

# Fundamentals of Machine Learning for Predictive Data Analytics

## Chapter 6: Probability-based Learning Sections 6.4, 6.5

John Kelleher and Brian Mac Namee and Aoife D'Arcy

[john.d.kelleher@dit.ie](mailto:john.d.kelleher@dit.ie)   [brian.macnamee@ucd.ie](mailto:brian.macnamee@ucd.ie)   [aoife@theanalyticsstore.com](mailto:aoife@theanalyticsstore.com)

- 1 Smoothing
- 2 Continuous Features: Probability Density Functions
- 3 Continuous Features: Binning
- 4 Bayesian Networks
- 5 Summary

# Smoothing

|                                     |  |
|-------------------------------------|--|
| $P(fr) = 0.3$                       | $P(\neg fr) = 0.7$                         |
| $P(CH = 'none'   fr) = 0.1666$      | $P(CH = 'none'   \neg fr) = 0$             |
| $P(CH = 'paid'   fr) = 0.1666$      | $P(CH = 'paid'   \neg fr) = 0.2857$        |
| $P(CH = 'current'   fr) = 0.5$      | $P(CH = 'current'   \neg fr) = 0.2857$     |
| $P(CH = 'arrears'   fr) = 0.1666$   | $P(CH = 'arrears'   \neg fr) = 0.4286$     |
| $P(GC = 'none'   fr) = 0.8334$      | $P(GC = 'none'   \neg fr) = 0.8571$        |
| $P(GC = 'guarantor'   fr) = 0.1666$ | $P(GC = 'guarantor'   \neg fr) = 0$        |
| $P(GC = 'coapplicant'   fr) = 0$    | $P(GC = 'coapplicant'   \neg fr) = 0.1429$ |
| $P(ACC = 'own'   fr) = 0.6666$      | $P(ACC = 'own'   \neg fr) = 0.7857$        |
| $P(ACC = 'rent'   fr) = 0.3333$     | $P(ACC = 'rent'   \neg fr) = 0.1429$       |
| $P(ACC = 'free'   fr) = 0$          | $P(ACC = 'free'   \neg fr) = 0.0714$       |

| CREDIT HISTORY | GUARANTOR/COAPPLICANT | ACCOMMODATION | FRAUDULENT |
|----------------|-----------------------|---------------|------------|
| paid           | guarantor             | free          | ?          |

$$P(fr) = 0.3$$

$$P(\neg fr) = 0.7$$

$$P(CH = \textit{paid} \mid fr) = 0.1666$$

$$P(CH = \textit{paid} \mid \neg fr) = 0.2857$$

$$P(GC = \textit{guarantor} \mid fr) = 0.1666$$

$$P(GC = \textit{guarantor} \mid \neg fr) = 0$$

$$P(ACC = \textit{free} \mid fr) = 0$$

$$P(ACC = \textit{free} \mid \neg fr) = 0.0714$$

$$(\prod_{k=1}^m P(\mathbf{q}[k] \mid fr)) \times P(fr) = 0.0$$

$$(\prod_{k=1}^m P(\mathbf{q}[k] \mid \neg fr)) \times P(\neg fr) = 0.0$$

| CREDIT HISTORY | GUARANTOR/COAPPLICANT | ACCOMMODATION | FRAUDULENT |
|----------------|-----------------------|---------------|------------|
| paid           | guarantor             | free          | ?          |

- The standard way to avoid this issue is to use **smoothing**.
- Smoothing takes some of the probability from the events with lots of the probability share and gives it to the other probabilities in the set.

- There are several different ways to smooth probabilities, we will use **Laplace smoothing**.

### Laplace Smoothing (conditional probabilities)

$$P(f = v|t) = \frac{\text{count}(f = v|t) + k}{\text{count}(f|t) + (k \times |\text{Domain}(f)|)}$$

|               |   |   |        |
|---------------|---|---|--------|
| Raw           | $P(GC = none \neg fr)$                                      | = | 0.8571 |
| Probabilities | $P(GC = guarantor \neg fr)$                                 | = | 0      |
|               | $P(GC = coapplicant \neg fr)$                               | = | 0.1429 |
| Smoothing     | $k$   | = | 3      |
| Parameters    | $count(GC \neg fr)$   | = | 14     |
|               | $count(GC = none \neg fr)$                                  | = | 12     |
|               | $count(GC = guarantor \neg fr)$                             | = | 0      |
|               | $count(GC = coapplicant \neg fr)$                           | = | 2      |
|               | $ Domain(GC) $  | = | 3      |
| Smoothed      | $P(GC = none \neg fr) = \frac{12+3}{14+(3 \times 3)}$       | = | 0.6522 |
| Probabilities | $P(GC = guarantor \neg fr) = \frac{0+3}{14+(3 \times 3)}$   | = | 0.1304 |
|               | $P(GC = coapplicant \neg fr) = \frac{2+3}{14+(3 \times 3)}$ | = | 0.2174 |

**Table:** Smoothing the posterior probabilities for the GUARANTOR/COAPPLICANT feature conditioned on FRAUDULENT being False.



|                                 |  |
|---------------------------------|--|
| $P(fr) = 0.3$                   | $P(\neg fr) = 0.7$                     |
| $P(CH = none fr) = 0.2222$      | $P(CH = none \neg fr) = 0.1154$        |
| $P(CH = paid fr) = 0.2222$      | $P(CH = paid \neg fr) = 0.2692$        |
| $P(CH = current fr) = 0.3333$   | $P(CH = current \neg fr) = 0.2692$     |
| $P(CH = arrears fr) = 0.2222$   | $P(CH = arrears \neg fr) = 0.3462$     |
| $P(GC = none fr) = 0.5333$      | $P(GC = none \neg fr) = 0.6522$        |
| $P(GC = guarantor fr) = 0.2667$ | $P(GC = guarantor \neg fr) = 0.1304$   |
| $P(GC = coapplicant fr) = 0.2$  | $P(GC = coapplicant \neg fr) = 0.2174$ |
| $P(ACC = own fr) = 0.4667$      | $P(ACC = own \neg fr) = 0.6087$        |
| $P(ACC = rent fr) = 0.3333$     | $P(ACC = rent \neg fr) = 0.2174$       |
| $P(ACC = Free fr) = 0.2$        | $P(ACC = Free \neg fr) = 0.1739$       |

