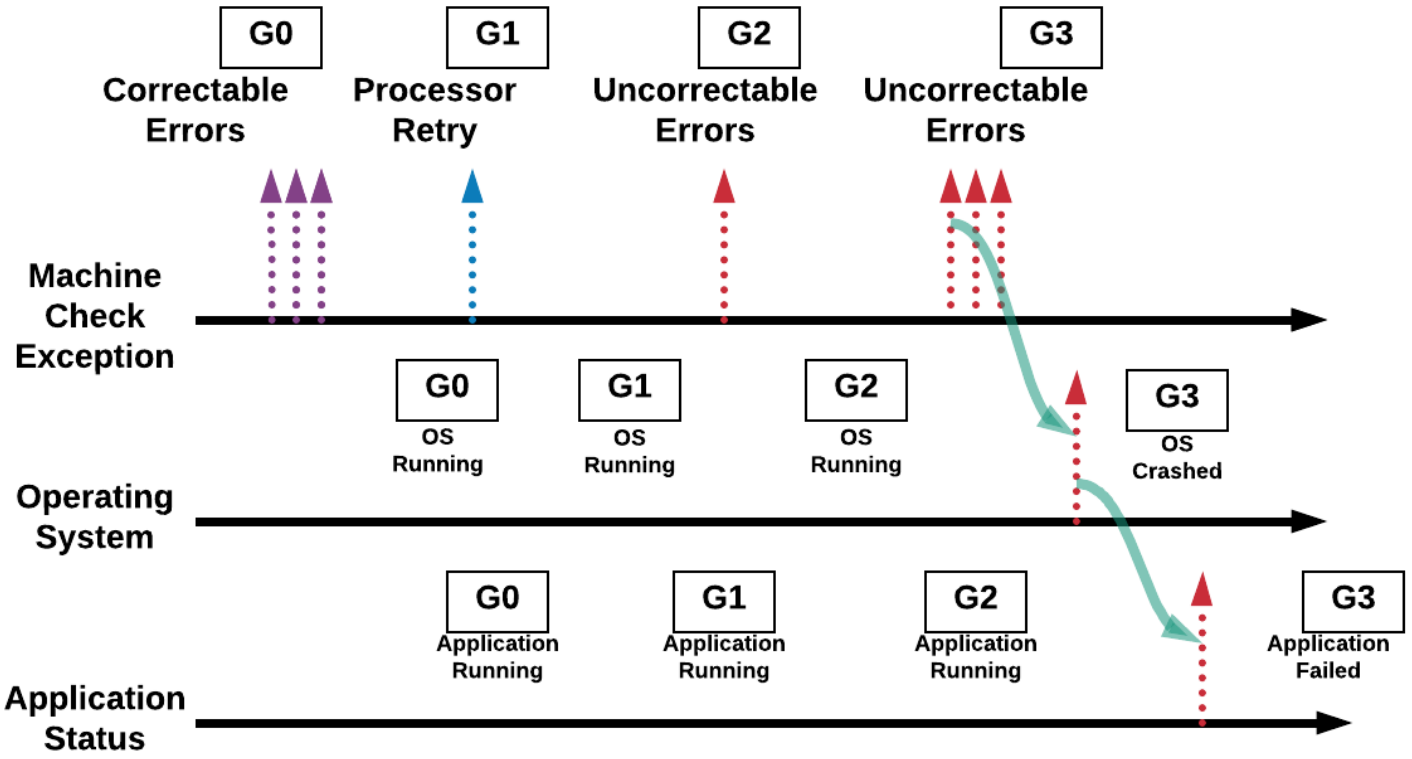
1363323625 local3 6 2013-03-15T00:00:25.07 c6-3c1s1n0 xtconsole 11144 p0-20130219t183043 TSC 0 ADDR bcb75daa0 MISC c00a000001000000

1363323625 local3 6 2013-03-15T00:00:25.07 c6-3c1s1n0 xtconsole 11144 p0-20130219t183043 **PROCESSOR** 2:600f12 TIME 1363323624 SOCKET 1 APIC 20

