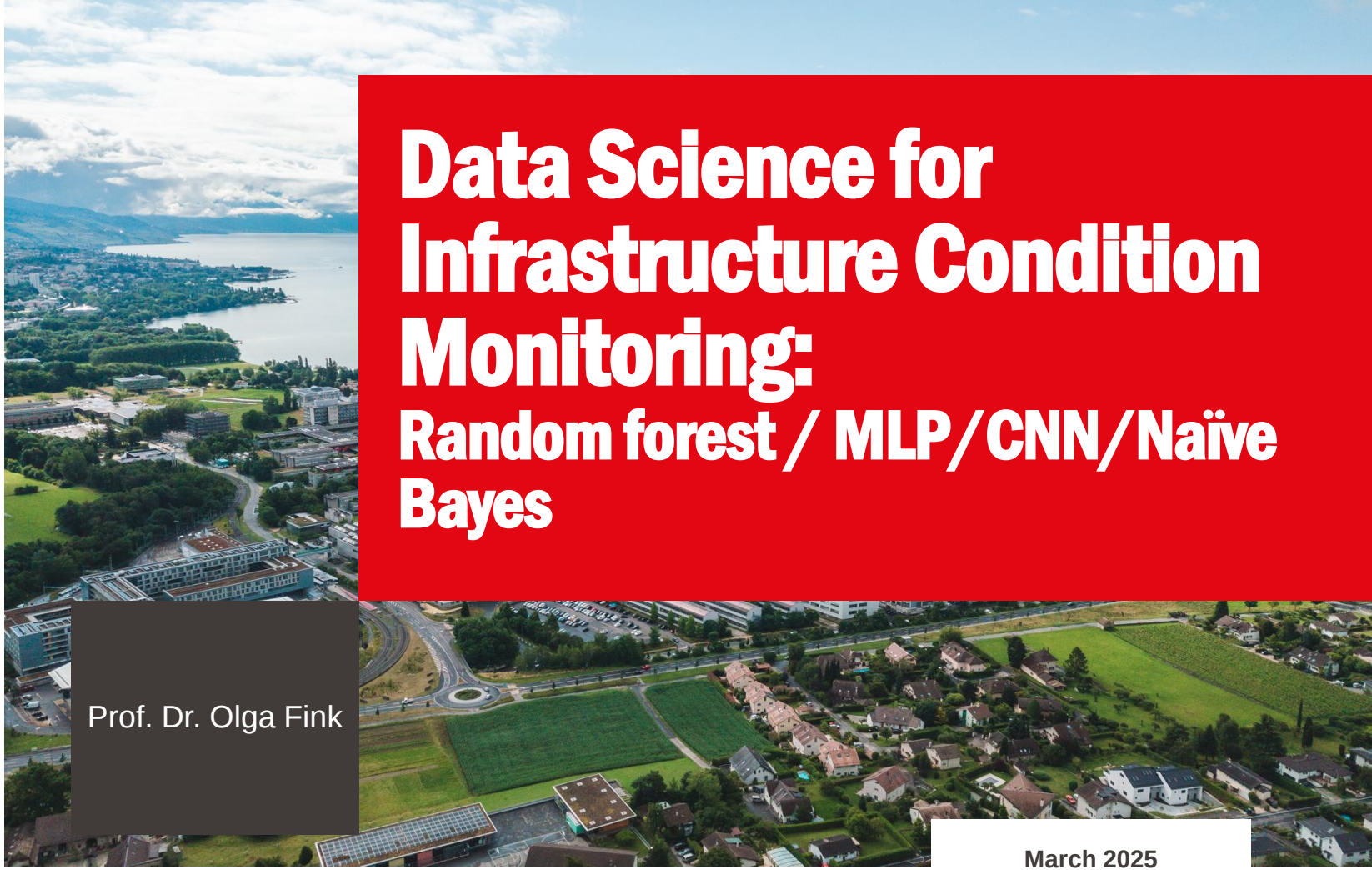


Data Science for Infrastructure Condition Monitoring:

Random forest / MLP/CNN/Naïve Bayes

Prof. Dr. Olga Fink

École polytechnique fédérale de Lausanne

March 2025

Computing probabilities: Example

- Let's suppose you were worried that you might have a rare disease. You decide to have a test done.
- The test gives a correct result in 99% of cases (99% correct if you have no disease and 99% correct if you have a disease).
- Disease is very rare: 1 in 10,000 people in the population is affected.
- The test is positive. With what probability are you actually ill?

Bayes Theorem

Theorem of Bayes

- P (A | B): the (conditional) probability of A, given that B has occurred
- P(A) und P(B) are a-priori probabilities.

$$P(A|B) = \frac{P(B|A) \bullet P(A)}{P(B)}$$

$$P(A|B) = \frac{P(B|A) \bullet P(A)}{P(B|A) \bullet P(A) + P(B|\overline{A}) \bullet P(\overline{A})}$$

Statistical tests

What Are Statistical Tests?

- Tools to make **inferences** about populations from sample data.
- Help determine if **observed differences** or **relationships** are **statistically significant**.
- **Why Are They Important?**
  - Distinguish **real effects** from **random variation**.
  - Support **hypothesis testing** in experiments and data analysis.
  - Provide evidence for **decision-making** in research and practice.

Key Concepts

- **Null Hypothesis (H_0):** No effect or difference.
- **Alternative Hypothesis (H_1):** There is an effect or difference.
- **P-value:** Probability of observing data under H_0.
- **Significance Level (α):** Common threshold = **0.05**.

Test	Purpose
t-test	Compare means between two groups
ANOVA	Compare means among three or more groups
Chi-Square Test	Association between categorical variables
Mann-Whitney U	Compare two groups, non-parametric
Wilcoxon Signed-Rank	Compare paired samples, non-parametric

t-Test

- Compares the means of two groups.
- Types:
  - Independent samples t-test: Compares means of two independent groups.
  - Paired samples t-test: Compares means from the same group at different times.
- Assumptions:
  - Data is continuous and normally distributed.
  - Variances are equal (homogeneity of variance).
  - Observations are independent.
- Example: Comparing the mean strain measurements of two different types of bridge materials under similar load conditions to determine if one material performs significantly better in reducing stress.

Hypothesis testing: Wald Test

- The Wald test (a.k.a. Wald Chi-Squared Test) is a parametric statistical measure to confirm whether a set of independent variables are collectively ‘significant’ for a model or not.
- Also used for confirming whether each independent variable present in a model is significant or not.
- A variable is said to be ‘significant’ if that variable adds some incremental value to the model.
- Variables which fail to add value to the model, can be omitted without affecting the model in any meaningful way.
- If the Wald test shows that the parameters for certain explanatory variables are zero, we can remove the variables from the model.
- If the test shows the parameters are not zero, we should include the variables in the model
- This test is widely used in logistic regression, linear regression, and many other statistical models for hypothesis testing.

Wald test

$$W = \frac{(\hat{\beta})^2}{SE(\hat{\beta})^2}$$

- where $\widehat{\beta}$ is the estimate of the parameter of interest and SE($\hat{\beta}$)is the standard error of that estimate.
- The standard error (SE) in the context of the Wald test represents the estimated variability or precision of the parameter estimate $\widehat{\beta}$. Essentially, it measures how much the estimate of the coefficient is expected to vary across different samples drawn from the same population. The smaller the SE, the more precise the estimate is considered to be.
- This statistic, which follows a chi-square distribution, tests the null hypothesis that $\hat{\beta}$ equals zero (implying the variable has no effect). If W exceeds the chi-square critical value at a chosen significance level, the null hypothesis is rejected, suggesting the parameter significantly differs from zero.

Example 1: Logistic regression for infrastructure condition monitoring

Water Pipe Failure Prediction

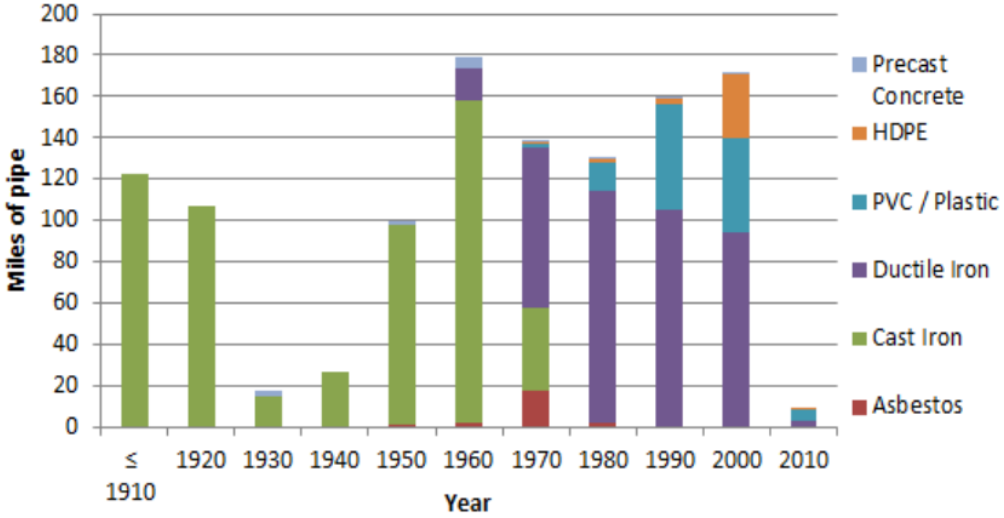


Figure 1. Miles of pipe installed by material (data from the 2014 Asset Management Report of the municipality)

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Water Pipe Failure Prediction

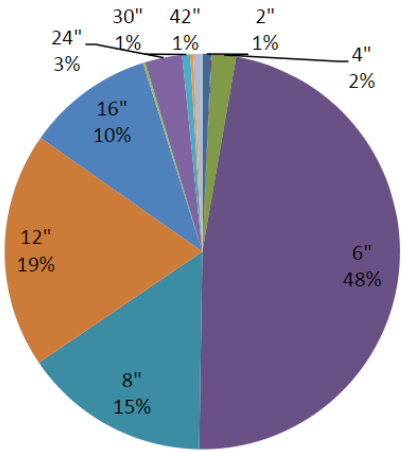


Figure 2. Water network analysis by pipe diameter (data from the 2014 Asset Management Report of the municipality)

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Water Pipe Failure Prediction

Table 1. Data categorization for developing the predictive failure model

Independent Variable	Category 1	Category 2	Category 3
Pipe diameter	6”	8”	12”

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Water Pipe Failure Prediction

Table 2. Log likelihood test of statistical significance of independent variables

Independent Variables in the Nested Model	Chi-square	D. F.	Sig.	Critical Value (95 %)	p value for critical level	Results
Age	10.35	1	0.001	3.841	0.05	Age is a significant variable
Age, Diameter	128.752	1	0.375	3.841	0.05	Diameter is not a significant variable