**Table:** The Laplace smoothed, with  $k = 3$ , probabilities needed by a Naive Bayes prediction model calculated from the fraud detection dataset. Notation key: FR=FRAUDULENT, CH=CREDIT HISTORY, GC = GUARANTOR/COAPPLICANT, ACC = ACCOMODATION, T='True', F='False'.

| CREDIT HISTORY | GUARANTOR/COAPPLICANT | ACCOMMODATION | FRAUDULENT |
|----------------|-----------------------|---------------|------------|
| paid           | guarantor             | free          | ?          |

---

|                                 |                                      |
|---------------------------------|--------------------------------------|
| $P(fr) = 0.3$                   | $P(\neg fr) = 0.7$                   |
| $P(CH = paid fr) = 0.2222$      | $P(CH = paid \neg fr) = 0.2692$      |
| $P(GC = guarantor fr) = 0.2667$ | $P(GC = guarantor \neg fr) = 0.1304$ |
| $P(ACC = Free fr) = 0.2$        | $P(ACC = Free \neg fr) = 0.1739$     |

---


$$(\prod_{k=1}^m P(\mathbf{q}[m]|fr)) \times P(fr) = 0.0036$$

$$(\prod_{k=1}^m P(\mathbf{q}[m]|\neg fr)) \times P(\neg fr) = 0.0043$$

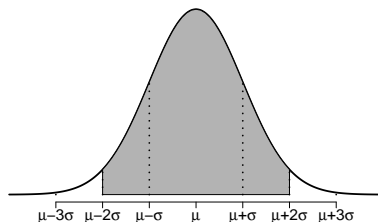

---

**Table:** The relevant smoothed probabilities, from Table 2 <sup>[9]</sup>, needed by the Naive Bayes prediction model in order to classify the query from the previous slide and the calculation of the scores for each candidate classification.

# Continuous Features: Probability Density Functions

- A **probability density function** (PDF) represents the probability distribution of a continuous feature using a mathematical function, such as the normal distribution.

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



- A PDF defines a density curve and the shape of the of the curve is determined by:
  - the statistical distribution that is used to define the PDF
  - the values of the statistical distribution parameters

## Table: Definitions of some standard probability distributions.

Normal

$x \in \mathbb{R}$

$\mu \in \mathbb{R}$

$\sigma \in \mathbb{R}_{>0}$

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

Student-*t*

$x \in \mathbb{R}$

$\phi \in \mathbb{R}$

$\rho \in \mathbb{R}_{>0}$

$\kappa \in \mathbb{R}_{>0}$

$z = \frac{x - \phi}{\rho}$

$$\tau(x, \phi, \rho, \kappa) = \frac{\Gamma(\frac{\kappa+1}{2})}{\Gamma(\frac{\kappa}{2}) \times \sqrt{\pi\kappa} \times \rho} \times \left(1 + \left(\frac{1}{\kappa} \times z^2\right)\right)^{-\frac{\kappa+1}{2}}$$

Exponential

$x \in \mathbb{R}$

$\lambda \in \mathbb{R}_{>0}$

$$E(x, \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Mixture of  $n$  Gaussians

$x \in \mathbb{R}$

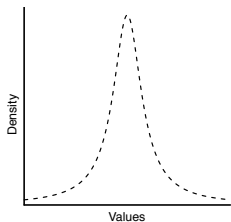
$\{\mu_1, \dots, \mu_n | \mu_i \in \mathbb{R}\}$

$\{\sigma_1, \dots, \sigma_n | \sigma_i \in \mathbb{R}_{>0}\}$

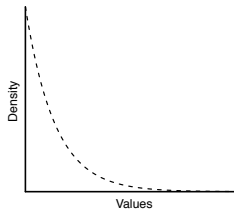
$\{\omega_1, \dots, \omega_n | \omega_i \in \mathbb{R}_{>0}\}$

$\sum_{i=1}^n \omega_i = 1$

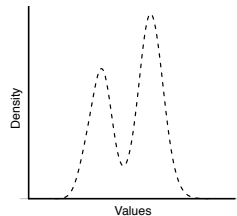
$$N(x, \mu_1, \sigma_1, \omega_1, \dots, \mu_n, \sigma_n, \omega_n) = \sum_{i=1}^n \frac{\omega_i}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x - \mu_i)^2}{2\sigma_i^2}}$$



(a) Normal/Student-t



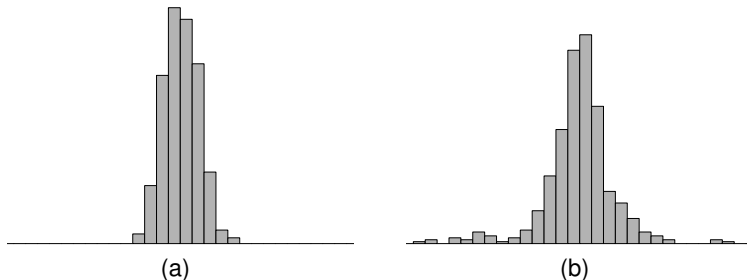
(b) Exponential



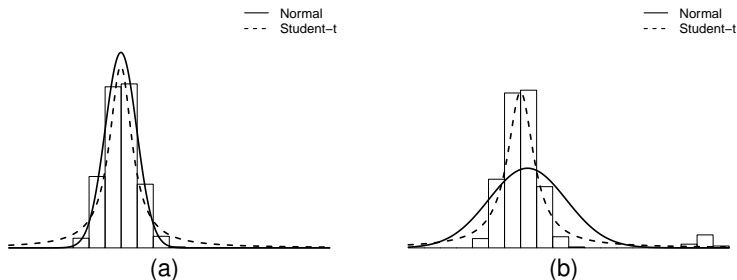
(c) Mixture of Gaussians

**Figure:** Plots of some well known probability distributions.

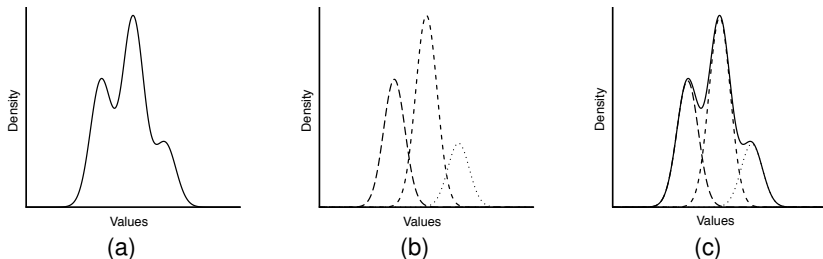




**Figure:** Histograms of two unimodal datasets: (a) the distribution has light tails; (b) the distribution has fat tails.