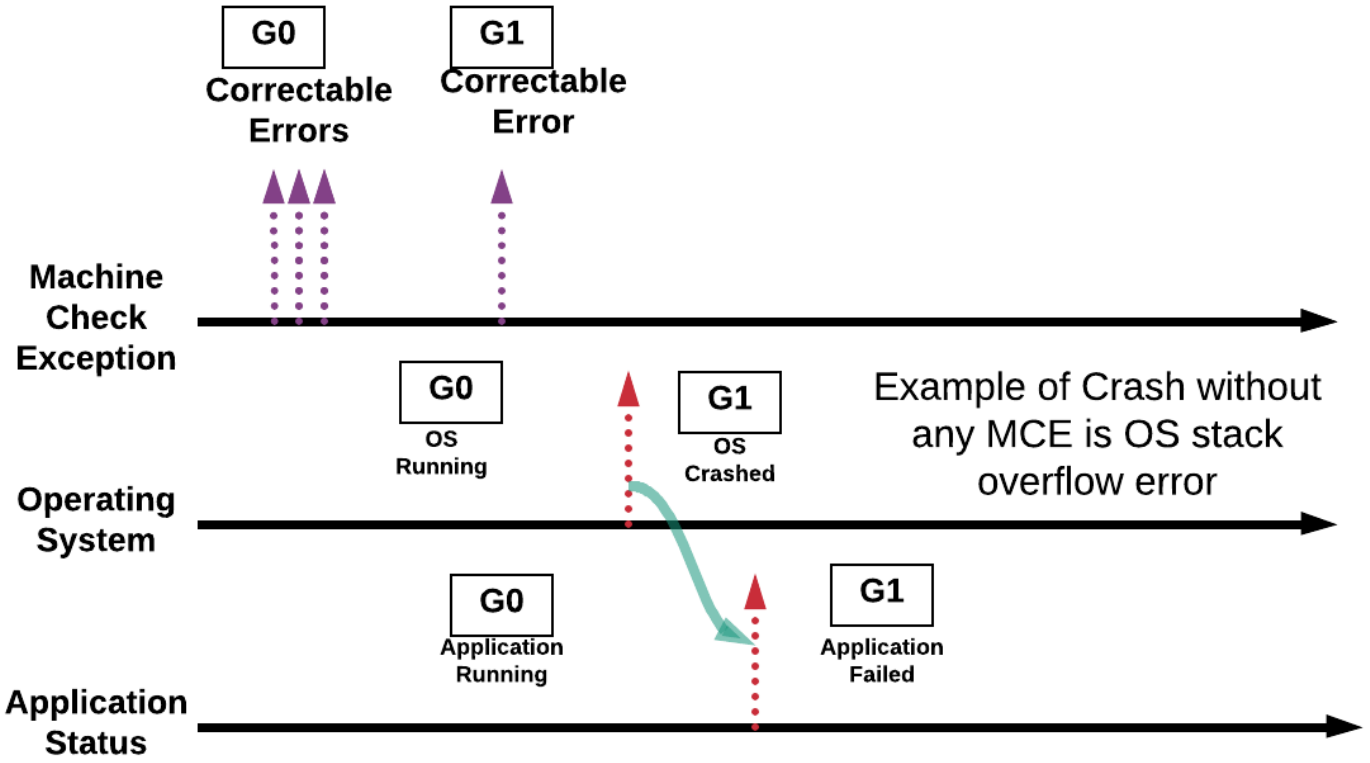
1452290276 daemon 5 2016-01-08T15:57:56-06:00 172.16.255.118 puppet-agent 25348 - (/Stage[main]/Sysctl/Exec[subscribe-sysctl]/returns) **kernel.panic = 0**

1452290276 local3 6 2015-09-25T00:00:37.369592-05:00 c8-6c2s7n3 **APP=NAMD, exit_code = 1**

Coalescing Like Events



Coalescing Like Events



Final Dataset

Cluster	MCE	OS	App
G0	C	R	**R**
G1	P	R	**R**
G2	U	R	**R**
G3	U	F	**R**
…	..	..	..
G999	C	F	**F**

Naïve Bayes Training

- Recall Naive Bayes Classifier

$$C^* = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

- In our example, we need to train and get following parameters
  - Prior probabilities : $P(App = Running), P(App = Fail)$
  - Conditional Probabilities for each feature

$P(OS = Running \mid App = F)$	$P(OS = Running \mid App = R)$
$P(OS = Failed \mid App = F)$	$P(OS = Failed \mid App = R)$