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Water Pipe Failure Prediction

$$f(x) = \log_e(P) = \log\left[\frac{P}{1-P}\right] = 15.25 - 0.131 * \text{Age} - 0.011 * \text{Diameter}$$

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Water Pipe Failure Prediction

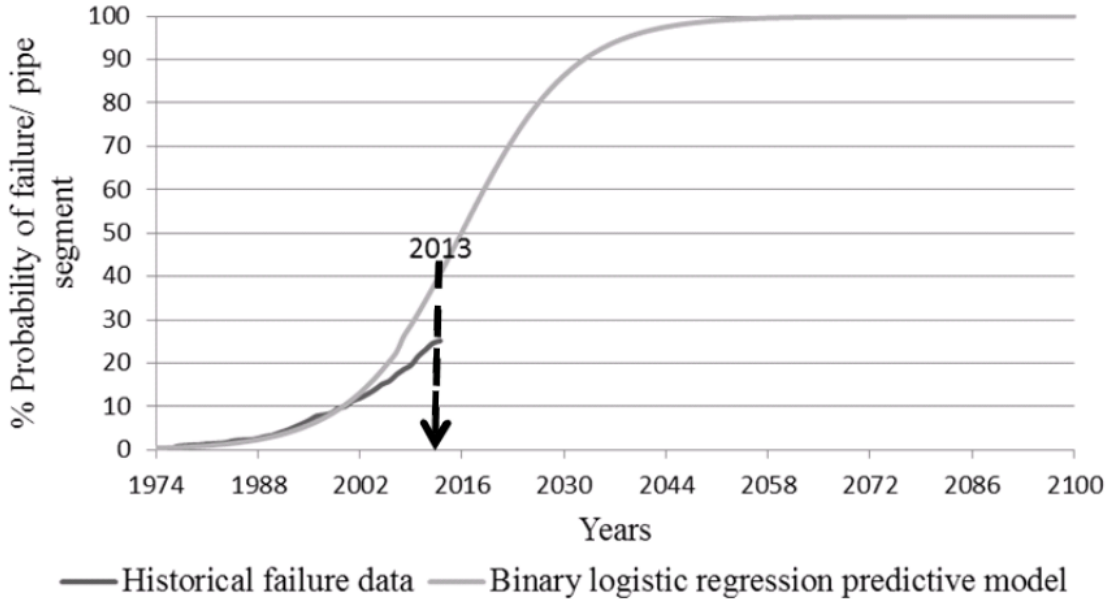


Figure 4. 6", 8" and 12" cast iron pipe (installed between 1900 and 1910) failure model using Binary Logistic Regression

Vladeanu, G. J., & Koo, D. D. (2015). A comparison study of water pipe failure prediction models using Weibull distribution and binary logistic regression. In *Pipelines 2015* (pp. 1590-1601).

Example 2: Logistic regression for infrastructure condition monitoring

Example (multinomial) Logistic regression: evaluation of the local sewer system

- The data comprised of
  - pipe segments/locations
  - length (manhole to manhole)
  - pipe material
  - pipe diameter
  - pipe age (current year minus year of installation)
  - depth (depth of backfill over the crown of pipe in ft)
  - soil conditions
  - Corrosivity
  - Slope
  - surface condition–highway/street
  - condition rating (1-5)

Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

Example of the data

ID	Diameter	Age	Pipe Material	Slope	Surface Condition	Depth	Length	pH	Soil Type	Corrosion Concrete	Corrosion Steel	Condition Rating
2472	12	43	PVC	0.24	Street	15	480.157	6.7	Sand	Low	Moderate	1
1814	10	50	VCP	0.1	Easement	15	421.0372	6.7	Sand	Low	Moderate	1
843	6	97	VCP	0.8	Alley	15	263.5681	6.7	Sand	Low	Moderate	1
2343	8	23	PVC	0.3	Street	15	235.9731	6.7	Sand	Low	Moderate	1
2795	18	50	VCP	0.08	Alley	15	80.58689	6.7	Sand	Low	Moderate	1
65	8	50	VCP	0.3	Street	11	535.9586	6.7	Sand	Low	Moderate	1
623	12	71	CONC	0.6	Highway	10	472.1441	6.7	Sand	Low	Moderate	1
624	24	64	CONC	0.12	Street	10	465.4685	6.7	Sand	Low	Moderate	1
2366	12	51	VCP	0.3	Alley	10	401.3963	6.7	Sand	Low	Moderate	1
3215	8	22	PVC	0.33	Street	10	384.402	6.7	Sand	Low	Moderate	1
3097	12	51	VCP	0.3	Street	10	325.2434	6.7	Sand	Low	Moderate	1
1365	8	24	PVC	0.4	Alley	10	283.7502	6.7	Sand	Low	Moderate	1
3327	48	29	PVC	0.14	Street	10	278.4683	6.7	Sand	Low	Moderate	1
2146	12	39	PVC	2.1	Street	10	159.0316	6.7	Sand	Low	Moderate	1
2295	15	66	VCP	0.32	Street	10	156.1034	6.7	Sand	Low	Moderate	1
285	8	35	PVC	0.8	Easement	10	99.28742	6.7	Sand	Low	Moderate	1
181	10	48	VCP	0.8	Alley	10	70.07311	6.7	Sand	Low	Moderate	1
47	8	16	PVC	0.4	Street	10	24.48685	6.7	Sand	Low	Moderate	1
2428	12	9	PVC	0.2	Street	8	479.9761	6.7	Sand	Low	Moderate	1

Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

Example of a derived equation for condition 1

$$g_1(x) = \ln\left[\frac{Pr(C=1)}{Pr(C=5)}\right]$$
$$= 0.978 * \text{Diameter} + 0.945 * \text{Age} + 1.023 * \text{Slope} + 1.018 * \text{Depth} + 0.999 * \text{Length}$$
$$+ 1.321 * \text{pH} + 1.146 * \text{MaterialCONC} + 1.899 * \text{MaterialPVC} + 0.721 * \text{SurfaceAlley}$$
$$+ 0.771 * \text{SurfaceEasement} + 0.879 * \text{SurfaceHighway} + 0.619 * \text{SoilTypeClay} + 1.037$$
$$* \text{SoilTypeLoam} + 0.942 * \text{SoilTypeRock} + 0.962 * \text{CorrosivityConcreteHigh} + 3.653$$
$$* \text{CorrosivityConcreteLow} + 1.533 * \text{CorrosivitySteelHigh} \quad (16)$$

where:

Pr (C = 1) is the probability of sanitary sewer pipe condition dependent variable being condition 1 relative to condition 5.
Pr (C = 5) is the probability of reference category condition 5.

Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

Probabilities for the sewer pipe conditions (all conditions)

$$\Pr(C = 1|x) = \frac{e^{g_1(x)}}{1 + e^{g_1(x)} + e^{g_2(x)} + e^{g_3(x)+e^{g_4(x)}}}$$

$$\Pr(C = 2|x) = \frac{e^{g_2(x)}}{1 + e^{g_1(x)} + e^{g_2(x)} + e^{g_3(x)+e^{g_4(x)}}}$$

$$\Pr(C = 3|x) = \frac{e^{g_3(x)}}{1 + e^{g_1(x)} + e^{g_2(x)} + e^{g_3(x)+e^{g_4(x)}}}$$

$$\Pr(C = 4|x) = \frac{e^{g_4(x)}}{1 + e^{g_1(x)} + e^{g_2(x)} + e^{g_3(x)+e^{g_4(x)}}}$$

$$\Pr(C = 5|x) = \frac{1}{1 + e^{g_1(x)} + e^{g_2(x)} + e^{g_3(x)+e^{g_4(x)}}}$$

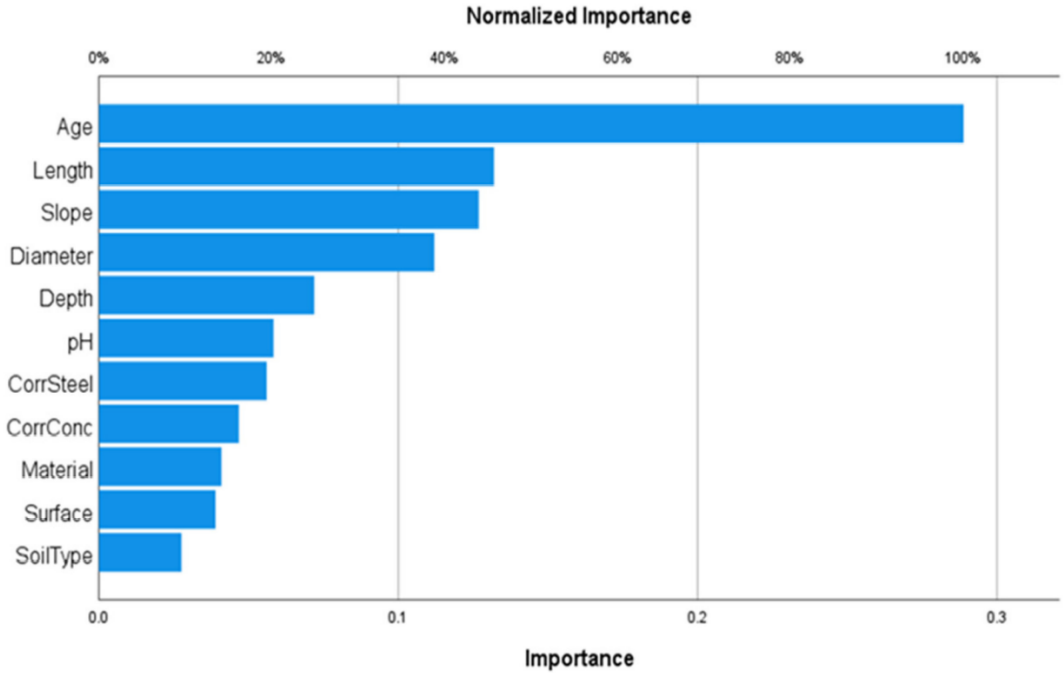
Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

Significant factors

Factors	Sewer Pipe Condition			
	1	2	3	4
Diameter	0.001	0.000	0.199	0.008
Age	0.000	0.001	0.000	0.807
Length	0.000	0.228	0.113	0.980
Material	0.503	0.025	0.001	0.280

Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

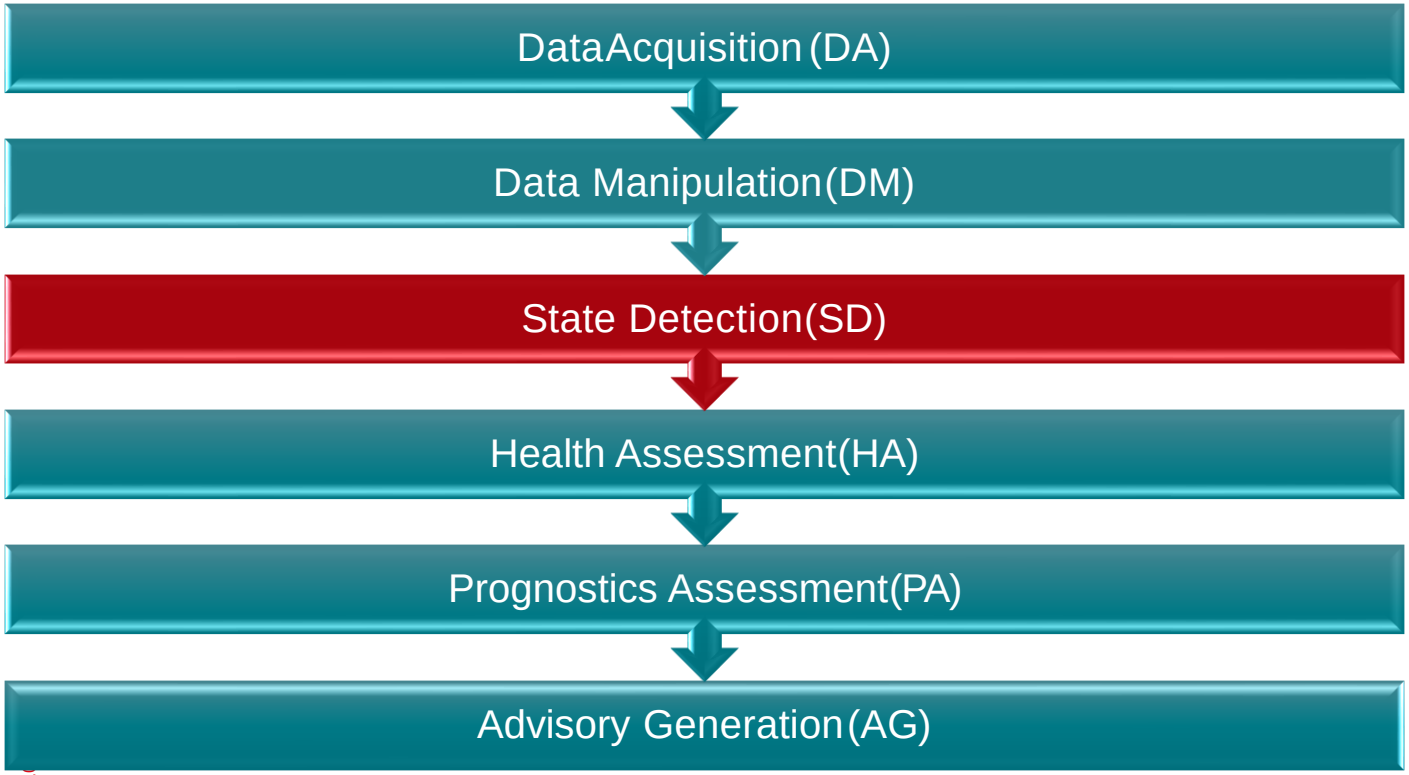
Influence of the variables



Atambo, D. O., Najafi, M., & Kaushal, V. (2022). Development and Comparison of Prediction Models for Sanitary Sewer Pipes Condition Assessment Using Multinomial Logistic Regression and Artificial Neural Network. *Sustainability*, *14*(9), 5549.

Decisio trees

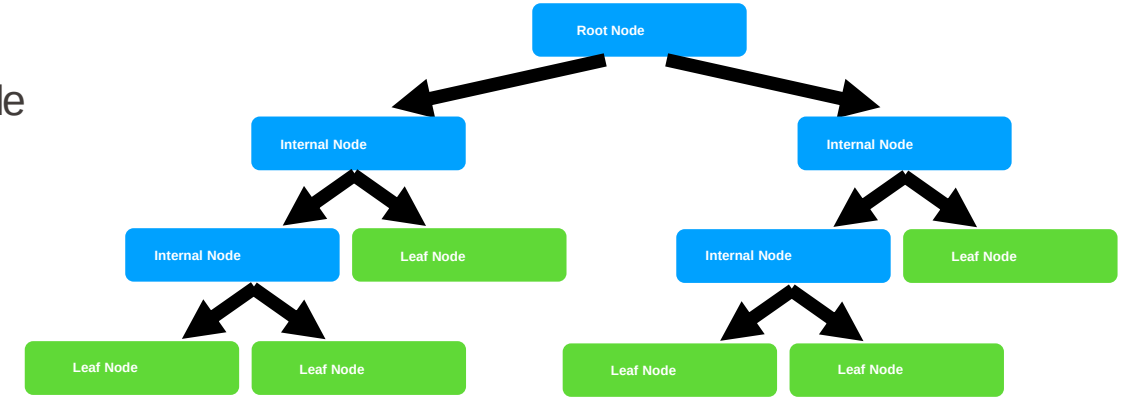
PHM Process


DataAcquisition (DA)
Data Manipulation (DM)
State Detection (SD)
Health Assessment (HA)
Prognostics Assessment (PA)
Advisory Generation (AG)

Decision Trees

Terminology

- Nodes are checked on a single feature
- Branches are feature values
- Leaves indicate class label



Decision Trees

Tree construction

- We select the most discriminative **Feature**
  - Discriminative power based on a score:
    - Information gain
    - Gini impurity
- We create a node based on this feature
- We repeat for each new branch until all the samples are classified

Decision Trees

Information gain

At a given branch in the tree, the set of **samples S** to be classified has **P positive** and **N negative** instances

The **entropy of the set S** is : $H(P,N) = -(\frac{P}{P+N}\log_2(\frac{P}{P+N}) + \frac{N}{P+N}\log_2(\frac{N}{P+N}))$

Note : $H(P,N) = 0$ →No uncertainty ; $H(P,N) = 1$ →Maximal uncertainty

Feature A partitions S into $S_1, S_2, \ldots, S_v$

The **entropy of the feature A** is : $H(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{P+N} H(P_i, N_i)$

The **information gain** obtained by splitting S using A is : $Gain(A) = H(P,N) - H(A)$

Decision Trees

Continuous features

With continuous features, we **cannot have a separate branch for each value**
→ use **binary decision trees**

Binary decision trees :

- For continuous feature A, a split is defined by $val(A) < X$
- For categorical feature A, a split is defined by a subset $X \subseteq domain(A)$

Decision Trees

Characteristics of decision tree induction

Automatic feature selection

Minimal data preparation

Non-linear model

Easy to interpret and explain

Sensitive to small perturbations in the data

Tend to overfit

No incremental updates

Ensemble approaches

Ensemble approaches

- Methods that combine the predictions of multiple models to improve overall accuracy and reduce overfitting
- The idea is to create an ensemble of models that are individually weak but collectively strong

Two common ensemble approaches

- **Bagging**

→short for Bootstrap Aggregating

→machine learning ensemble method

→combines the predictions of multiple models to improve overall accuracy and reduce overfitting.

→works by randomly selecting subsets of the original dataset (with replacement)

→training a separate model on each subset

→then aggregating the predictions of all models to produce a final prediction.

- **Boosting**

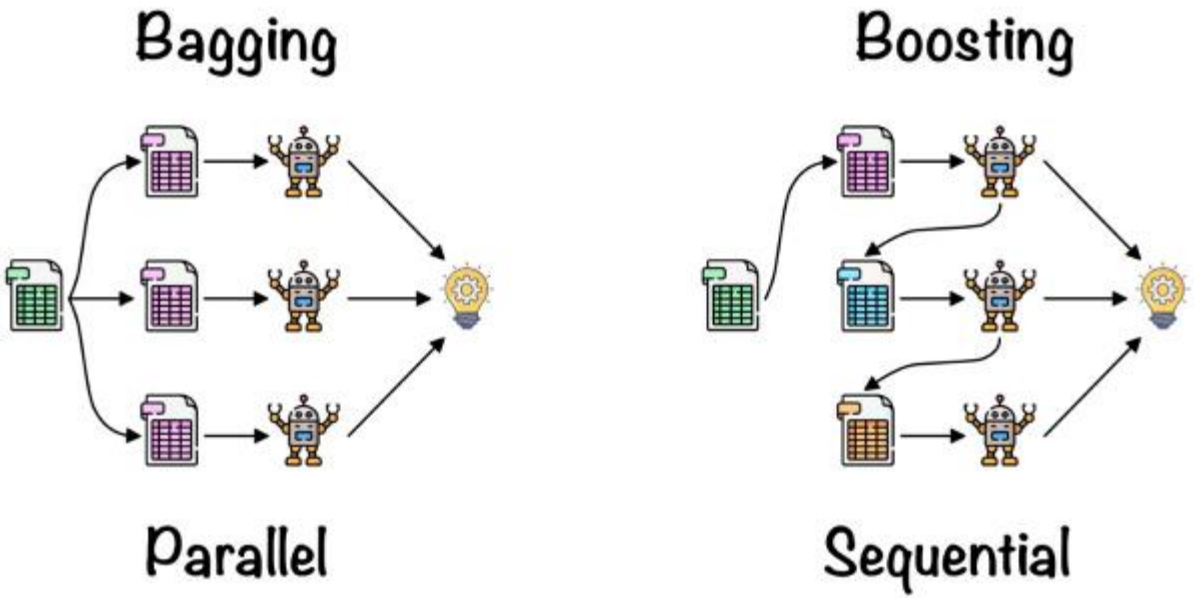
→machine learning ensemble method

→combines weak learners to create a stronger model

→based on the idea of **iteratively** adding weak models to the ensemble, where each **subsequent** model is trained to improve the performance of the previous model.

