Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P237-cd

Two Algorithms for Defect Detection in Wafer Fabrication

Thomas Olschewski

Institute of Mathematical Stochastics, Technische Universität Dresden, Germany.
E-mail: thomas.olschewski@tu-dresden.de

Suitability of two algorithms for learning chip defect detection based on high-dimensional measurement data from wafer fabrication is examined, some results from applying them to real-world chip data are reported and a selection of mathematical properties of the indicator used in one of the algorithms is presented. In a number of series of experiments and parameter studies with different product types, the algorithms turned out to be effective in detecting the binary overall defect state of measurement steps for which measurement data is available, and in reducing input data dimensionality and/or sample count.

Keywords: Wafer fabrication, data analysis, defect detection, dimensional reduction, high-dimensional, mode analysis.

1. Introduction and Related Work

Methods of Machine learning (ML) and artificial intelligence, being a research field in rapid development, have started to become omnipresent in data mining tasks and got boosted in popularity by multiple layer feedforward artificial neural nets (Rumelhart et al. (1986)) and deep learning, recurrent neural networks and others, enabling new applications like autonomous driving or chatbots recently—Sarker (2021), Mandic and Chambers (2001), Kiran et al. (2022), Gao et al. (2022).

Starting with perceptrons (Rosenblatt (1962) and Minsky and Papert (1969)), a wide variety of ML methods has been developed for different purposes since the mid of the 20th century. Among these there are outlier detection methods (Domingues et al. (2018), Olschewski et al. (2020)) based on angle analysis (Kriegel et al. (2008), Pham and Pagh (2012)), Isolation Forest (Liu et al. (2008)), rapid distance-based outlier detection (Sugiyama and Borgwardt (2013)), SVM (support vector machines) by Cortes and Vapnik (1995) and Boser et al. (1992), and the methods of Sumikawa et al. (2013) for wafer classification, just to name some few examples. Other ML approaches include stochastic learning of disjunctive normal forms (Valiant (1984) and Valiant (1985)). See Angluin and Laird (1988), Kearns and Li (1993) and Ben-David et al. (2003) for results on the complexity of constructing binary raters using single monomials.

Recent multinational research projects like iRel4.0 (iRel4.0 (2020)) and Productive 4.0 (Productive4.0 (2017)) emphasize the importance of improving product reliability in wafer fabrication. In this paper, we examine the suitability of two algorithms for learning defect detection in high-dimensional chip data, one (Section 2) based on a fractional integer indicator, one (Section 3) relying on detecting certain specifics of the feature distributions, and two algorithms for dimensional reduction (Section 2.1 and description in Section 3). The data material is composed of analog (voltages, currents ...) and digital (count- and flag-register contents ...) values from chips on semiconductor wafers. See Baker (2010) for fundamentals. Both classifiers aim at restricting optimization steps to low-dimensional search spaces for the sake of improving explainability of results (Barredo Arrieta et al. (2020), Samek et al. (2021), Zeiler and Fergus (2014)), finding global optima and reducing the number of internal constants to be specified in the presence of small numbers of samples (Rumelhart et al. (1986)).

1.1. *Formal setting*

We consider some lot of m chips, each represented by n measurements, as a matrix $X \in \mathbb{R}^{m\times n}$ where missing values in some column are replaced

by the lower median of all non-missing values in this column. The type and sequence of measurements is the same for all chips.

For both of the two algorithms, we apply column-wise auto-scaling to X as preprocessing: $x^*_{i,j} = \frac{x_{i,j}-\mu_j}{\sigma_j}$ (or 0 if $\sigma_j = 0$) where μ_j and σ_j are the mean and standard deviation of the j-th column, respectively.

Set $\mathbb{B} = \{0,1\}$. We assume one bit $v_i \in \mathbb{B}$ being assigned to every chip represented by the i-th row x_i of matrix X, which induces a partitioning $\{1,\dots,m\} = I^- \sqcup I^+$. In our terms, a positive chip ($v_i = 1$) is always defective. Let $H : \mathbb{B}^n \to \mathbb{N}_{\geqq 0}$ be the Hamming weight.

