ε.δε.μ²
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

1. Non-Parametric Classifiers, *K*-Nearest Neighbors - KNN
2. Statistical Evaluation of Binary Classification: Confusion Matrix, ROC, AUC
3. Parametric Probabilistic Classification: MLE & MAP Estimation, Bayes & Naive Bayes Classifiers

Prof. Vasilis Maglaris
maglaris@netmode.ntua.gr
www.netmode.ntua.gr
Room 002, New ECE Building
Tuesday May 13, 2025

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Classification based on *K*-Nearest Neighbors (1/2)

https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/

- **Non-Parametric Methods of Supervised Learning**: Based on *distance* estimates from known sample vectors (*patterns*), without assuming a *probabilistic model* for the (labeled) training dataset that would determine the *form* of the function $y = h(\mathbf{x})$.
- By contrast, **Parametric Methods** are based on pre-selecting a *form* for the input-output function and learn its *coefficients* from the (labeled) training data (e.g. *Linear* & *Logistic Regression*, *Perceptron*, *Simple Neural Networks*)

K-Nearest Neighbors Algorithm (KNN)

Assume a *training* set $\{\mathbf{x}(n), d(n)\}, n = 1,2, \dots, N$ of *vectors* $\mathbf{x}(n) = [x_1(n)\ x_2(n) \dots x_m(n)]^{\mathrm{T}}$ with *labels* $d(n) \to \mathcal{C}$ indicating the class of $\mathbf{x}(n)$ (e.g. $\mathcal{C} \in \{0,1\}$ for binary classification) and a distance *metric*, e.g. *Euclidean Distance* $\|\mathbf{x}(i), \mathbf{x}(j)\| = \sqrt{\sum_{k=1}^{m}\left(x_k(i) - x_k(j)\right)^2}$) or *Hamming Distance* - number of opposite binary digits between $(\mathbf{x}(i), \mathbf{x}(j))$

- **Training Phase** (*Lazy Learning Method*): Storage of training sample (N *labeled* patterns)
- **Test Phase**: A new vector $\hat{\mathbf{x}}$ is classified according to the *labels* of the *majority* of K nearest neighbors amongst the N training *patterns* (vectors) $\mathbf{x}(n)$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

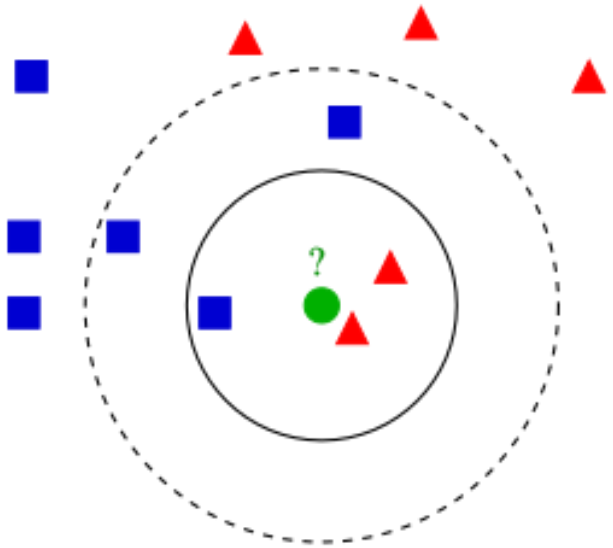
Classification based on *K*-Nearest Neighbors (2/2)

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

K: Positive Integer **hyperparameter**, *odd* for binary classification, usually selected after *cross validation* trials

$K = 1$: A new vector $\hat{\mathbf{x}}$ is classified according to the class of its nearest neighbor (*pattern*)

$K \gg 1$: Tolerant in distortion/noise of classification regions *but* with increased storage requirements and prone to undesirable impact of *outliers*, i.e. rare patterns with extreme characteristics: Such cases require vector normalization, reduction of coordinates and filtering of *outliers*



Binary Classification

$$\mathcal{C} \in \{Blue\_Square, Red\_Triangle\}$$

$$\text{Test } \hat{\mathbf{x}} : \{Green\_Circle\}$$

$$K = 3 \rightarrow \text{Test} \in Red_{Triangle}$$

$$K = 5 \rightarrow \text{Test} \in Blue\_Square$$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Statistical Evaluation of Binary Classification: Confusion Matrix, ROC, AUC (1/3)