→a weak learner is a model that performs slightly better than random guessing

Bagging vs. Boosting



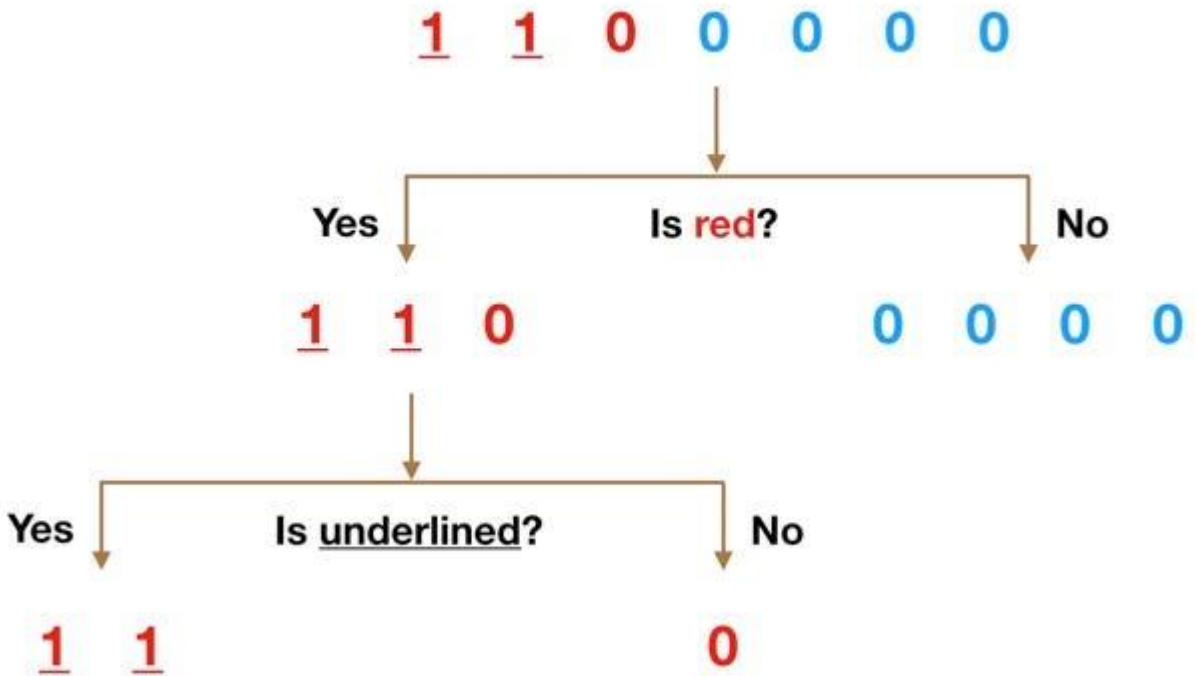
Source: www.towardsdatascience.com

Random forest

Random forest

- Random forest is used for both classification and regression tasks.
- It is an ensemble learning method that combines multiple decision trees to make predictions.
- The name "random forest" comes from the fact that the algorithm creates a "forest" of decision trees that are constructed using a random subset of the training data and a random subset of the features.
- Decision tree:
  - goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features
  - follows a set of if-else conditions to visualize the data and classify it according to the conditions

Example of a decision tree

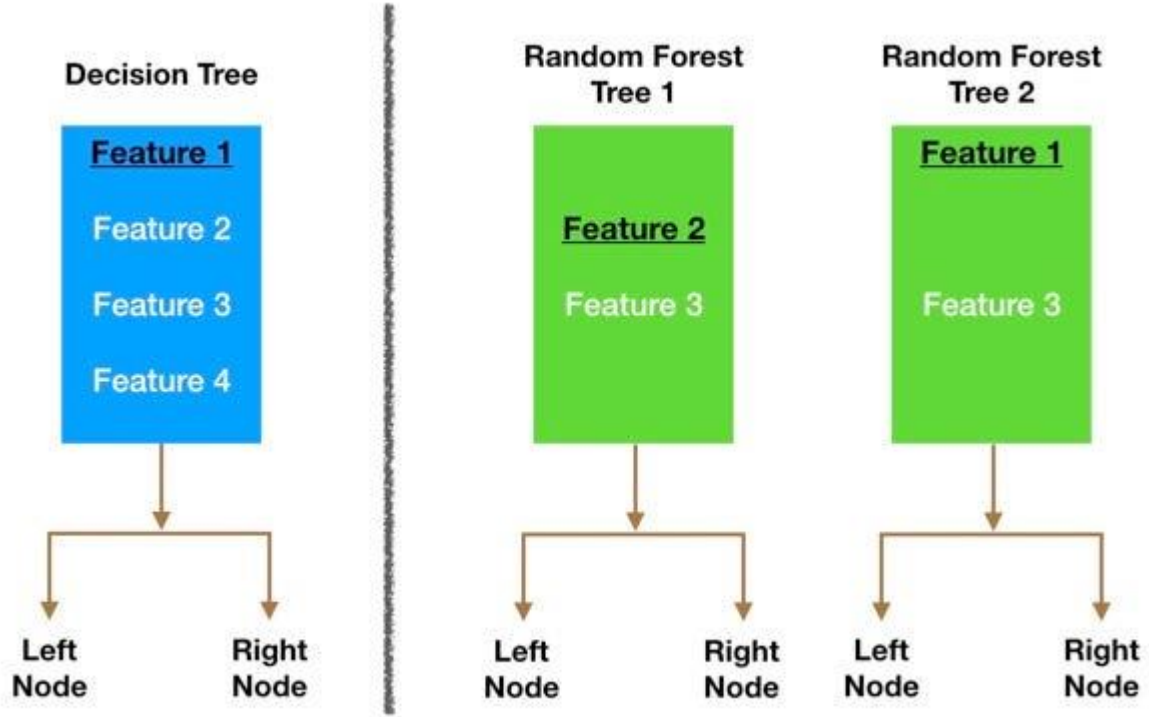


Source: www.towardsdatascience.com

Basics of Random Forest

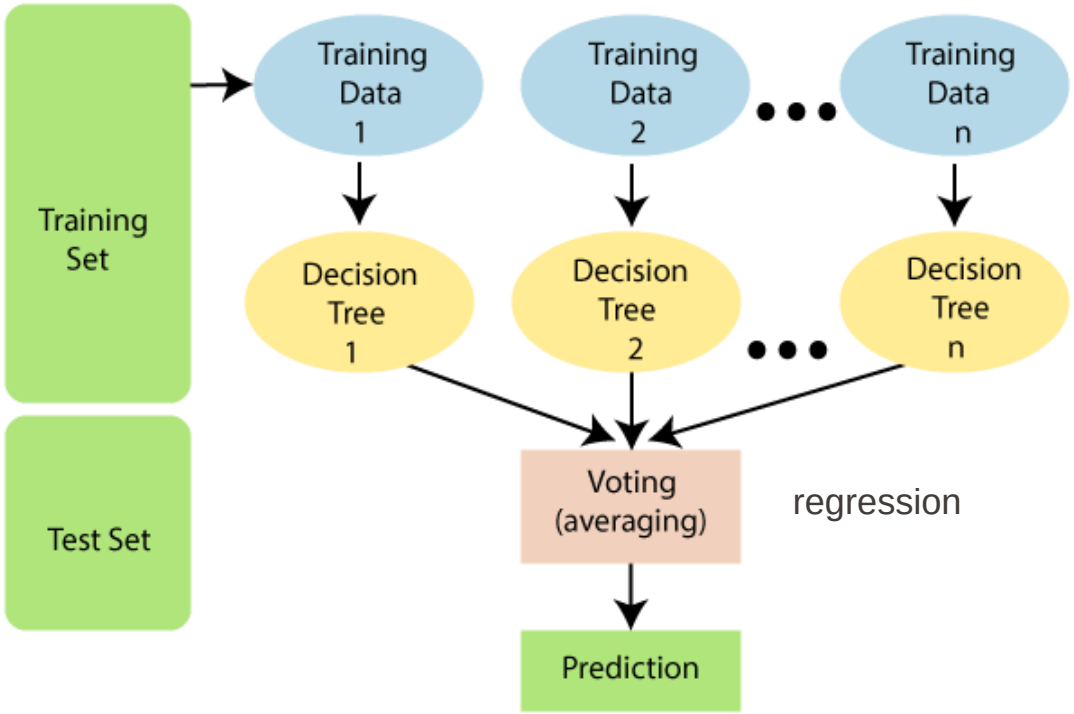
1. Randomly select a subset of the training data.
2. Randomly select a subset of the features.
3. Construct a decision tree using the selected data and features.
4. Repeat steps 1-3 multiple times to create a forest of decision trees.
5. To make a prediction, the algorithm combines the predictions of all the decision trees in the forest. For classification tasks, it uses the majority vote of the trees to determine the predicted class. For regression tasks, it takes the average

Node splitting



Source: www.towardsdatascience.com

Basic principle



regression

Voting (majority)

Classification

Source: www.towardsdatascience.com

Assumptions

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

- The predictions from each tree must have very low correlations.

Why use Random Forest?

- It takes less training time as compared to other algorithms.
- It can predict output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Advantages of random forest

- **Diversity:** Not all attributes/variables/features are considered while making an individual tree; each tree is different.
- **Immune to the curse of dimensionality:** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization:** Each tree is created independently out of different data and attributes.
- **Stability/Robustness:** Stability/Robustness arises because the result is based on majority voting/ averaging.
- **Interpretability:** Easier to interpret the single decision trees.

Example Random Forest for infrastructure condition monitoring

Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild

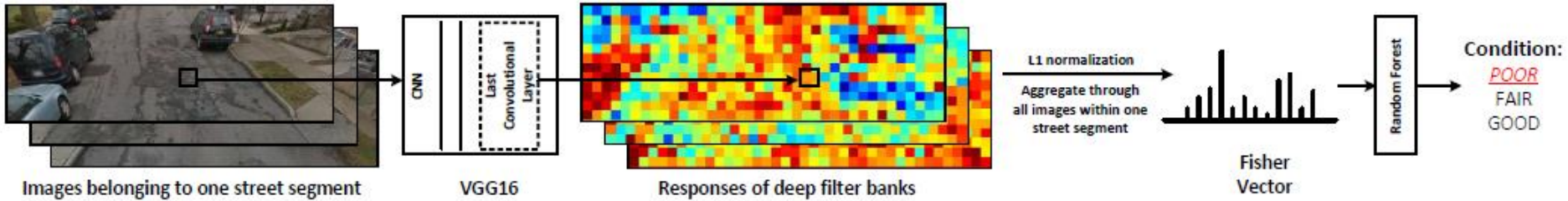
Ma, K., Hoai, M. and Samaras, D., 2017, September. Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. In *BMVC*.

Challenges

- Class imbalance:
  - Only 0.7% of the pavement data is rated poor.
  - fair and good, correspond to 28.8% and 70.5% of the data respectively
- Images are taken under diverse environmental conditions
- images from the same category can look drastically different, depending on the construction materials (e.g., concrete, asphalt, composite) and weather and illumination conditions (e.g., sunny, snow, shadow)
- The estimated time gap between when an image was taken and when it was rated is 1.2 year (estimated on a small subset of the data) → label noise

Ma, K., Hoai, M. and Samaras, D., 2017, September. Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. In *BMVC*.

Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild



Ma, K., Hoai, M. and Samaras, D., 2017, September. Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. In *BMVC*.