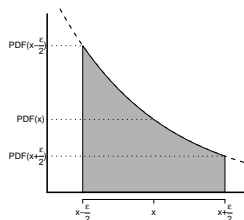


**Figure:** Illustration of the robustness of the student- $t$  distribution to outliers: (a) a density histogram of a unimodal dataset overlaid with the density curves of a normal and a student- $t$  distribution that have been fitted to the data; (b) a density histogram of the same dataset with outliers added, overlaid with the density curves of a normal and a student- $t$  distribution that have been fitted to the data. The student- $t$  distribution is less affected by the introduction of outliers. (This figure is inspired by Figure 2.16 in (Bishop, 2006).)

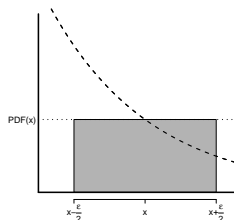


**Figure:** Illustration of how a mixture of Gaussians model is composed of a number of normal distributions. The curve plotted using a solid line is the mixture of Gaussians density curve, created using an appropriately weighted summation of the three normal curves, plotted using dashed and dotted lines.

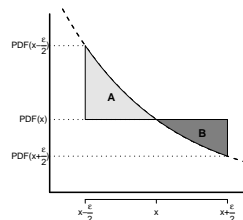
- A PDF is an abstraction over a density histogram and consequently PDF represents probabilities in terms of area under the curve.
- To use a PDF to calculate a probability we need to think in terms of the area under an interval of the PDF curve.
- We can calculate the area under a PDF by looking this up in a probability table or to use integration to calculate the area under the curve within the bounds of the interval.



(a)



(b)



(c)

**Figure:** (a) The area under a density curve between the limits  $x - \frac{\epsilon}{2}$  and  $x + \frac{\epsilon}{2}$ ; (b) the approximation of this area computed by  $PDF(x) \times \epsilon$ ; and (c) the error in the approximation is equal to the difference between area A, the area under the curve omitted from the approximation, and area B, the area above the curve erroneously included in the approximation. Both of these areas will get smaller as the width of the interval gets smaller, resulting in a smaller error in the approximation.

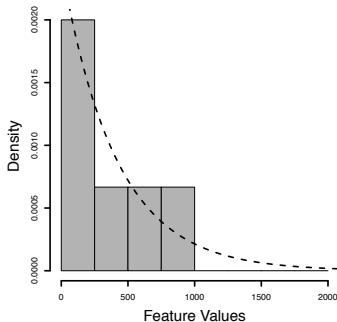
- There is no hard and fast rule for deciding on **interval size**  
- instead, this decision is done on a case by case basis  
and is dependent on the precision required in answering a question.
- To illustrate how PDFs can be used in Naive Bayes models  
we will extend our loan application fraud detection query to  
have an ACCOUNT BALANCE feature

**Table:** The dataset from the loan application fraud detection domain with a new continuous descriptive features added: ACCOUNT BALANCE

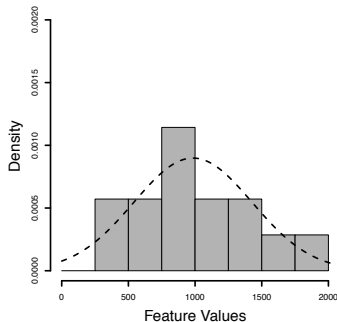
| ID | CREDIT HISTORY | GUARANTOR/<br>CoAPPLICANT | ACCOMMODATION | ACCOUNT<br>BALANCE | FRAUD |
|----|----------------|---------------------------|---------------|--------------------|-------|
| 1  | current        | none                      | own           | 56.75              | true  |
| 2  | current        | none                      | own           | 1,800.11           | false |
| 3  | current        | none                      | own           | 1,341.03           | false |
| 4  | paid           | guarantor                 | rent          | 749.50             | true  |
| 5  | arrear         | none                      | own           | 1,150.00           | false |
| 6  | arrear         | none                      | own           | 928.30             | true  |
| 7  | current        | none                      | own           | 250.90             | false |
| 8  | arrear         | none                      | own           | 806.15             | false |
| 9  | current        | none                      | rent          | 1,209.02           | false |
| 10 | none           | none                      | own           | 405.72             | true  |
| 11 | current        | coapplicant               | own           | 550.00             | false |
| 12 | current        | none                      | free          | 223.89             | true  |
| 13 | current        | none                      | rent          | 103.23             | true  |
| 14 | paid           | none                      | own           | 758.22             | false |
| 15 | arrear         | none                      | own           | 430.79             | false |
| 16 | current        | none                      | own           | 675.11             | false |
| 17 | arrear         | coapplicant               | rent          | 1,657.20           | false |
| 18 | arrear         | none                      | free          | 1,405.18           | false |
| 19 | arrear         | none                      | own           | 760.51             | false |
| 20 | current        | none                      | own           | 985.41             | false |

- We need to define two PDFs for the new ACCOUNT BALANCE (AB) feature with each PDF conditioned on a different value in the domain or the target:
  - $P(AB = X|fr) = PDF_1(AB = X|fr)$
  - $P(AB = X|\neg fr) = PDF_2(AB = X|\neg fr)$
- Note that these two PDFs do not have to be defined using the same statistical distribution.





(a)



(b)

**Figure:** Histograms, using a bin size of 250 units, and density curves for the ACCOUNT BALANCE feature: (a) the fraudulent instances overlaid with a fitted exponential distribution; (b) the non-fraudulent instances overlaid with a fitted normal distribution.

- From the shape of these histograms it appears that
  - the distribution of values taken by the ACCOUNT BALANCE feature in the set of instances where the target feature FRAUDULENT=*'True'* follows an exponential distribution
  - the distributions of values taken by the ACCOUNT BALANCE feature in the set of instances where the target feature FRAUDULENT=*'False'* is similar to a normal distribution.
- Once we have selected the distributions the next step is to fit the distributions to the data.