Statistical Binary Classification – Parametric Classifiers

Classification of examples (vectors, patterns) of a sample in 2 classes $\mathcal{C} \in \{\boldsymbol{P}, \boldsymbol{N}\}$:

Positive $\boldsymbol{P}$, ***Negative*** $\boldsymbol{N}$

- Diagnosis of infections: **Positive ≙ Infected Sample Element**
- Identification of anomalies: **Positive ≙ Anomalous Sample Element** (*outlier*)
- Recognition of binary patterns (e.g. cats - dogs): **Positive ≙ Cat**, **Negative ≙ Dog**
- Target identification signalling: **Positive ≙ Foe**, **Negative ≙ Friend** (**IFF:** Identify Friend or Foe)

The **parametric** classification algorithms identify parameters of pre-selected input-output function $y = h(\mathbf{x})$, conforming to *training data*:

- Assumption of ***Gaussian*** distribution for the sample space to be identified with 2 parameters *mean value* - *variance*
- Examples of Parametric Methods:
  - Parameter tuning in ***logistic regression*** from a ***labeled*** training sample
  - Parameter tuning of synaptic weights of ***perceptron*** neural networks via iterative ***back-propagation*** of output sample elements and convergence to statistically inferred properties from training datasets

https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/
Parametric Methods: Linear & Logistic Regression, Perceptron Convergence, Bayes Classifiers…
Non-Parametric Methods: KNN, Decision Trees, SVM…

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Statistical Evaluation of Binary Classification: Confusion Matrix, ROC, AUC (2/3)

Confusion Matrix

- **Incorrect Predictions**: False Positives - **FP**, False Negatives - **FN**
- **Correct Predictions**: True Positives - **TP**, True Negatives - **TN**

Rates of Correct/Incorrect Predictions:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \qquad \mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} = 1-\mathrm{TPR}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1-\mathrm{TNR}$$

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Confusion Matrix

Binary Classifier Evaluation Metrics

Accuracy: $\mathbf{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$ (ratio of total *correct* predictions)

Sensitivity, **Recall**: $\mathbf{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ (*correct* positive predictions out of *actual* positive examples)

Precision: $\mathbf{PRE} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$ (*correct* positive predictions out of *positive* predictions)

F1-Score: $\mathbf{F1} = \frac{\mathrm{TP}}{\mathrm{TP}+\frac{1}{2}(\mathrm{FP}+\mathrm{FN})} = \frac{2}{\mathbf{TPR}^{-1}+\mathbf{ACC}^{-1}}$ (harmonic mean of *recall* & *precision*)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Statistical Evaluation of Binary Classification: Confusion Matrix, ROC, AUC (3/3)

Example of Image Recognition: Cat or Dog

https://en.wikipedia.org/wiki/Confusion_matrix

Test Sample 12 images: 8 cats (class $\boldsymbol{P}$), 4 dogs (class $\boldsymbol{N}$)

The binary classifier predicts after training 7 cats and 5 dogs (9 correct, 3 incorrect) as shown in the ***Confusion Matrix***:

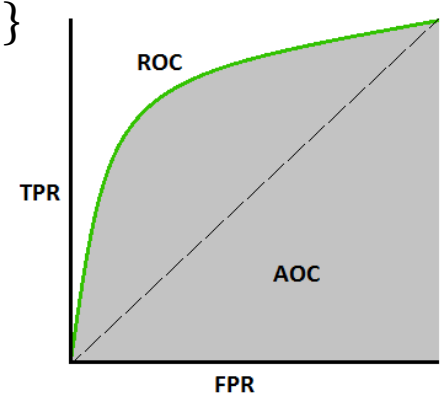
Actual class \ Predicted class	Cat	Dog
Cat	6	2
Dog	1	3

- Accuracy: $\mathbf{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} = \frac{6+3}{12} = 3/4$
- Sensitivity (Recall): $\mathbf{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \frac{6}{6+2} = 3/4$

Receiver Operating Characteristics (ROC), Area Under the Curve (AUC)

https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

- Definitions from target identification in radar receivers dyring WW2
- Operational separation choices (***threshold values***) in binary classification (***Positive*** or ***Negative*** Prediction) depending on the administrator preference to select among ***Receiver Operating Characteristics*** **(ROC)** points
- Graph **ROC**: Function $\mathrm{FPR} \to \mathrm{TPR}, \{0 \le \mathrm{FPR} \le 1,\ 0 \le \mathrm{TPR} \le 1\}$
- Good operational choice in **ROC**: $\mathrm{TPR} \gg \mathrm{FPR}$
- Ideal choice: $\mathrm{TPR} = 1, \mathrm{FPR} = 0$