$P(MCE = Correctable \mid App = F)$	$P(MCE = Correctable \mid App = R)$
$P(MCE = Processor\ Retry \mid App = F)$	$P(MCE = Processor\ Retry \mid App = R)$
$P(MCE = Uncorrectable \mid App = F)$	$P(MCE = Uncorrectable \mid App = R)$
$P(MCE = No\ error \mid App = F)$	$P(MCE = No\ error \mid App = R)$

Training the Model

$$C = \underset{k \in \{1,..,K\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$$

Cluster	MCE	OS	App
G0	C	R	R
G1	P	R	R
G2	U	R	R
G3	U	F	R
…	..	..	..
G999	C	F	F

Counting

Prior Probabilities

App	#	P(App)
R	950	0.95
F	50	0.05

CPTs

MCE	App	#	P(MCE|App)
C	R	66	0.07
P	R	5	0.005
U	R	5	0.005
NE	R	874	0.92
C	F	2	0.04
P	F	5	0.1
U	F	40	0.8
NE	F	3	0.06

OS	App	#	P(OS|App)
R	R	940	0.99
F	R	10	0.01
R	F	45	0.9
F	F	5	0.1

Naïve Bayes Inference – Example 1

Inference task: Is application running or failed ?

C1: $P(MCE|App = F)\ P(OS|App = F)\ P(App = F)$
C2: $P(MCE|App = R)P(OS|App = R)\ P(App = R)$

- Given instance:
  - MCE = P, OS = R
- Posterior probabilities
  - $P(C_1|\boldsymbol{x}) \propto\ P(MCE = P|App = \boldsymbol{F})\ P(OS = \mathrm{R}|App = \boldsymbol{F})\ P(App = \boldsymbol{F})$
  - $P(C_2|\boldsymbol{x}) \propto P(MCE = P|App = \boldsymbol{R})\ P(OS = \mathrm{R}|App = \boldsymbol{R})\ P(App = \boldsymbol{R})$
- Evaluation via substitution
  - $P(C_1|\boldsymbol{x}) \propto 0.1 * 0.9 * 0.05 = 0.0045$
  - $P(C_2|\boldsymbol{x}) \propto 0.005 * 0.99 * 0.95 =$ **0.0047025**
- Apple MAP Rule: Select hypothesis corresponding to max P
  - **C2 selected, i.e., APP = R (running)**

Naïve Bayes Inference – Example 2

Inference task: Is application running or failed ?

C1: $P(MCE|App = F)\ P(OS|App = F)\ P(App = F)$
C2: $P(MCE|App = R)P(OS|App = R)\ P(App = R)$

- Given instance:
  - MCE = U, OS = F
- Posterior probabilities
  - $P(C_1|\boldsymbol{x}) \propto P(MCE = U|App = \boldsymbol{F})\ P(OS = F|App = \boldsymbol{F})\ P(App = \boldsymbol{F})$
  - $P(C_2|\boldsymbol{x}) \propto P(MCE = U|App = \boldsymbol{R}) \times P(OS = \mathrm{F}|App = \boldsymbol{R})\ P(App = \boldsymbol{R})$
- Evaluation via substitution
  - $P(C_1|\boldsymbol{x}) \propto 0.8 * 0.1 * 0.05 =$ **4.0e-03**
  - $P(C_2|\boldsymbol{x}) \propto$ 0.005 * 0.01 * 0.95 = 4.75e-05
- Apple MAP Rule: Select hypothesis corresponding to max P
  - **C1 selected, i.e., APP = F (failed)**

What happens if we don't have discrete features?

Back to the Dogs

- We again have two classes:
  - C_1 = German Shepherd (GS)
  - C_2 = Dalmatian (D)
- Instead of categorizing height as "*tall*" or "*short*", we use its exact value
- We have a dog whose breed we do not know, say $\tilde{d}$
- We know that $\tilde{d}$'s height is 38 cm
- Based on the above information and some training data, can we find the breed of $\tilde{d}$?