- To fit the exponential distribution we simply compute the sample mean,  $\bar{x}$ , of the ACCOUNT BALANCE feature in the set of instances where FRAUDULENT='True' and set the  $\lambda$  parameter equal to one divided by  $\bar{x}$ .
- To fit the normal distribution to the set of instances where FRAUDULENT='False' we simply compute the sample mean and sample standard deviation,  $s$ , for the ACCOUNT BALANCE feature for this set of instances and set the parameters of the normal distribution to these values.

**Table:** Partitioning the dataset based on the value of the target feature and fitting the parameters of a statistical distribution to model the ACCOUNT BALANCE feature in each partition.

| ID                            | ... | ACCOUNT |       |
|-------------------------------|-----|---------|-------|
|                               |     | BALANCE | FRAUD |
| 1                             |     | 56.75   | true  |
| 4                             |     | 749.50  | true  |
| 6                             |     | 928.30  | true  |
| 10                            | ... | 405.72  | true  |
| 12                            |     | 223.89  | true  |
| 13                            |     | 103.23  | true  |
| AB                            |     | 411.22  |       |
| $\lambda = 1 / \overline{AB}$ |     | 0.0024  |       |

| ID     | ... | ACCOUNT  |       |
|--------|-----|----------|-------|
|        |     | BALANCE  | FRAUD |
| 2      |     | 1 800.11 | false |
| 3      |     | 1 341.03 | false |
| 5      |     | 1 150.00 | false |
| 7      |     | 250.90   | false |
| 8      |     | 806.15   | false |
| 9      |     | 1 209.02 | false |
| 11     |     | 550.00   | false |
| 14     |     | 758.22   | false |
| 15     |     | 430.79   | false |
| 16     |     | 675.11   | false |
| 17     |     | 1 657.20 | false |
| 18     |     | 1 405.18 | false |
| 19     |     | 760.51   | false |
| 20     |     | 985.41   | false |
| AB     |     | 984.26   |       |
| sd(AB) |     | 460.94   |       |

**Table:** The Laplace smoothed (with  $k = 3$ ) probabilities needed by a naive Bayes prediction model calculated from the dataset in Table 5<sup>[23]</sup>, extended to include the conditional probabilities for the new ACCOUNT BALANCE feature, which are defined in terms of PDFs.

|  |  |
|--|--|
| $P(fr) = 0.3$  | $P(\neg fr) = 0.7$   |
| $P(CH = none fr) = 0.2222$   | $P(CH = none \neg fr) = 0.1154$  |
| $P(CH = paid fr) = 0.2222$   | $P(CH = paid \neg fr) = 0.2692$  |
| $P(CH = current fr) = 0.3333$  | $P(CH = current \neg fr) = 0.2692$   |
| $P(CH = arrears fr) = 0.2222$  | $P(CH = arrears \neg fr) = 0.3462$   |
| $P(GC = none fr) = 0.5333$   | $P(GC = none \neg fr) = 0.6522$  |
| $P(GC = guarantor fr) = 0.2667$  | $P(GC = guarantor \neg fr) = 0.1304$   |
| $P(GC = coapplicant fr) = 0.2$   | $P(GC = coapplicant \neg fr) = 0.2174$   |
| $P(ACC = own fr) = 0.4667$   | $P(ACC = own \neg fr) = 0.6087$  |
| $P(ACC = rent fr) = 0.3333$  | $P(ACC = rent \neg fr) = 0.2174$   |
| $P(ACC = free fr) = 0.2$   | $P(ACC = free \neg fr) = 0.1739$   |
| $P(AB = x fr)$   | $P(AB = x \neg fr)$  |
| $\approx E\left(\begin{matrix} x, \\ \lambda = 0.0024 \end{matrix}\right)$ | $\approx N\left(\begin{matrix} x, \\ \mu = 984.26, \\ \sigma = 460.94 \end{matrix}\right)$ |

**Table:** A query loan application from the fraud detection domain.

| <b>Credit History</b> | <b>Guarantor/<br/>CoApplicant</b> | <b>Accomodation</b> | <b>Account<br/>Balance</b> | <b>Fraudulent</b> |
|-----------------------|-----------------------------------|---------------------|----------------------------|-------------------|
| paid                  | guarantor                         | free                | 759.07                     | ?                 |

**Table:** The probabilities, from Table 7 <sup>[29]</sup>, needed by the naive Bayes prediction model to make a prediction for the query  $\langle CH = \text{'paid'}, GC = \text{'guarantor'}, ACC = \text{'free'}, AB = 759.07 \rangle$  and the calculation of the scores for each candidate prediction.

|   |   |         |   |   |         |
|---|---|---------|---|---|---------|
| $P(fr)$   | = | 0.3     | $P(\neg fr)$  | = | 0.7     |
| $P(CH = \text{paid} fr)$  | = | 0.2222  | $P(CH = \text{paid} \neg fr)$   | = | 0.2692  |
| $P(GC = \text{guarantor} fr)$   | = | 0.2667  | $P(GC = \text{guarantor} \neg fr)$  | = | 0.1304  |
| $P(ACC = \text{free} fr)$   | = | 0.2     | $P(ACC = \text{free} \neg fr)$  | = | 0.1739  |
| $P(AB = 759.07 fr)$   |   |         | $P(AB = 759.07 \neg fr)$  |   |         |
| $\approx E \left( \begin{array}{c} 759.07, \\ \lambda = 0.0024 \end{array} \right)$ | = | 0.00039 | $\approx N \left( \begin{array}{c} 759.07, \\ \mu = 984.26, \\ \sigma = 460.94 \end{array} \right)$ | = | 0.00077 |
| $(\prod_{k=1}^m P(\mathbf{q}[k] fr)) \times P(fr) = 0.0000014$                      |   |         |   |   |         |
| $(\prod_{k=1}^m P(\mathbf{q}[k] \neg fr)) \times P(\neg fr) = 0.0000033$            |   |         |   |   |         |

# Continuous Features: Binning



- In Section 3.6.2 we explained two of the best known binning techniques **equal-width** and **equal-frequency**.
- We can use these techniques to *bin* continuous features into categorical features
- In general we recommend **equal-frequency binning**.