Area under ROC: **AUC** (***A***rea ***U***nder the ***C***urve) for $0 \le \mathrm{FPR} \le 1$

Measure of ***Separation Capability*** of a Classifier

- No separation capability: $\mathbf{AUC} = 0.5$
- Separation excellence: $\mathbf{AUC} \gg 0.5$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Probabilistic Classification Models, Bayesian MLE & MAP Estimation (1/3)

(T. Mitchell 2016, "*Machine Learning*" http://www.cs.cmu.edu/~tom/mlbook/Joint_MLE_MAP.pdf)

Bayes Rule

$$P(\mathrm{y}|\mathbf{x}) = \frac{P(\mathbf{x}, y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

Bayes Parametric Estimation Model

- **Assumption**: Elements (examples) $\mathbf{x}(i) \in \{\boldsymbol{X}\}$ of the sample-space are distributed according to a ***known*** distribution (e.g. ***Gauss***) into ***discrete classes*** $\mathcal{C}$ based on discrete probabilities of a parameter θ of the sample-space, e.g. mean of $\mathbf{x}(i)$ for every class
- Estimators $\hat{\theta}$ of parameter θ indicate the ***class*** $\mathcal{C}$ of an example $\mathbf{x}(i)$
- The $\hat{\theta}$ are inferred from observing examples $\mathbf{x}(i)$ of a ***training sample*** $\boldsymbol{\mathcal{D}}$, a subset of the sample-space $\{\boldsymbol{X}\}$ with outputs or ***labels*** $d(i) = \theta$ known to a supervisor (***teacher***) (***supervised learning***)
- According to the ***Bayes Rule***:

$$P(\theta|\boldsymbol{\mathcal{D}}) = \frac{P(\boldsymbol{\mathcal{D}}|\theta)P(\theta)}{P(\boldsymbol{\mathcal{D}})}$$

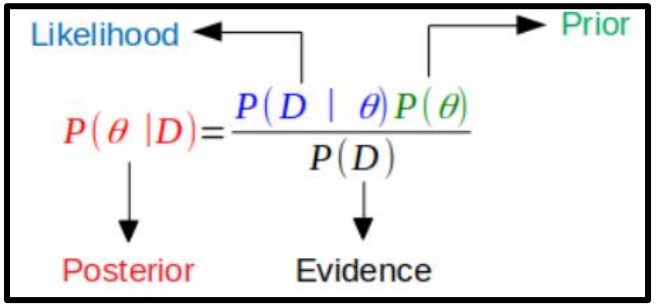
where

- $P(\boldsymbol{\mathcal{D}})$: **Evidence**, joint probability of observing examples $\mathbf{x}(i)$ in $\boldsymbol{\mathcal{D}}$, sampled from $\{\boldsymbol{X}\}$
- $P(\theta)$: **Prior** probability of parameter θ assigned to examples in the sample-space $\{\boldsymbol{X}\}$
- $P(\boldsymbol{\mathcal{D}} \mid \theta)$: **Likelihood**, joint probability of all examples in $\boldsymbol{\mathcal{D}}$ conditioned to parameter θ
- $P(\theta \mid \boldsymbol{\mathcal{D}})$: **Posterior** probability of parameter θ for all examples in $\boldsymbol{\mathcal{D}}$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Probabilistic Classification Models, Bayesian MLE & MAP Estimation (2/3)

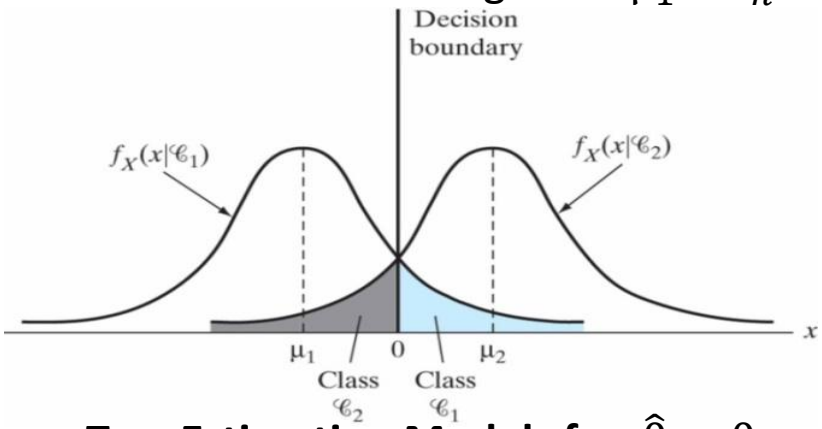
http://www.cs.cmu.edu/~tom/mlbook/Joint_MLE_MAP.pdf