Results

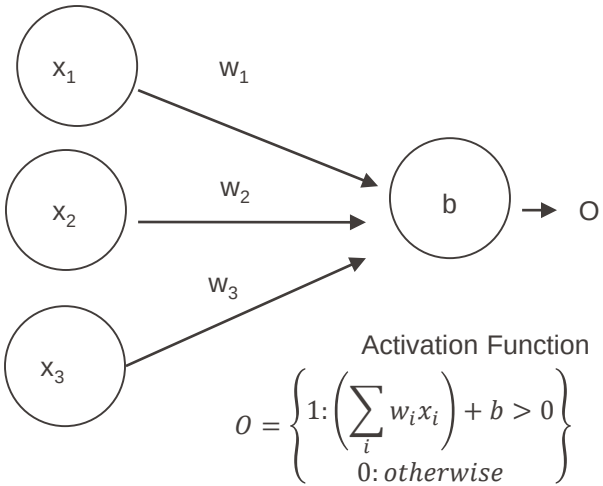
Model	FV-SIFT SVM	FC-CNN SVM	FV-CNN Image SVM	FV-CNN Patch SVM	FV-CNN Patch L1 SVM	FV-CNN Patch L1 RF
POOR	78.0	68.3	1.2	18.5	33.6	72.2
FAIR	35.8	35.2	41.8	36.1	30.6	50.7
GOOD	46.6	42.6	84.4	86.7	85.9	51.7
AVG	53.5	48.7	42.5	47.1	50.0	**58.2**

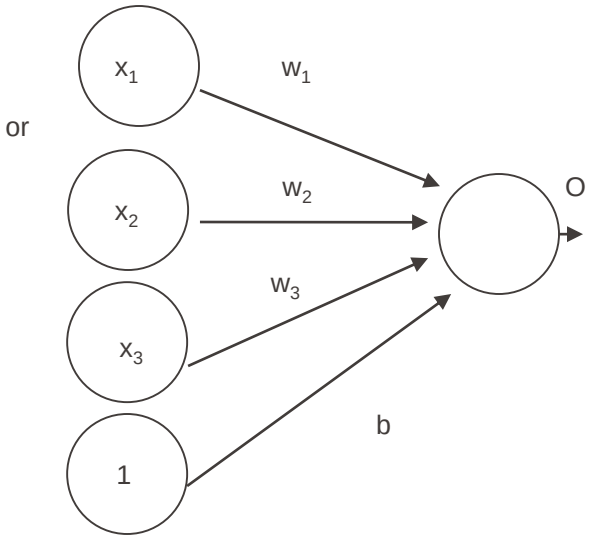
Ma, K., Hoai, M. and Samaras, D., 2017, September. Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. In *BMVC*.

Multi-Layer-Perceptrons Recap

Perceptrons

- Initial proposal of connectionist networks
- Rosenblatt, 50's and 60's
- Essentially a linear discriminant composed of nodes, weights





Linear Perceptrons

They are multivariate linear models:

$$\text{Out}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$$

"training" consists of minimizing sum-of-squared residuals by gradient descent

$$E = \sum_k (\text{Out}(x_k) - y_k)^2$$

$$= \sum_k (w^T x_k - y_k)^2$$

The Perceptron was only capable of handling linearly separable data

Source: Moore, 2003

Linear Perceptron Training Rule

$$\frac{\partial E}{\partial w_j} = \sum_{k=1}^{R} \frac{\partial}{\partial w_j} (y_k - \mathbf{w}^T \mathbf{x}_k)^2$$

$$= \sum_{k=1}^{R} 2(y_k - \mathbf{w}^T \mathbf{x}_k) \frac{\partial}{\partial w_j} (y_k - \mathbf{w}^T \mathbf{x}_k)$$

$$= -2 \sum_{k=1}^{R} \delta_k \frac{\partial}{\partial w_j} \mathbf{w}^T \mathbf{x}_k$$

…where…

$$\delta_k = y_k - \mathbf{w}^T \mathbf{x}_k$$

$$= -2 \sum_{k=1}^{R} \delta_k \frac{\partial}{\partial w_j} \sum_{i=1}^{m} w_i x_{ki}$$

$$= -2 \sum_{k=1}^{R} \delta_k x_{kj}$$

Source: Moore, 2003

Linear Perceptron Training Rule

$$E = \sum_{k=1}^{R} (y_k - \mathbf{w}^T \mathbf{x}_k)^2$$

$$w_j \leftarrow w_j - \eta \frac{\partial E}{\partial w_j}$$

…where…

$$\frac{\partial E}{\partial w_j} = -2 \sum_{k=1}^{R} \delta_k x_{kj}$$

$$w_j \leftarrow w_j + 2\eta \sum_{k=1}^{R} \delta_k x_{kj}$$

η is the Learning Rate → a small positive number, e.g. η = 0.05

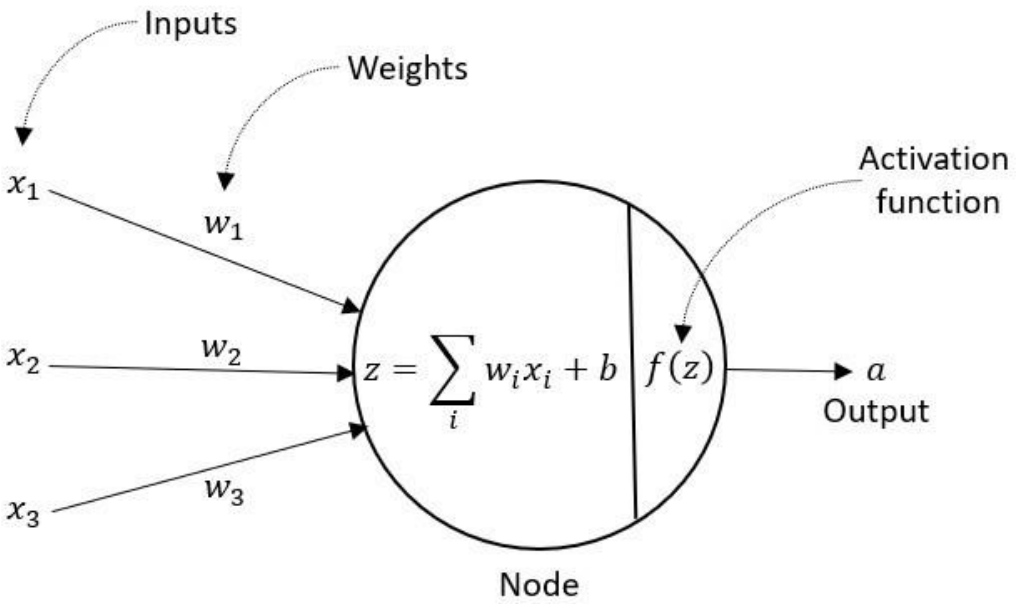
Delta rule

$$\delta_i \leftarrow y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i$$

$$w_j \leftarrow w_j + \eta \delta_i x_{ij}$$

Source: Moore, 2003

Basic principle neurons



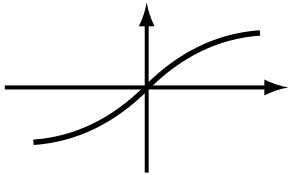
Activation functions

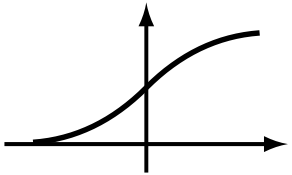
Popular activation functions:

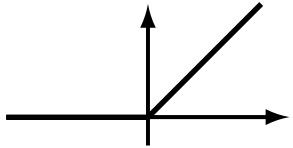
tanh(x)

sigmoid(x) = $\frac{1}{1+e^{-x}}$

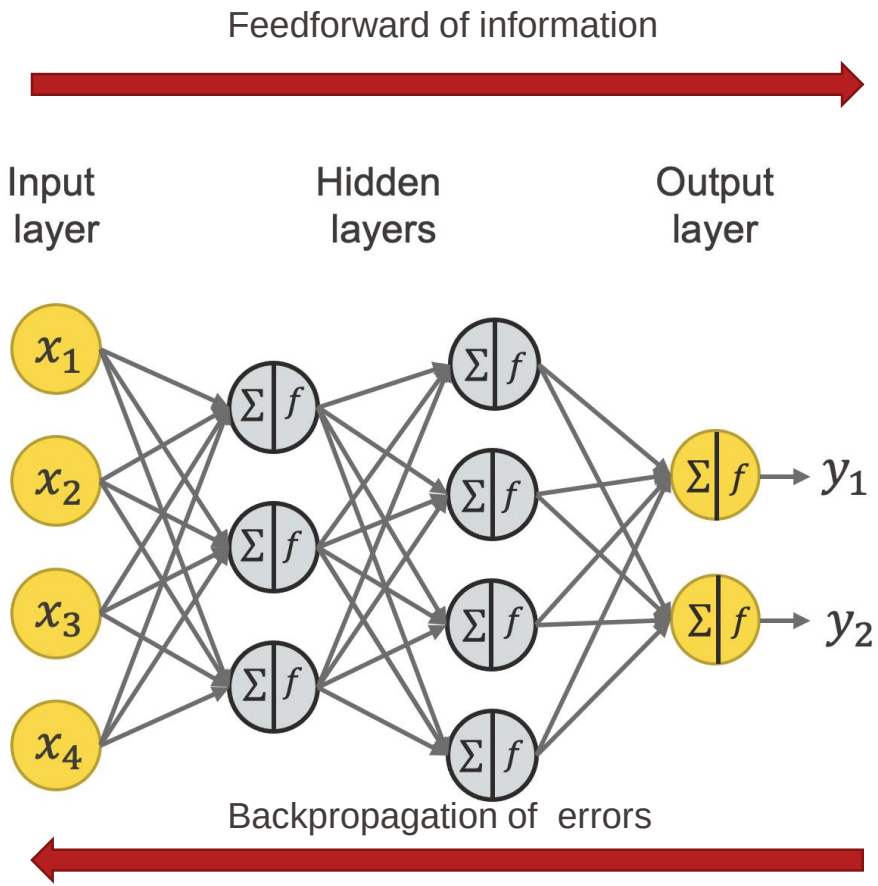
ReLU(x) = max(0,x) → Rectified Linear Unit







Basic principle neural network



Gradient Descent

Suppose we have a scalar function $\mathrm{f}(\mathrm{w}) : \Re \rightarrow \Re$

We want to find a local minimum.
Assume our current weight is *w*

Gradient descent rule: $w \leftarrow w - \eta \frac{\partial}{\partial w} \mathrm{f}(w)$

η is again the Learning Rate

Source: Moore, 2003

Gradient Descent in "m" Dimensions

Given $\mathbf{f(w)} : \Re^m \rightarrow \Re$

$$\nabla f(w) = \begin{pmatrix} \frac{\partial}{\partial w_1} f(w) \\ \vdots \\ \frac{\partial}{\partial w_m} f(w) \end{pmatrix}$$ points in direction of steepest ascent.