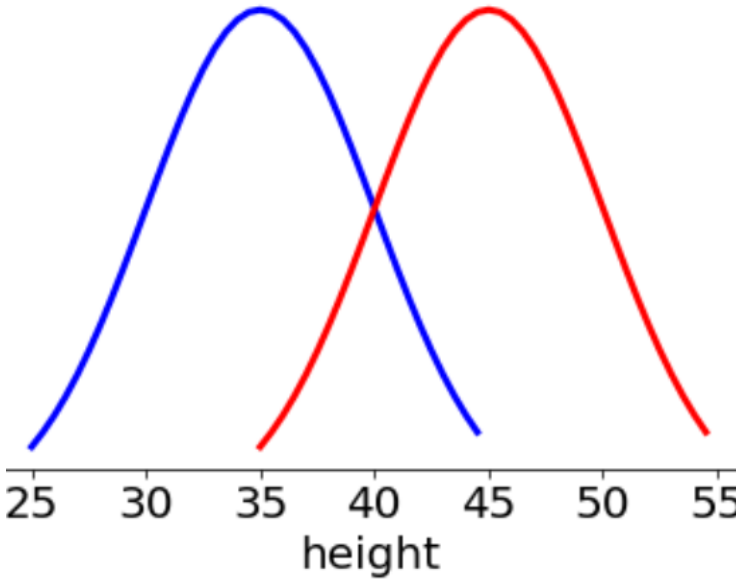
German Shepherd

Dalmatian

Naïve Bayes with continuous variables

- Height can be any real number- therefore its **distribution** will be **continuous**
- With enough data, we can approximate the distribution of heights for the two classes
  - We can also assume a parametric form for the distribution…

- Here is the distribution of height based on some training data (visualized on the right)
  - German Shepherd: $\mathcal{N}(35, 25)$
  - Dalmatian: $\mathcal{N}(45, 25)$
- Priors are equal



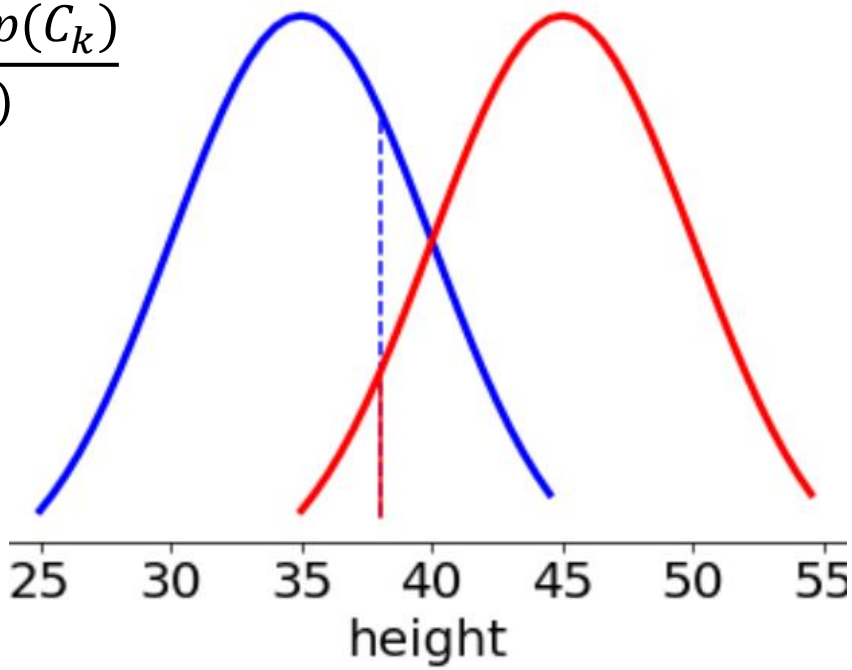
Guess the dog breed: Calculations

$$p(C_k|38) = \frac{p(38|C_k)\,p(C_k)}{p(38)}$$

$$p(GS|38) = \frac{p(38|GS)\,p(GS)}{p(38)}$$

$$\propto \frac{1}{\sqrt{50\pi}} \exp\left(-\frac{(38-35)^2}{50}\right) * 0.5$$

$$= 0.067 * 0.5 = \mathbf{0.034}$$



$$p(D|38) = \frac{p(38|D)\,p(D)}{p(38)}$$

$$\propto \frac{1}{\sqrt{50\pi}} \exp\left(-\frac{(38-45)^2}{50}\right) * 0.5$$

$$= 0.03 * 0.5 = \mathbf{0.015}$$

$p(D|38) < p(GS|38)$, therefore, dog $\tilde{d}$ is more likely to be a German Shepherd.