$P(\theta)$: *Prior Assumption* based on user-experience of $\{\boldsymbol{X}\}$ *or* inferred from the *observed labeled training subset* $\boldsymbol{\mathcal{D}} \subset \{\boldsymbol{X}\}$ used to provide *evidence* of sample-space statistics

A Binary Parametric Classifier

Assumption: Examples x_n are ***Gaussian*** with average $\theta = \mu_1$ if $x_n \rightarrow \mathcal{C}_1$ or $\theta = \mu_2$ if $x_n \rightarrow \mathcal{C}_2$



Likelihood Densities: $f_X(x|\mathcal{C}_1), f_X(x|\mathcal{C}_2)$

Classification Errors: Shaded areas

Two Estimation Models for $\hat{\theta} \approx \theta$

1. Maximum Likelihood Estimation (**MLE**): Select θ that makes $\boldsymbol{\mathcal{D}}$ the most probable option

$$\hat{\theta} = \arg\max_{\theta} P(\boldsymbol{\mathcal{D}}|\theta)$$

2. Maximum a Posteriori Probability (**MAP**) Estimation : Select the most probable θ given the observed data $\boldsymbol{\mathcal{D}}$ and our *prior assumptions* summarized by $P(\theta)$

$$\hat{\theta} = \arg\max_{\theta} P(\theta|\boldsymbol{\mathcal{D}}) = \arg\max_{\theta} \frac{P(\boldsymbol{\mathcal{D}}|\theta)P(\theta)}{P(\boldsymbol{\mathcal{D}})} = \arg\max_{\theta} P(\boldsymbol{\mathcal{D}}|\theta)P(\theta)$$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Probabilistic Classification Models, Bayesian MLE & MAP Estimation (3/3)

(T. Mitchell 2016, "*Machine Learning*" http://www.cs.cmu.edu/~tom/mlbook/Joint_MLE_MAP.pdf)

Example: ***Bernoulli*** experiment for tossing a coin with output $X \in \{heads, tails\} \triangleq \{1,0\}$

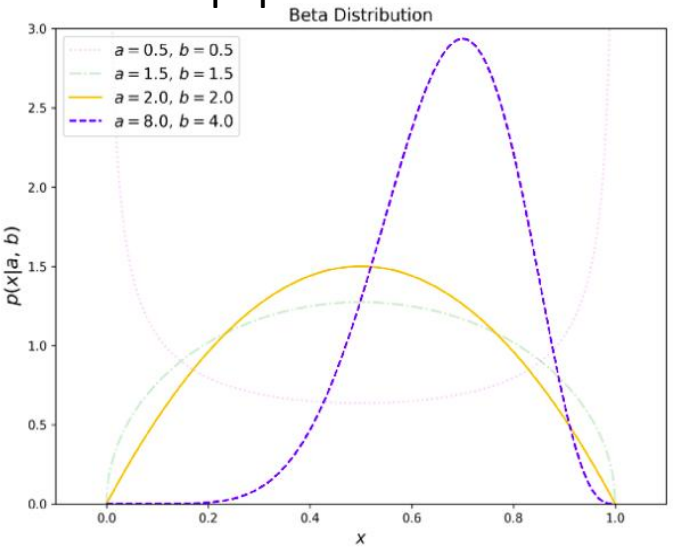
- Training Sample $\boldsymbol{\mathcal{D}} = \{x(1), x(2), \dots, x(50)\}$ of 50 ***trials*** (examples) used to infer $\hat{\theta}$ as an estimate of the $heads$ probability $\theta = P(X = 1)$
- If the trials resulted into $a_1 = 24\ heads, a_0 = 26\ tails$, the **MLE** estimate of θ is

$$\hat{\theta} = \arg\max_{\theta} P(\boldsymbol{\mathcal{D}}|\theta) = \frac{a_1}{(a_1+a_0)} = \mathbf{0.48}, \text{ with } P(\boldsymbol{\mathcal{D}}|\theta) = \theta^{a_1}(1-\theta)^{a_0}$$