$|\nabla f(w)|$ is the gradient in that direction

GRADIENT DESCENT RULE: $w \leftarrow w - \eta \nabla f(w)$

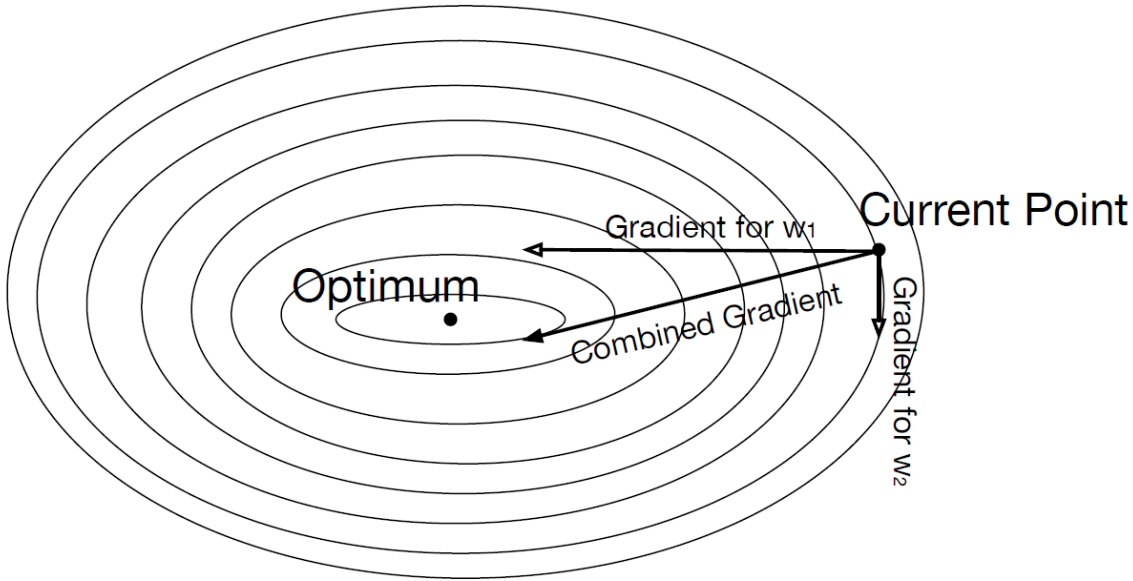
Equivalently $w_j \leftarrow w_j - \eta \frac{\partial}{\partial w_j} f(w)$

….where w_j is the jth weight

"just like a linear feedback system"

Source: Moore, 2003

Gradient Descent



Source: Moore, 2003

Gradient descent with sigmoid on a perceptron

First, notice $g'(x) = g(x)(1-g(x))$

Because: $g(x) = \frac{1}{1+e^{-x}}$ so $g'(x) = \frac{-e^{-x}}{(1+e^{-x})^2}$

$$= \frac{1-1-e^{-x}}{(1+e^{-x})^2} = \frac{1}{(1+e^{-x})^2} - \frac{1}{1+e^{-x}} = \frac{-1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) = -g(x)(1-g(x))$$

$$\text{Out(x)} = g\left(\sum_k w_k x_k\right)$$

$$\text{E} = \sum_i \left(y_i - g\left(\sum_k w_k x_{ik}\right)\right)^2$$

$$\frac{dE}{dw_j} = \frac{dE}{dg}\frac{dg}{ds}\frac{ds}{dw_j}$$

$$\frac{\partial \text{E}}{\partial w_j} = \sum_i 2\left(y_i - g\left(\sum_k w_k x_{ik}\right)\right)\left(-\frac{\partial}{\partial w_j} g\left(\sum_k w_k x_{ik}\right)\right)$$

$$= \sum_i -2\left(y_i - g\left(\sum_k w_k x_{ik}\right)\right) g'\left(\sum_k w_k x_{ik}\right) \frac{\partial}{\partial w_j} \sum_k w_k x_{ik}$$

$$= \sum_i -2\delta_i g(s_i)(1-g(s_i))x_{ij}$$

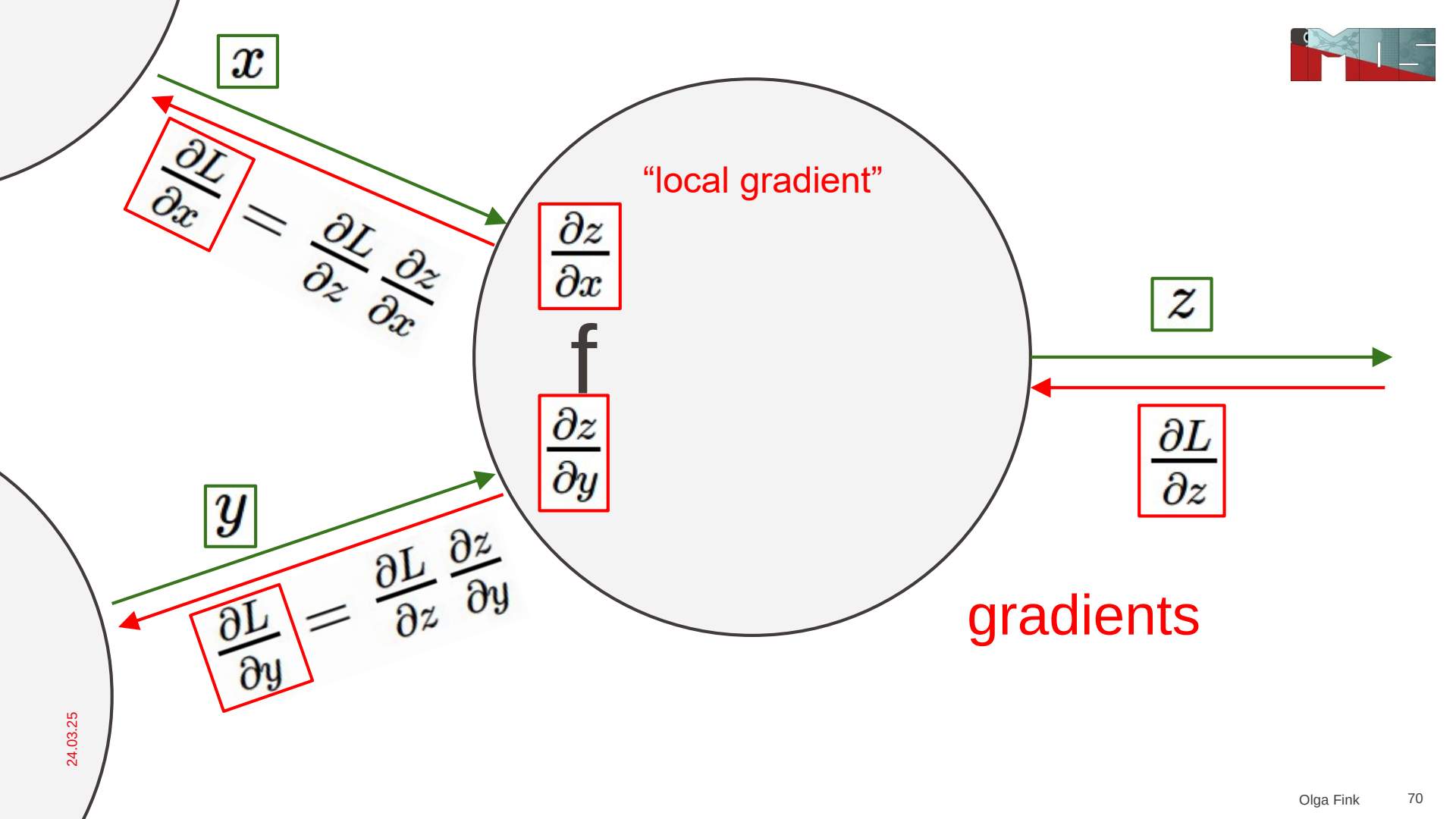
The sigmoid perceptron update rule:

$$w_j \leftarrow w_j + \eta \sum_{i=1}^{R} \delta_i g_i (1-g_i) x_{ij}$$

where $g_i = g\left(\sum_{j=1}^{m} w_j x_{ij}\right)$

$$\delta_i = y_i - g_i$$

Source: Moore, 2003


x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
“local gradient”
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

Improving Simple Gradient Descent

Momentum

Don't just change weights according to the current datapoint.

Re-use changes from earlier iterations.

Let $\Delta\mathbf{w}(t)$ = weight changes at time t.

Let $-\eta \frac{\partial \mathrm{E}}{\partial \mathbf{w}}$ be the change we would make with regular gradient descent.

Instead we use $\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta\mathbf{w}(t)$

$$\Delta\mathbf{w}(t+1) = -\eta \frac{\partial \mathrm{E}}{\partial \mathbf{w}} + \alpha \Delta\mathbf{w}(t)$$

Momentum damps oscillations.

Source: Moore, 2003

Class imbalance

Introduction to Class Imbalance

- **What is Class Imbalance?**: A situation in datasets where classes are not represented equally.
- **Impact on Machine Learning**: Models become biased towards the majority class, potentially compromising accuracy.
- **Significance**: Crucial to address for fair and effective machine learning outcomes in various applications, from finance to healthcare and in particular for infrastructure monitoring.

Imbalance of datasets

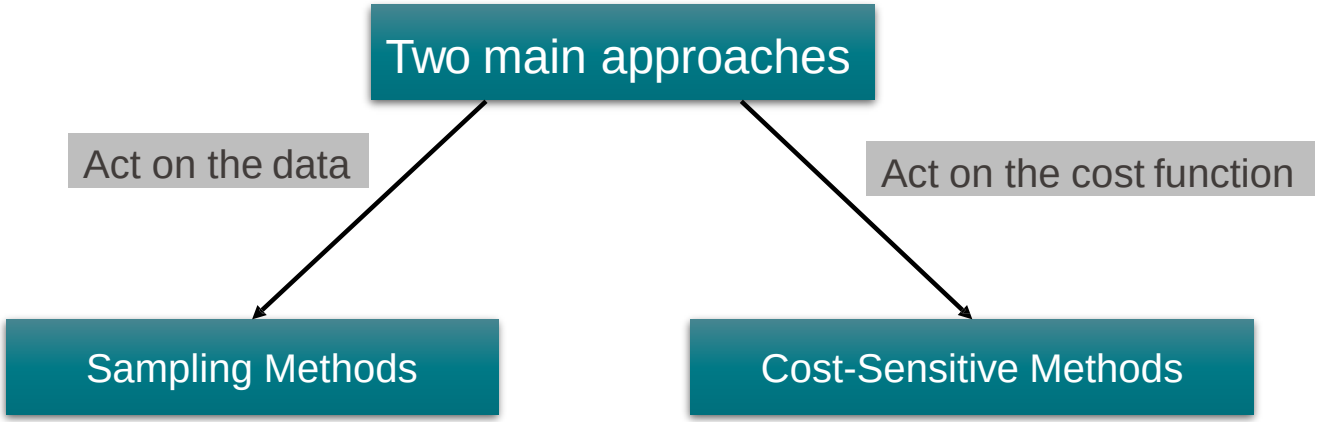
- Between-class
- Within-class
- Intrinsic and extrinsic
- Relativity and rarity
- Imbalance and small sample size

Source: H. He and E. A. Garcia, 2009

Effects of Class Imbalance

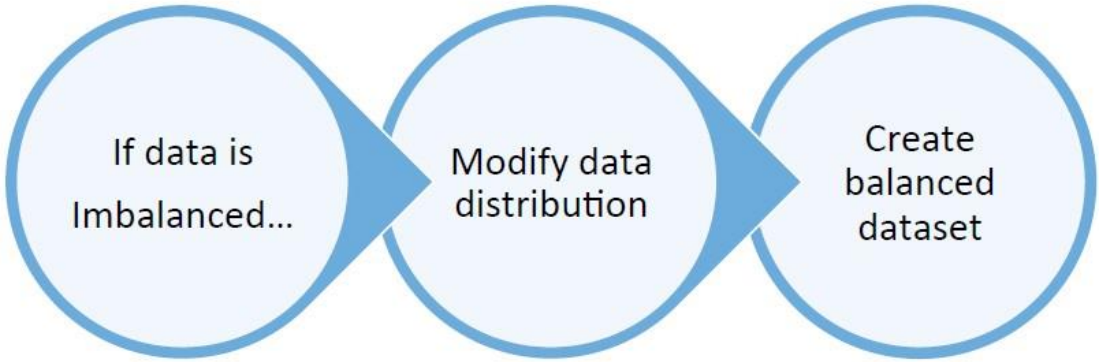
- **Skews Model Performance:** Models might perform well overall while failing on the minority class.
- **Leads to Misleading Accuracy:** High accuracy scores can be deceptive, not reflecting true predictive performance.
- **Compromises Model Generalization:** Models may struggle to generalize to unseen data, especially from the minority class.
- **Affects Model Fairness:** Risks unfair outcomes, particularly in sensitive applications like loan approval or disease screening.
- **Increases False Negatives:** Vital in contexts where missing the minority class (like fraud or disease) is costly.
- **Reduces Recall for Minority Class:** Lower ability to correctly identify all actual positive cases.
- **Encourages Poor Decision-making:** Biased models can lead to decisions that perpetuate existing inequalities.