**Table:** The dataset from a loan application fraud detection domain with a second continuous descriptive feature added: LOAN AMOUNT

| ID | CREDIT HISTORY | GUARANTOR/ COAPPLICANT | ACCOMMODATION | ACCOUNT BALANCE | LOAN AMOUNT | FRAUD |
|----|----------------|------------------------|---------------|-----------------|-------------|-------|
| 1  | current        | none                   | own           | 56.75           | 900         | true  |
| 2  | current        | none                   | own           | 1 800.11        | 150 000     | false |
| 3  | current        | none                   | own           | 1 341.03        | 48 000      | false |
| 4  | paid           | guarantor              | rent          | 749.50          | 10 000      | true  |
| 5  | arrears        | none                   | own           | 1 150.00        | 32 000      | false |
| 6  | arrears        | none                   | own           | 928.30          | 250 000     | true  |
| 7  | current        | none                   | own           | 250.90          | 25 000      | false |
| 8  | arrears        | none                   | own           | 806.15          | 18 500      | false |
| 9  | current        | none                   | rent          | 1 209.02        | 20 000      | false |
| 10 | none           | none                   | own           | 405.72          | 9 500       | true  |
| 11 | current        | coapplicant            | own           | 550.00          | 16 750      | false |
| 12 | current        | none                   | free          | 223.89          | 9 850       | true  |
| 13 | current        | none                   | rent          | 103.23          | 95 500      | true  |
| 14 | paid           | none                   | own           | 758.22          | 65 000      | false |
| 15 | arrears        | none                   | own           | 430.79          | 500         | false |
| 16 | current        | none                   | own           | 675.11          | 16 000      | false |
| 17 | arrears        | coapplicant            | rent          | 1 657.20        | 15 450      | false |
| 18 | arrears        | none                   | free          | 1 405.18        | 50 000      | false |
| 19 | arrears        | none                   | own           | 760.51          | 500         | false |
| 20 | current        | none                   | own           | 985.41          | 35 000      | false |

**Table:** The LOAN AMOUNT continuous feature discretized into 4 equal-frequency bins.

| ID | LOAN<br>AMOUNT | BINNED<br>LOAN<br>AMOUNT | FRAUD |
|----|----------------|--------------------------|-------|
| 15 | 500            | bin1                     | false |
| 19 | 500            | bin1                     | false |
| 1  | 900            | bin1                     | true  |
| 10 | 9,500          | bin1                     | true  |
| 12 | 9,850          | bin1                     | true  |
| 4  | 10,000         | bin2                     | true  |
| 17 | 15,450         | bin2                     | false |
| 16 | 16,000         | bin2                     | false |
| 11 | 16,750         | bin2                     | false |
| 8  | 18,500         | bin2                     | false |

| ID | LOAN<br>AMOUNT | BINNED<br>LOAN<br>AMOUNT | FRAUD |
|----|----------------|--------------------------|-------|
| 9  | 20,000         | bin3                     | false |
| 7  | 25,000         | bin3                     | false |
| 5  | 32,000         | bin3                     | false |
| 20 | 35,000         | bin3                     | false |
| 3  | 48,000         | bin3                     | false |
| 18 | 50,000         | bin4                     | false |
| 14 | 65,000         | bin4                     | false |
| 13 | 95,500         | bin4                     | true  |
| 2  | 150,000        | bin4                     | false |
| 6  | 250,000        | bin4                     | true  |

- Once we have discretized the data we need to record the raw continuous feature threshold between the bins so that we can use these for query feature values.

**Table:** The thresholds used to discretize the LOAN AMOUNT feature in queries.

| Bin Thresholds |      |               |
|----------------|------|---------------|
|                | Bin1 | $\leq 9,925$  |
| $9,925 <$      | Bin2 | $\leq 19,250$ |
| $19,225 <$     | Bin3 | $\leq 49,000$ |
| $49,000 <$     | Bin4 |               |

**Table:** The Laplace smoothed (with  $k = 3$ ) probabilities needed by a naive Bayes prediction model calculated from the fraud detection dataset. Notation key: FR = FRAUD, CH = CREDIT HISTORY, AB = ACCOUNT BALANCE, GC = GUARANTOR/COAPPLICANT, ACC = ACCOMMODATION, BLA = BINNED LOAN AMOUNT.

|  |   |        |  |   |        |
|--|---|--------|--|---|--------|
| $P(fr)$  | = | 0.3    | $P(\neg fr)$   | = | 0.7    |
| $P(CH = none fr)$  | = | 0.2222 | $P(CH = none \neg fr)$   | = | 0.1154 |
| $P(CH = paid fr)$  | = | 0.2222 | $P(CH = paid \neg fr)$   | = | 0.2692 |
| $P(CH = current fr)$   | = | 0.3333 | $P(CH = current \neg fr)$  | = | 0.2692 |
| $P(CH = arrears fr)$   | = | 0.2222 | $P(CH = arrears \neg fr)$  | = | 0.3462 |
| $P(GC = none fr)$  | = | 0.5333 | $P(GC = none \neg fr)$   | = | 0.6522 |
| $P(GC = guarantor fr)$   | = | 0.2667 | $P(GC = guarantor \neg fr)$  | = | 0.1304 |
| $P(GC = coapplicant fr)$   | = | 0.2    | $P(GC = coapplicant \neg fr)$  | = | 0.2174 |
| $P(ACC = own fr)$  | = | 0.4667 | $P(ACC = own \neg fr)$   | = | 0.6087 |
| $P(ACC = rent fr)$   | = | 0.3333 | $P(ACC = rent \neg fr)$  | = | 0.2174 |
| $P(ACC = free fr)$   | = | 0.2    | $P(ACC = free \neg fr)$  | = | 0.1739 |
| $P(AB = x fr)$   |   |        | $P(AB = x \neg fr)$  |   |        |
| $\approx E\left(\begin{matrix} x, \\ \lambda = 0.0024 \end{matrix}\right)$ |   |        | $\approx N\left(\begin{matrix} x, \\ \mu = 984.26, \\ \sigma = 460.94 \end{matrix}\right)$ |   |        |
| $P(BLA = bin1 fr)$   | = | 0.3333 | $P(BLA = bin1 \neg fr)$  | = | 0.1923 |
| $P(BLA = bin2 fr)$   | = | 0.2222 | $P(BLA = bin2 \neg fr)$  | = | 0.2692 |
| $P(BLA = bin3 fr)$   | = | 0.1667 | $P(BLA = bin3 \neg fr)$  | = | 0.3077 |
| $P(BLA = bin4 fr)$   | = | 0.2778 | $P(BLA = bin4 \neg fr)$  | = | 0.2308 |