- For **MAP** estimation the *Priors* $P(\theta)$ are needed, e.g. from empirical understanding of the sampling process. In case we believe that the coin favors $heads$ with $P(1) \approx 0.6$ we may modify the experiment adding to observed a_1, a_2 some *biased* imaginary examples β_1, β_2:

$$a_1 \to a_1 + \beta_1 = 24 + 9 = 33,\ a_0 \to a_0 + \beta_0 = 26 + 1 = 27 \text{ and } \hat{\theta} = \frac{33}{33+27} = \mathbf{0.55} \approx 0.6$$

- With equiprobable choices of θ, **MAP ≡ MLE**



https://towardsdatascience.com/understanding-conjugate-priors-21b2824cddae

Justification of MAP Estimation

- The *Likelihood* $P(\boldsymbol{\mathcal{D}}|\theta) = \theta^{a_1}(1-\theta)^{a_0}$ is *Bernoulli*: $Beta(a_1, a_0)$
- Assume that the *Prior* $P(\theta)$ is $P(\theta) \propto K\theta^{\beta_1-1}(1-\theta)^{\beta_0-1} \triangleq Beta(\beta_1, \beta_0)$
- If the *Posterior* $P(\theta|\boldsymbol{\mathcal{D}})$ is similar in form with $P(\theta)$ (***Conjugate Distributions***) then: $P(\theta|\boldsymbol{\mathcal{D}}) \propto Beta(\alpha_1 + \beta_1, \alpha_0 + \beta_0)$
- The estimate $\hat{\theta} = \arg\max_{\theta} P(\theta|\boldsymbol{\mathcal{D}})$ follows from the maximum of $Beta(33,27)$ that occurs at $\theta \cong \mathbf{0.55} = \hat{\theta}$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Probabilistic Classification Models, Bayesian MLE & MAP Estimation (3/3)

(T. Mitchell 2016, "*Machine Learning*" http://www.cs.cmu.edu/~tom/mlbook/Joint_MLE_MAP.pdf)

Probabilities ~ Relative Frequency of examples $\{\mathbf{x}(i), d(i)\}$ in the Training Set $\mathcal{D}$

- Input $\mathbf{x}(i) = (Gender, HoursWorked)$ of $m = 2$ binary coordinates (features)
- Output (label) $d(i) \cong y(i) = h(\mathbf{x}(i)) = Wealth \in \{\text{poor}, \text{rich}\}$

Gender	HoursWorked	Wealth	probability
female	< 40.5	poor	0.2531
female	< 40.5	rich	0.0246
female	≥ 40.5	poor	0.0422
female	≥ 40.5	rich	0.0116
male	< 40.5	poor	0.3313
male	< 40.5	rich	0.0972
male	≥ 40.5	poor	0.1341
male	≥ 40.5	rich	0.1059

Estimates of Probabilities
$P(\mathbf{x}, y) = P(\text{G}, \text{HW}, y)$

where

$\text{G} \in \{M, F\}$
$\text{HW} \in \{light, hard\}$
$y \in \{\text{poor}, \text{rich}\}$

Posterior $P(y|\mathbf{x})$: $P(\text{rich}|F, light) = \frac{0.0246}{0.2531+0.0246} \sim \mathbf{0.09}$

Gender (G)	HrsWorked (HW)	*P*(rich | G,HW)	*P*(poor | G,HW)
F	<40.5 (*light*)	**0.09**	0.91
F	>40.5 (*hard*)	0.21	0.79
M	<40.5 (*light*)	0.23	0.77
M	>40.5 (*hard*)	0.38	0.62

$m = 2$ features $\{G, HW\}$ require 4 posterior probabilities (m features require 2^m)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Conditional Independence Assumption, Naïve Bayes Classifier (1/2)

(T. Mitchell 2016, "*Machine Learning*" http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf)

Bayes rule for *Random Variables* X, Y: $P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}$

Conditional Independence of *Random Variable* $\{X|Y,Z\}$ from Y: $P(X|Y,Z) = P(X|Z)$

Simplifying Approximation - Naive Bayes Classifier

- For a two-dimensional sample with elements $\mathbf{x} = [x_1\ x_2]^{\mathrm{T}}$ assume that x_i's are *Conditionally Independent* from the *output* y: $P(x_1|x_2,y) \cong P(x_1|y)$. Then:

$$P(\mathbf{x}|y) = \mathrm{P}(x_1,x_2|y) = P(x_1|x_2,y) \times P(x_2|y) \cong P(x_1|y) \times P(x_2|y)$$