1.2. *The tasks*

The types of tasks we want to solve are formalized as follows (Olschewski et al. (2020), Olschewski (2021b) and Olschewski (2021a)). Given is a lot consisting of m chips with n measurements $x_{i,j}$ and valuation v_i each, represented by matrix $X \in \mathbb{R}^{m\times n}$ and $v = (v_1,\dots,v_m) \in \mathbb{B}^m$. We will write "chip i" for the i-th chip $x_i = (x_{i,1},\dots,x_{i,n})$.

> *Given the $x_{i,j}$ and v_i for chips i in a small training set $I \subset \{1,\dots,m\}$, predict the valuation $v_k \in \mathbb{B}$ of the remaining chips $k \in \{1,\dots,m\} \setminus I$ of the lot, based on their measurements $x_k = (x_{k,1},\dots,x_{k,n})$.*

The chip measurements are partitioned into different measurement steps such as S1, S2, S3 in course of wafer fabrication. Determinants for the difficulty level of the task include: (i) product type, m, n, (ii) measurement steps from which data is available, (iii) size of sample sets, and (iv) type of sampling: by random or prescribed by former needle card insertions.

2. Algorithm Using Fractional Indicator

See Algorithm 1. Let $\theta = \theta_t : \mathbb{R} \to \{0,1\}$ be some thresholding function like $\chi_{[t,\infty)}$ and I^-, I^+ as in Section 1.1. c can be set by optimizing for Cohen's kappa (for example) of the two 0-1 vectors $(v_1,\dots,v_m)$ and $\big(P(1),\dots,P(m)\big)$, skipping indices in T.

Input: $X \in \mathbb{R}^{m\times n}$ auto-scaled by column
Input: I^-, I^+ with
$I^- \sqcup I^+ = \{1,\dots,m\}$,
$\square \in \{\min, \max, \text{mean}, ...\}$
Input: $\theta_t : \mathbb{R} \to \mathbb{B}$, threshold $t > 0$
Input: cutoff $c > 0$
Output: $T \subset I^+$, $z_\square(i), P(i)$ for $i \in \{1,\dots,m\} \setminus T$
for $i \in \{1,\dots,m\}$ **do**
  for $j \in \{1,\dots,n\}$ **do**
    $x^*_{i,j} := \theta_t(x_{i,j})$
  end
end
Select some training set $T \subset I^+$ randomly
for $i \in \{1,\dots,m\} \setminus T$ **do**
  $x^*_i, x^*_k :=$ column i, k of X^*
  $z_\square(i) := \square\left\{\frac{\langle x^*_i, x^*_k\rangle}{H(x^*_i)} \mid k \in T\right\}$
end
for $i \in \{1,\dots,m\} \setminus T$ **do**
  $P(i) := \begin{cases} 1, & z_\square(i) \geqq c \\ 0, & z_\square(i) < c \end{cases}$
end

Algorithm 1: Z$(X, I^-, I^+, \square, \theta_t, t, c)$

2.1. *Dimensional reduction*

Function dim-reduce(X, s) (Algorithm 2) keeps only those columns j^* of X in which there are at least s positive chips i satisfying $|x_{i,j^*}| = \max\{|x_{i,1}|,\dots,|x_{i,n}|\}$. Reducing n translates into reducing the measurement count directly.

2.2. *Results*

2.2.1. *Detecting Iris type*

When applied to the classic Iris flower data set (Fisher (1936), Dua and Graff (2017)), $z_{\max}$ of Algorithm 1 classifies setosa, versicolor and virginica with kappa values 0.928, 0.557 and 0.797, respectively, using 20% to 30% training set size.

2.2.2. *Detecting chip defects*

In this series of classifications by Algorithm 1, the influence of dimensional reduction by Algorithm

Input: $X =\in \mathbb{R}^{m\times n}$ auto-scaled by column, $\{1,\ldots,m\} = I^- \sqcup I^+$
Input: sharpness $s \in \mathbb{N}_{\geqq 0}$
Output: $E \in \mathbb{R}^{m\times n^*}$, n^*
for $i \in I^+$ **do**
  $M := \max\{|x_{ij}| : j \in \{1,\ldots,n\}\}$
  $\text{MaxIndices}_i := \{j \in \{1,\ldots,n\} : |x_{ij}| = M\}$
end
for $j = 1\ldots n$ **do**
  $\text{NumOccu}_j := \sum_{i\in I^+} |\text{MaxIndices}_i \cap \{j\}|$
end
$k := 0$
for $j = 1\ldots n$ **do**
  if $|\text{NumOccu}_j| \geqq s$ **then**
    $k := k + 1$
    for $i = 1\ldots m$ **do**
      $e_{i,k} := x_{i,j}$
    end
  end
end
$n^* := k$