**Table:** A query loan application from the fraud detection domain.

| <b>Credit History</b> | <b>Guarantor/<br/>CoApplicant</b> | <b>Accomodation</b> | <b>Account<br/>Balance</b> | <b>Loan<br/>Amount</b> | <b>Fraudulent</b> |
|-----------------------|-----------------------------------|---------------------|----------------------------|------------------------|-------------------|
| paid                  | guarantor                         | free                | 759.07                     | 8,000                  | ?                 |

**Table:** The relevant smoothed probabilities, from Table 13 <sup>[37]</sup>, needed by the naive Bayes model to make a prediction for the query  $\langle CH = \text{'paid'}, GC = \text{'guarantor'}, ACC = \text{'free'}, AB = 759.07, LA = 8\,000 \rangle$  and the calculation of the scores for each candidate prediction.

|   |   |
|---|---|
| $P(fr) = 0.3$   | $P(\neg fr) = 0.7$  |
| $P(CH = \text{paid} fr) = 0.2222$   | $P(CH = \text{paid} \neg fr) = 0.2692$  |
| $P(GC = \text{guarantor} fr) = 0.2667$  | $P(GC = \text{guarantor} \neg fr) = 0.1304$   |
| $P(ACC = \text{free} fr) = 0.2$   | $P(ACC = \text{free} \neg fr) = 0.1739$   |
| $P(AB = 759.07 fr)$   | $P(AB = 759.07 \neg fr)$  |
| $\approx E\left(\begin{matrix} 759.07, \\ \lambda = 0.0024 \end{matrix}\right) = 0.00039$ | $\approx N\left(\begin{matrix} 759.07, \\ \mu = 984.26, \\ \sigma = 460.94 \end{matrix}\right) = 0.00077$ |
| $P(BLA = \text{bin1} fr) = 0.3333$  | $P(BLA = \text{bin1} \neg fr) = 0.1923$   |
| $(\prod_{k=1}^m P(\mathbf{q}[k]   fr)) \times P(fr) = 0.000000462$                        |   |
| $(\prod_{k=1}^n P(\mathbf{q}[k]   \neg fr)) \times P(\neg fr) = 0.000000633$              |   |

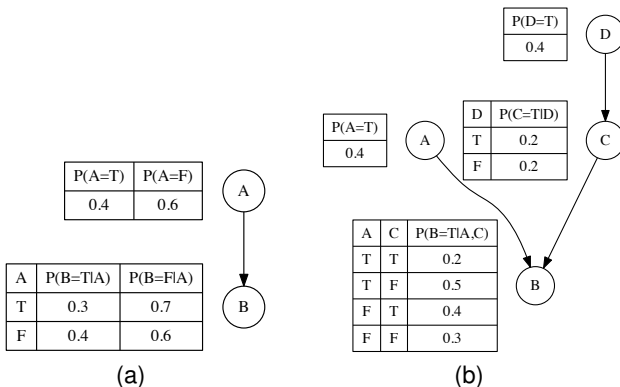
# Bayesian Networks



- **Bayesian networks** use a graph-based representation to encode the structural relationships—such as direct influence and conditional independence—between subsets of features in a domain.
- Consequently, a Bayesian network representation is generally more compact than a full joint distribution, yet is not forced to assert global conditional independence between all descriptive features.

A Bayesian Network is a directed acyclical graph that is composed of thee basic elements:

- nodes
- edges
- conditional probability tables (CPT)



**Figure:** (a) A Bayesian network for a domain consisting of two binary features. The structure of the network states that the value of feature A directly influences the value of feature B. (b) A Bayesian network consisting of 4 binary features with a path containing 3 generations of nodes: D, C, and B.

- In probability terms the directed edge from A to B in Figure (a) on the previous slide states that:

$$P(A, B) = P(B|A) \times P(A) \quad (1)$$

- For example, the probability of the event  $a$  and  $\neg b$  is

$$P(a, \neg b) = P(\neg b|a) \times P(a) = 0.7 \times 0.4 = 0.28$$

- Equation (1)<sup>[44]</sup> can be generalized to the statement that for any network with  $N$  nodes, the probability of an event  $x_1, \dots, x_n$ , can be computed using the following formula:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(x_i)) \quad (2)$$

- For example, using the more complex Bayesian network in figure (b) above, we can calculate the probability of the joint event  $P(a, \neg b, \neg c, d)$  as follows:

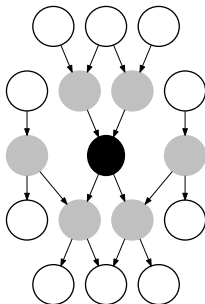
$$\begin{aligned}P(a, \neg b, \neg c, d) &= P(\neg b|a, \neg c) \times P(\neg c|d) \times P(a) \times P(d) \\&= 0.5 \times 0.8 \times 0.4 \times 0.4 = 0.064\end{aligned}$$

- We can use Bayes' Theorem to invert the dependencies between nodes in a network.
- Returning to the simpler network in figure (a) above we can calculate  $P(a|\neg b)$  as follows:

$$\begin{aligned}P(a|\neg b) &= \frac{P(\neg b|a) \times P(a)}{P(\neg b)} = \frac{P(\neg b|a) \times P(a)}{\sum_i P(\neg b|A_i)} \\&= \frac{P(\neg b|a) \times P(a)}{(P(\neg b|a) \times P(a)) + (P(\neg b|\neg a) \times P(\neg a))} \\&= \frac{0.7 \times 0.4}{(0.7 \times 0.4) + (0.6 \times 0.6)} = 0.4375\end{aligned}$$

- For conditional independence we need to take into account not only the parents of a node but also the state of its children and their parents.
- The set of nodes in a graph that make a node independent of the rest of the graph are known as the **Markov blanket** of a node.





**Figure:** A depiction of the Markov blanket of a node. The gray nodes define the Markov blanket of the black node. The black node is conditionally independent of the white nodes given the state of the gray nodes.