- Generalizing to m *features* of sample elements $\mathbf{x} = [x_1\ x_2\ \dots x_m]^{\mathrm{T}}$ the *likelihood* is:

$$P(\mathbf{x}|y) = P(x_1,x_2,\dots,x_m\,|y) \cong \prod_{k=1}^{m} P(x_k|y)$$

- The ***Naive Bayes Classifier*** is based on estimating the *posterior* $P(d|\mathbf{x}) \cong P(y|\mathbf{x})$ based on the *training sample likelihoods* $P(\mathbf{x}|y)$:

$$P(y|\mathbf{x}) = \frac{P(y)P(\mathbf{x}|y)}{P(\mathbf{x})} \propto P(y)P(x_1|y)P(x_2|y)\dots P(x_m|y)$$

- A new ***test*** sample point $\mathbf{x}^{new} = [x_1^{new}\ x_2^{new}\ \dots x_m^{new}]^{\mathrm{T}}$ requires $\sim m$ likelihoods for classification instead of 2^m, a significant simplification to counter the ***curse of dimensionality***

The *priors* $P(y)$ are approximated from the occurrence frequency in the training set, assuming a ***Multinomial Distribution*** for **discrete** x_i **values**, or ***Gaussian Distribution*** (***Gaussian Naive Bayes Classifier***) for **continuous** x_i

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Conditional Independence Assumption, Naïve Bayes Classifier (2/2)

(T. Mitchell 2016, "*Machine Learning*" http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf

The ***Naive Bayes Classifier*** is based on approximating the *posterior* $P(d|\mathbf{x}) \cong P(y|\mathbf{x})$ from the product of ***conditionally independent*** feature *likelihoods*

$$P(y|\mathbf{x}) \propto P(y)P(x_1|y)P(x_2|y) \dots P(x_m|y)$$

Naive Bayes Algorithm:

From a *labeled training sample* $\boldsymbol{\mathcal{D}} = \{(\mathbf{x}(1), d(1)), \dots, (\mathbf{x}(N), d(N))\}$ estimate:

- The *priors* $P(d) \cong P(y)$ for all possible *classes* d, e.g. $d \in \{0,1\}$ for binary classification
- The *likelihood* $P(x_k = l|y = d) \triangleq \theta_{kld}$ for every *discrete* feature x_k, $k = 1,2, \dots, m$ of the *training elements* $\mathbf{x}$ that were classified within class (*label*) d

A new example of the ***test*** sample $\mathbf{x}^{new} = [x_1^{new}\ x_2^{new} \dots x_m^{new}]^{\mathrm{T}}$, $x_k^{new} = l$ will be assigned to ***class*** y^{new} that will result from:

$$y^{new} \leftarrow \arg\max_y P(y) \prod_{k=1}^{m} P(x_k^{new}|y)$$

or

$$y^{new} \leftarrow \arg\max_d P(d) \prod_{k=1}^{m} \theta_{kld}$$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Example of a Naïve Bayes Classifier

https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf

Training Sample of 1000 Training Examples

Fruit	Long	Sweet	Yellow	Total
Banana	**400** (80%)	**350** (70%)	**450** (90%)	**500** (50%)
Orange	**0** (0%)	**150** (50%)	**300** (100%)	**300** (30%)
Other	**100** (50%)	**150** (75%)	**50** (25%)	**200** (20%)
Total	**500** (50%)	**650** (65%)	**800** (80%)	**1000**

$$P(y|\mathbf{x}) \propto P(y)P(x_1|y)P(x_2|y) \dots P(x_m|y)$$

- $P(\text{Banana}|\text{Long, Sweet, Yellow}) \propto (0.5) \times (0.8) \times (0.7) \times (0.9) = 0.252$
- $P(\text{Orange}|\text{Long, Sweet, Yellow}) = 0$
- $P(\text{Other}|\text{Long, Sweet, Yellow}) \propto (0.2) \times (0.5) \times (0.75) \times (0.25) = 0.01875$

Classification of a New (Test) Example

Fruits with features (characteristics) $\mathbf{x} = (\text{Long, Sweet, Yellow})$ are classified as $y = (\text{Banana})$ with the greatest posterior probability $P(y|\mathbf{x}) \propto 0.252$

Note: The value of the posterior can be determined by normalizing to 1 the sum of probabilities

$$P(y|\mathbf{x}) = \frac{0.252}{0.252 + 0.01875} = 0.931$$