Learning with imbalanced datasets



Source: H. He and E. A. Garcia, 2009

Sampling methods



Create balance though sampling

Source: H. He and E. A. Garcia, 2009

Sampling Methods

Compensate the lack of data by:

Increase Dataset

Generate new data points for the smallest class

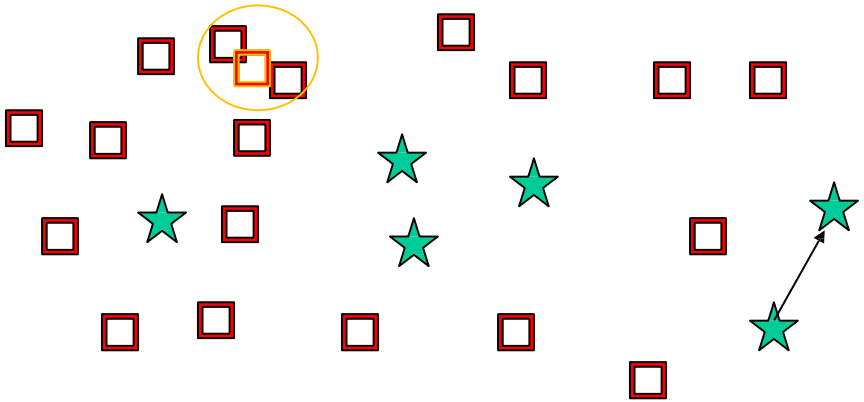
Decrease Dataset

Remove redundant datapoints from the largest class

Source: H. He and E. A. Garcia, 2009

Undersampling

Remove redundant datapoints

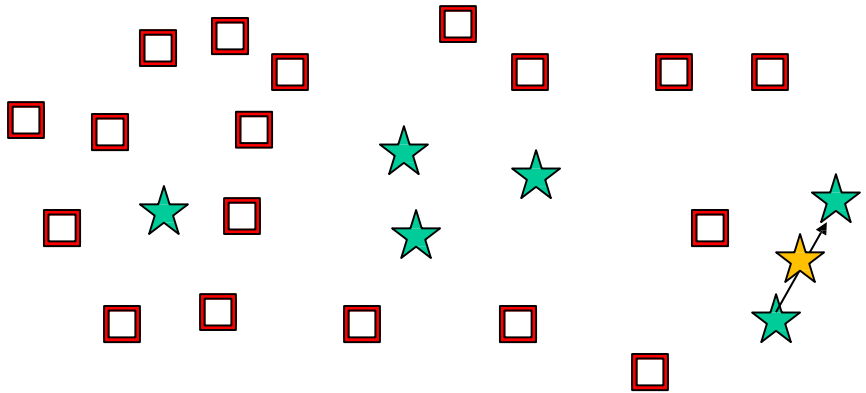


Looses statistics – good only if enough datapoints on undersampled class and for low dimensional datasets

Source: H. He and E. A. Garcia, 2009

Oversampling

Pick neighbour and create new datapoint



Risk overfitting, especially if one does this for points that are noise

Source: H. He and E. A. Garcia, 2009

Random Sampling

S: training data set; S_{min}: set of minority class samples, S_{maj}: set of majority class samples; ***E***: generated samples

Random oversampling

- Expand the minority
- $|S'_{min}| \leftarrow |S_{min}| + |E|$
- $|S'| \leftarrow |S_{min}| + |S_{maj}| + |E|$
- Overfitting due to multiple "tied" instances

Random undersampling

- Shrink the majority
- $|S'_{maj}| \leftarrow |S_{maj}| - |E|$
- $|S'| \leftarrow |S_{min}| + |S_{maj}| - |E|$
- Loss of important concepts

Source: H. He and E. A. Garcia, 2009

Sampling methods: Informed Undersampling

- *EasyEnsemble*
  - ***Unsupervised**: use random subsets of the majority class to create balance and form multiple classifiers*
- *BalanceCascade*
  - ***Supervised**: iteratively create balance and pull out redundant samples in majority class to form a final classifier*

Source: H. He and E. A. Garcia, 2009

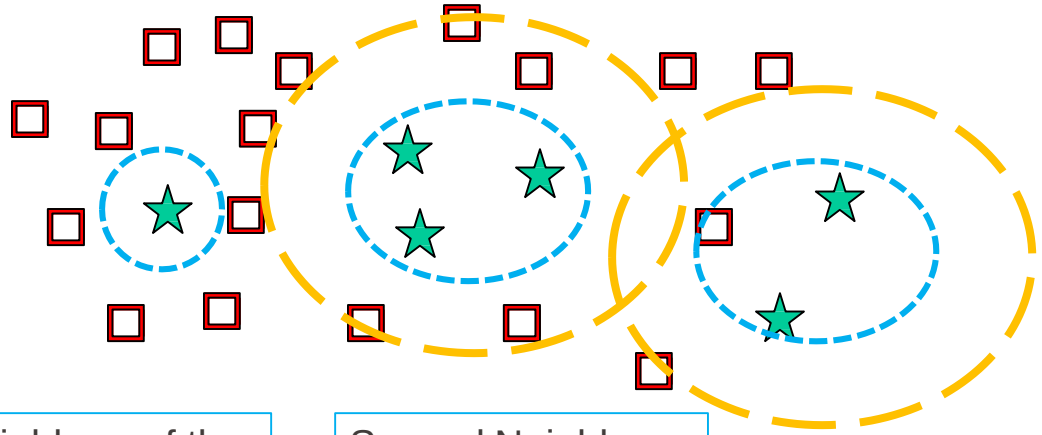
Sampling methods: Informed Undersampling

- **Start with Imbalanced Data**
  - Majority class heavily outweighs the minority class.
- **Compute Informativeness (e.g., Entropy H(n))**
  - Measure how informative each **majority class** sample is.
  - Common metric: **Entropy H(n)** → higher means more informative.
- **Rank Samples**
  - Sort majority samples by informativeness (e.g., H(n)).
- **Select N*maj Samples**
  - Retain the **most informative** samples (top-N or threshold-based).
  - Discard redundant or less informative ones.
- **Create a Balanced Dataset**
  - Combine selected **majority samples** with **minority class** data.
- **Train the Model**
  - Train on the balanced set for **better performance** and **reduced bias**.

Source: H. He and E. A. Garcia, 2009

SMOTE: Synthetic minority oversampling technique

Generate new samples inbetween existing datapoints based on their local density and their borders with the other class. Can use cleaning techniques (undersampling) to remove redundancy in the end.



No Neighbors of the same class → **noise**

Several Neighbors of the same class

Surrounded by the other class → **in danger**

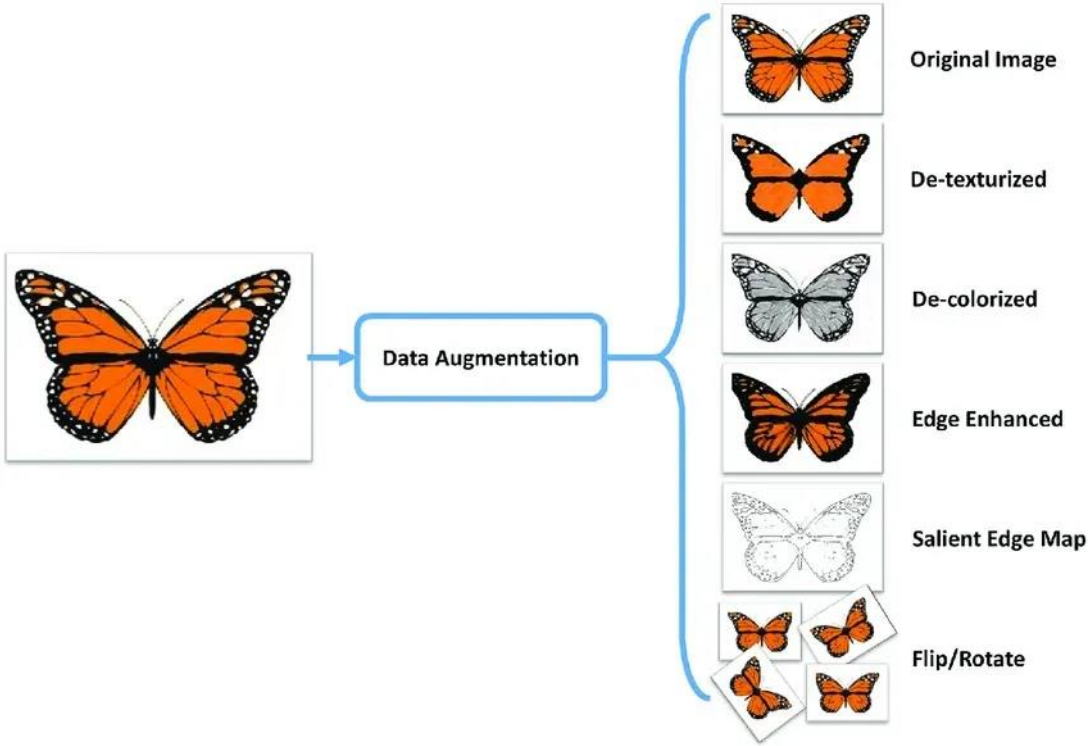
Surrounded only on one side by the other class → **safe**

Source: H. He and E. A. Garcia, 2009

Data augmentation

- **Introduction of Synthetic Samples**: Generating new instances in the minority class using techniques like SMOTE to balance class distribution.
- **Image Manipulation**: For image data, employing rotations, flips, crops, and color variations to create additional examples of the minority class.
- **Interpolation**: Creating synthetic samples by interpolating between existing minority class instances.
- **Noise Injection**: Adding slight variations to data to generate new samples without altering the class meaning.
- **Utilizing Generative models (such as GANs)**: Generating realistic, synthetic data for the minority class using Generative Adversarial Networks.
- **Adaptive Resampling**: Dynamically adjusting augmentation strategies based on the model's performance to better address class imbalance.
- **Evaluation and Adjustment**: Continuously monitoring the impact of augmentation on model performance and adjusting strategies to avoid overfitting.