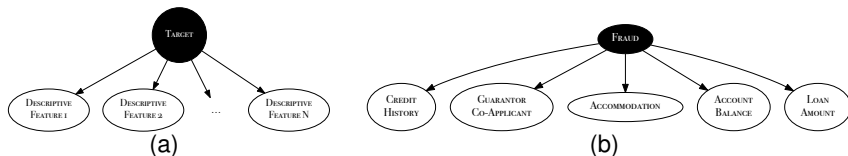
- The conditional independence of a node  $x_i$  in a graph with  $n$  nodes is defined as:

$$P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | \text{Parents}(x_i)) \prod_{j \in \text{Children}(x_i)} P(x_j | \text{Parents}(x_j)) \quad (3)$$

- Applying the equation of the preceding slide to the network in figure (b) above we can calculate the probability of  $P(c|\neg a, b, d)$  as

$$\begin{aligned}P(c|\neg a, b, d) &= P(c|d) \times P(b|c, \neg a) \\ &= 0.2 \times 0.4 = 0.08\end{aligned}$$

- A naive Bayes classifier is a Bayesian network with a specific topological structure.



**Figure:** (a) A Bayesian network representation of the conditional independence asserted by a naive Bayes model between the descriptive features given knowledge of the target feature; (b) a Bayesian network representation of the conditional independence assumption for the naive Bayes model in the fraud example.

- When we computed a conditional probability for a target feature using a naive Bayes model, we used the following calculation

$$P(t|\mathbf{d}[1], \dots, \mathbf{d}[n]) = P(t) \prod_{j \in \text{Children}(t)} P(\mathbf{d}[j]|t)$$

- This equation is equivalent to Equation (3)<sup>[50]</sup> from earlier.

- Computing a conditional probability for a node becomes more complex if the value of one or more of the parent nodes is unknown.

- For example, in the context of the network in figure (b) above, to compute  $P(b|a, d)$  where the status of node  $C$  is unknown we would do the following calculations:

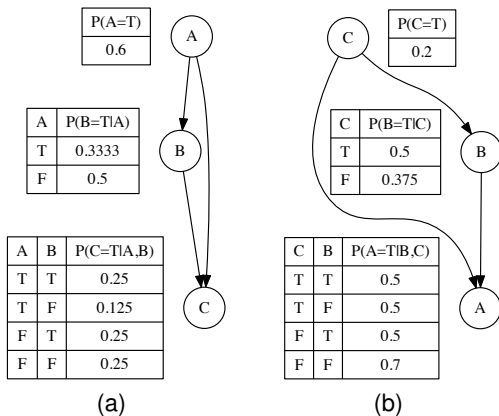
- 1 Compute the distribution for  $C$  given  $D$ :  $P(c | d) = 0.2$ ,  
 $P(\neg c | d) = 0.8$

- 2 Compute  $P(b | a, C)$  by summing out  $C$ :  
 $P(b | a, C) = \sum_i P(b | a, C_i)$

$$\begin{aligned} P(b | a, C) &= \sum_i P(b | a, C_i) = \sum_i \frac{P(b, a, C_i)}{P(a, C_i)} \\ &= \frac{(P(b | a, c) \times P(a) \times P(c)) + (P(b | a, \neg c) \times P(a) \times P(\neg c))}{(P(a) \times P(c)) + (P(a) \times P(\neg c))} \\ &= \frac{(0.2 \times 0.4 \times 0.2) + (0.5 \times 0.4 \times 0.8)}{(0.4 \times 0.2) + (0.4 \times 0.8)} = 0.44 \end{aligned}$$



- This example illustrates the power of Bayesian networks.
  - When complete knowledge of the state of all the nodes in the network is not available, we clamp the values of nodes that we do have knowledge of and sum out the unknown nodes.



**Figure:** Two different Bayesian networks, each defining the same full joint probability distribution.

- We can illustrate that these two networks encode the same joint probability distribution by using each network to compute  $P(\neg a, b, c)$
- Using network (a) we get:

$$\begin{aligned}P(\neg a, b, c) &= P(c|\neg a, b) \times P(b|\neg a) \times P(\neg a) \\ &= 0.25 \times 0.5 \times 0.4 = 0.05\end{aligned}$$

- Using network (b) we get:

$$\begin{aligned}P(\neg a, b, c) &= P(\neg a|c, b) \times P(b|c) \times P(c) \\ &= 0.5 \times 0.5 \times 0.2 = 0.05\end{aligned}$$

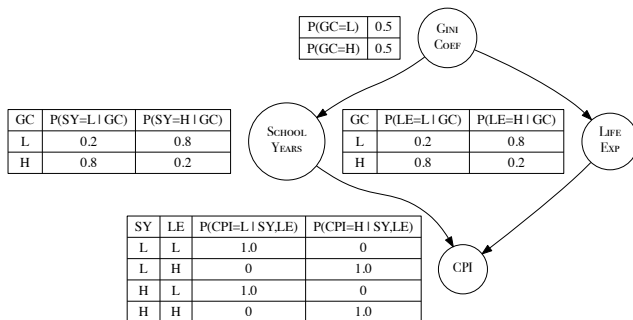
- The simplest way to construct a Bayesian network is to use a hybrid approach where:
  - 1 the topology of the network is given to the learning algorithm,
  - 2 and the learning task involves inducing the CPT from the data.

**Table:** (a) Some socio-economic data for a set of countries; (b) a binned version of the data listed in (a).

| COUNTRY<br>ID | GINI<br>COEF | SCHOOL<br>YEARS | LIFE<br>EXP | CPI  | GINI<br>COEF | SCHOOL<br>YEARS | LIFE<br>EXP | CPI  |
|---------------|--------------|-----------------|-------------|------|--------------|-----------------|-------------|------|
| Afghanistan   | 27.82        | 0.40            | 59.61       | 1.52 | low          | low             | low         | low  |
| Argentina     | 44.49        | 10.10           | 75.77       | 3.00 | high         | low             | low         | low  |
| Australia     | 35.19        | 11.50           | 82.09       | 8.84 | low          | high            | high        | high |
| Brazil        | 54.69        | 7.20            | 73.12       | 3.77 | high         | low             | low         | low  |
| Canada        | 32.56        | 14.20           | 80.99       | 8.67 | low          | high            | high        | high |
| China         | 42.06        | 6.40            | 74.87       | 3.64 | high         | low             | low         | low  |
| Egypt         | 30.77        | 5.30            | 70.48       | 2.86 | low          | low             | low         | low  |
| Germany       | 28.31        | 12.00           | 80.24       | 8.05 | low          | high            | high        | high |
| Haiti         | 59.21        | 3.40            | 45.00       | 1.80 | high         | low             | low         | low  |
| Ireland       | 34.28        | 11.50           | 80.15       | 7.54 | low          | high            | high        | high |
| Israel        | 39.2         | 12.50           | 81.30       | 5.81 | low          | high            | high        | high |
| New Zealand   | 36.17        | 12.30           | 80.67       | 9.46 | low          | high            | high        | high |
| Nigeria       | 48.83        | 4.10            | 51.30       | 2.45 | high         | low             | low         | low  |
| Russia        | 40.11        | 12.90           | 67.62       | 2.45 | high         | high            | low         | low  |
| Singapore     | 42.48        | 6.10            | 81.788      | 9.17 | high         | low             | high        | high |
| South Africa  | 63.14        | 8.50            | 54.547      | 4.08 | high         | low             | low         | low  |
| Sweden        | 25.00        | 12.80           | 81.43       | 9.30 | low          | high            | high        | high |
| U.K.          | 35.97        | 13.00           | 80.09       | 7.78 | low          | high            | high        | high |
| U.S.A         | 40.81        | 13.70           | 78.51       | 7.14 | high         | high            | high        | high |
| Zimbabwe      | 50.10        | 6.7             | 53.684      | 2.23 | high         | low             | low         | low  |

(a)

(b)



**Figure:** A Bayesian network that encodes the causal relationships between the features in the corruption domain. The CPT entries have been calculated using the data from Table 16 <sup>[61]</sup>(b).

$$\mathbb{M}(\mathbf{q}) = \arg \max_{l \in \text{levels}(t)} \text{BayesianNetwork}(t = l, \mathbf{q}) \quad (4)$$

## Example

- We wish to predict the CPI for a country with the follow profile:

GINI COEF = *'high'*, SCHOOL YEARS = *'high'*



$$\begin{aligned} P(CPI = H | SY = H, GC = H) &= \frac{P(CPI = H, SY = H, GC = H)}{P(SY = H, GC = H)} \\ &= \frac{\sum_{i \in H, L} P(CPI = H, SY = H, GC = H, LE = i)}{P(SY = H, GC = H)} \end{aligned}$$

$$\begin{aligned} & \sum_{i \in \{H, L\}} P(CPI = H, SY = H, GC = H, LE = i) \\ &= \sum_{i \in \{H, L\}} P(CPI = H | SY = H, LE = i) \times P(SY = H | GC = H) \\ & \quad \times P(LE = i | GC = H) \times P(GC = H) \\ &= (P(CPI = H | SY = H, LE = H) \times P(SY = H | GC = H) \\ & \quad \times P(LE = H | GC = H) \times P(GC = H)) \\ & \quad + (P(CPI = H | SY = H, LE = L) \times P(SY = H | GC = H) \\ & \quad \times P(LE = L | GC = H) \times P(GC = H)) \\ &= (1.0 \times 0.2 \times 0.2 \times 0.5) + (0 \times 0.2 \times 0.8 \times 0.5) = 0.02 \end{aligned}$$

$$\begin{aligned}P(SY = H, GC = H) &= P(SY = H | GC = H) \times P(GC = H) \\&= 0.2 \times 0.5 = 0.1\end{aligned}$$

$$P(CPI = H | SY = H, GC = H) = \frac{0.02}{0.1} = 0.2$$

- Because of the calculation complexity that can arise when using Bayesian networks to do exact inference a popular approach is to approximate the required probability distribution using **Markov Chain Monte Carlo** algorithms.
- **Gibbs sampling** is one of the best known MCMC algorithms.
  - 1 Clamp the values of the evidence variables and randomly assign the values of the non-evidence variables.
  - 2 Generate samples by changing the value of one of the non-evidence variables using the distribution for the node conditioned on the state of the rest of the network.

**Table:** Examples of the samples generated using Gibbs sampling.

| Sample Number | Gibbs Iteration | Feature Updated | GINI COEF | SCHOOL YEARS | LIFE EXP | CPI  |
|---------------|-----------------|-----------------|-----------|--------------|----------|------|
| 1             | 37              | CPI             | high      | high         | high     | low  |
| 2             | 44              | LIFE EXP        | high      | high         | high     | low  |
| 3             | 51              | CPI             | high      | high         | high     | low  |
| 4             | 58              | LIFE EXP        | high      | high         | low      | high |
| 5             | 65              | CPI             | high      | high         | high     | low  |
| 6             | 72              | LIFE EXP        | high      | high         | high     | low  |
| 7             | 79              | CPI             | high      | high         | low      | high |
| 8             | 86              | LIFE EXP        | high      | high         | low      | low  |
| 9             | 93              | CPI             | high      | high         | high     | low  |
| 10            | 100             | LIFE EXP        | high      | high         | high     | low  |
| 11            | 107             | CPI             | high      | high         | low      | high |
| 12            | 114             | LIFE EXP        | high      | high         | high     | low  |
| 13            | 121             | CPI             | high      | high         | high     | low  |
| 14            | 128             | LIFE EXP        | high      | high         | high     | low  |
| 15            | 135             | CPI             | high      | high         | high     | low  |
| 16            | 142             | LIFE EXP        | high      | high         | low      | low  |

$$\mathbb{M}(\mathbf{q}) = \arg \max_{l \in \text{levels}(t)} \text{Gibbs}(t = l, \mathbf{q}) \quad (5)$$

# Summary



- Naive Bayes models can suffer from zero probabilities of relatively rare events. **Smoothing** is an easy way to combat this.
- Two ways to handle continuous features in probability-based models are: **Probability density functions** and **Binning**
- Using probability density functions requires that we match the observed data to an existing distribution.
- Although binning results in information loss it is a simple and effective way to handle continuous features in probability-based models.
- Bayesian network representation is generally more compact than a full joint distribution, yet is not forced to assert global conditional independence between all descriptive features.

- 1 Smoothing
- 2 Continuous Features: Probability Density Functions
- 3 Continuous Features: Binning
- 4 Bayesian Networks
- 5 Summary