Data augmentation



Data augmentation in time series

- Jittering, where random noise is added
- Scaling, to adjust the amplitude
- Window slicing, to create sub-sequences
- Time warping, to simulate variations in the speed of time series events;
- Rotation, for multivariate time series to capture different perspectives of the same phenomena.

Cost-sensitive methods

Source: H. He and E. A. Garcia, 2009

Cost-Sensitive Neural Networks

Modifying probability estimate of outputs	Altering outputs directly	Modify learning rate	Replacing error--minimizing function
• *Applied only at testing stage* • *Maintain original neural networks*	• *Bias neural networks during training to focus on expensive class*	• *Set η higher for costly examples and lower for low--cost examples*	• *Use expected cost minimization function instead*

Source: H. He and E. A. Garcia, 2009

Naïve Bayes

Probability Basics

- Prior, conditional and joint probability for random variables
  - Prior probability: $P(x)$
  - Conditional probability: $P(x_1 \mid x_2), P(x_2 \mid x_1)$
  - Joint probability: $\mathbf{x} = (x_1, x_2), P(\mathbf{x}) = P(x_1, x_2)$
  - Relationship: $P(x_1, x_2) = P(x_2 \mid x_1)P(x_1) = P(x_1 \mid x_2)P(x_2)$
  - Independence: $P(x_2 \mid x_1) = P(x_2), P(x_1 \mid x_2) = P(x_1), P(x_1, x_2) = P(x_1)P(x_2)$

- $$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c)P(c)}{P(\mathbf{x})}$$

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Discriminative

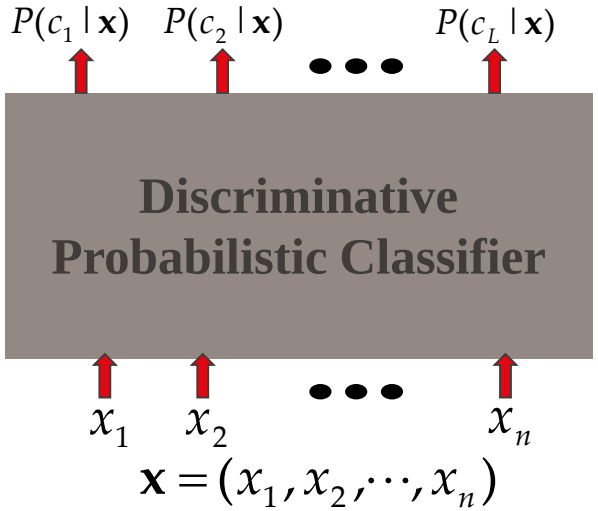
Generative

Source: Ke Chen, 2011

Probabilistic Classification Principle

- Establishing a probabilistic model for classification
  - **Discriminative model**

$$P(c \mid \mathbf{x}) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$



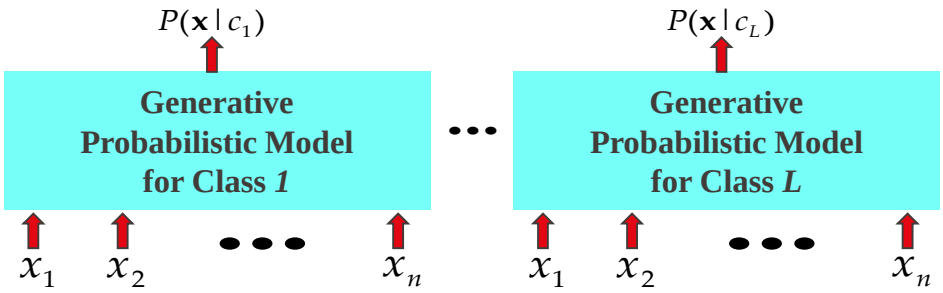
- To train a discriminative classifier (regardless its probabilistic or non-probabilistic nature), all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output *L* probabilities for *L* class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic discriminative classifier .

Source: Ke Chen, 2011

Probabilistic Classification Principle

- Establishing a probabilistic model for classification (cont.)
  - **Generative model (must be probabilistic)**

$$P(\mathbf{x} \mid c) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

- *L* probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output *L* probabilities for a given input with *L* models
- "Generative" means that such a model can produce data subject to the distribution via sampling.

Source: Ke Chen, 2011

Probabilistic Classification Principle

- **M**aximum **A P**osterior (**MAP**) classification rule
  - For an input $\boldsymbol{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier $P(c_1 \mid \mathbf{x}), \ldots, P(c_L \mid \mathbf{x})$.
  - Assign $\boldsymbol{x}$ to label c^* if $P(c^* \mid \mathbf{x})$ is the largest.
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i)P(c_i)$$

$$\text{for } i = 1, 2, \cdots, L$$

Common factor for all L probabilities

  - Then apply the MAP rule to assign a label

Source: Ke Chen, 2011

Naïve Bayes

- Bayes classification

$$P(c \mid \mathbf{x}) \propto P(\mathbf{x} \mid c)P(c) = P(x_1, \cdots, x_n \mid c)P(c) \text{ for } c = c_1, ..., c_L.$$

Difficulty: learning the joint probability $P(x_1, \cdots, x_n \mid c)$ is often infeasible!

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

$$\begin{aligned} P(x_1, x_2, \cdots, x_n \mid c) &= P(x_1 \mid x_2, \cdots, x_n, c)P(x_2, \cdots, x_n \mid c) \\ &= P(x_1 \mid c)P(x_2, \cdots, x_n \mid c) \\ &= P(x_1 \mid c)P(x_2 \mid c) \cdots P(x_n \mid c) \end{aligned}$$

Applying the independence assumption

  - Apply the MAP classification rule: assign $\mathbf{x}' = (a_1, a_2, \cdots, a_n)$ to c^* **if**

$$[P(a_1 \mid c^*) \cdots P(a_n \mid c^*)]P(c^*) > [P(a_1 \mid c) \cdots P(a_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

estimate of $P(a_1, \cdots, a_n \mid c^*)$ — esitmate of $P(a_1, \cdots, a_n \mid c)$

Source: Ke Chen, 2011

Naïve Bayes

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set **S** of F features and L classes,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;

For every feature value x_{jk} of each feature x_j $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

$\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in S;

Output: $F * L$ conditional probabilistic (generative) models

  - Test Phase: Given an unknown instance $\mathbf{x}' = (a_1', \cdots, a_n')$

"Look up tables" to assign the label c^* to **X'** if

$$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_n' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c_i) \cdots \hat{P}(a_n' \mid c_i)]\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \cdots, c_L$$

Source: Ke Chen, 2011

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Source: Ke Chen, 2011

Example

- Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

Source: Ke Chen, 2011

Example

- Test Phase
  - Given a new instance, predict its label

    X'=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play=*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

  - Decision making with the MAP rule

P(*Yes*|**X**') ≈ [P(*Sunny*|*Yes*)P(*Cool*|*Yes*)P(*High*|*Yes*)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053
P(*No*|**X**') ≈ [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|**X**') < P(*No*|**X**'), we label **X**' to be "*No*".

Source: Ke Chen, 2011

Example SHM

- Data Assumption
- Assume we have historical data that indicate the following probabilities:
- **Prior Probabilities of Each Condition:**
  - *P*(Good)=0.70
  - *P*(Minor Damage)=0.20
  - *P*(Major Damage)=0.10
- **Likelihood of Vibration Frequency Deviations (Hz):**
  - Good Condition: Normally around 0.5 Hz deviation.
  - Minor Damage: Deviations around 2 Hz.
  - Major Damage: Deviations exceed 4 Hz.
- **Likelihood of Maximum Daily Temperature Variation (°C):**
  - Good Condition: Variation within ±5°C.
  - Minor Damage: Variation within ±10°C.
  - Major Damage: Variation exceeds ±15°C.

Example SHM

- For simplicity, let's assume we categorize the deviations into "low," "medium," and "high" for both features and assign probabilities based on our historical data.
- Scenario:
- One day, sensors on the bridge report a vibration frequency deviation of 3 Hz and a maximum daily temperature variation of 12°C. We need to classify the bridge's condition based on this data.

Example SHM cont.

- Let's simplify and assume:
- The probability of observing a 3 Hz deviation is:
  - P(3 Hz|Good)=0.1
  - P(3 Hz|Minor Damage)=0.7
  - P(3 Hz|Major Damage)=0.2
- The probability of observing a 12°C variation is:
  - P(12°C|Good)=0.05
  - P(12°C|Minor Damage)=0.6
  - P(12°C|Major Damage)=0.35
- Calculating Probabilities
- To classify the bridge's condition, we calculate the posterior probability for each condition using Bayes' theorem, focusing on the product of the likelihood and the prior probability for simplicity.
- Let's calculate these probabilities.
- Based on the calculated probabilities, the bridge's health condition is classified as follows:
  - Good: 3.7%
  - Minor Damage: 88.9%
  - Major Damage: 7.4%

Naïve Bayes

- Algorithm: Continuous-valued Features
  - Numberless values taken by a continuous-valued feature
  - Conditional probability is often modelled with the normal distribution

$$\hat{P}(x_j \mid c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji} : standard deviation of feature values x_j of examples for which $c = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_F),\ C = c_1, \cdots, c_L$

    Output: $F \times L$ normal distributions and $P(C = c_i)\ \ i = 1, \cdots, L$
  - Test Phase: Given an unknown instance $\mathbf{X}' = (a_1', \cdots, a_n')$
    - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
    - Apply the MAP rule to assign a label (the same as done for the discrete case)

Source: Ke Chen, 2011

Naïve Bayes

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

    Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

    No: 27.3, 30.1, 17.4, 29.5, 15.1
  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \ \sigma_{No} = 7.09$$

  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

Source: Ke Chen, 2011

Summary

- Probabilistic Classification Principle
  - Discriminative vs. Generative models: learning P(c|x) vs. P(x|c)P(c)
  - Generative models for classification: MAP and Bayesian rule
- Naïve Bayes: the conditional independence assumption
  - Training and test are very efficient.
  - Two different data types lead to two different learning algorithms.
- Naïve Bayes: a popular generative model for classification
  - Performance competitive to many state-of-the-art classifiers even in the presence of violating the conditional independence assumption
  - Many successful applications, e.g., spam mail filtering, …
  - A good candidate of a base learner in ensemble learning

Source: Ke Chen, 2011