Algorithm 2: dim-reduce(X, sharpness)

2 as a preprocessing step on the results is examined. 34550 chips (one lot) with 150 measurements per chip of product D had to be classified by computing $z_{\min}$ in one run for S2≠"0" with 10% training set $|T|$ and thresholding function $\theta_t(x) = \chi_{[t,\infty)}$ with $t = 0.1$.

As can be seen in Table 1, 115 (76.7%) of the 150 features can be omitted by Algorithm 2 with only a small decrease in classification quality: kappa 0.837 instead of 0.870 and $\frac{\text{TP}}{\text{FP}} > 20$ instead of $\frac{\text{TP}}{\text{FP}} = \frac{1086}{0} = +\infty$.

Figures 1, 2 and 3 belong to the "Sharpness 10" line of Table 1: $z_{\min}$ indicator over chip number—all positive objects relocated to the left for better visibility—, indicator histograms on all (all positive, all negative, resp.) objects, kappa value of the prediction in dependency of the $z_{\min}$ cutoff c. See Olschewski (2021b) for more results by Algorithm 1.

Table 1. Dimensional reduction in classifying S2≠"0" for product D.

Sharpness	#Feat. omit.	%Feat. omit.	Accu %	Kappa
0	0	0	0.991	0.870
1	92	61.3	0.991	0.866
2	102	68.0	0.987	0.821
3	103	68.7	0.986	0.800
4	103	68.7	0.986	0.800
5	107	71.3	0.984	0.776
10	115	76.7	0.989	0.837
20	126	84.0	0.981	0.712
25	133	88.7	0.964	0.377
30	135	90.0	0.959	0.223
40	138	92.0	0.959	0.040

3. Algorithm Matching Modes

See Algorithm 3. In computing histograms, if multiple intervals have maximum frequency, then we use the interval with the lowest index for

Fig. 1. S2≠"0" for product D with dimred sharpness 10.

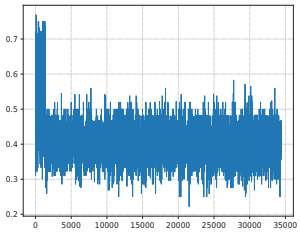
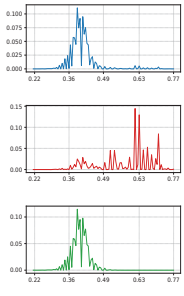


Fig. 2. $z_{\min}$ histogram for S2≠"0" of product D with dimred sharpness 10: all (all positive, all negative) objects.



$I^{(j)}$. Sorting $\{1,\ldots,n\}$ by decreasing $n_{\text{diff}}(j)$ values assigns every j its rank $r(j)$ where $r(j) = 1$ means highest n_{diff} value. Given some $t \in \{1,\ldots,n\}$, we call the t columns with highest ranks "Cics" (candidate indicator columns).

Reducing the n features to those occurring in C as in Algorithm 3 works as a dimensional reduction.

3.1. *Results*

3.1.1. *Detecting Iris type*

Algorithm 3 with $\mathsf{nb} = 5$ or 6, using 60%/1% positive/negative training sets and columns #3 and #4 as Cics classified the classic Iris flower data set (Fisher (1936), Dua and Graff (2017)) with kappa values 0.937 (setosa), 0.727 (versicolor) and 0.756 (virginica).

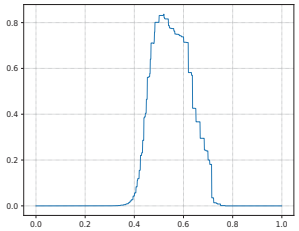
3.1.2. *Detecting S3 fails with feature reduction*

Algorithm 3 has also been used for S3 feature reduction in finding those S3 fail chips which are neither S1- nor S2-fails. One data lot of product G consists of 1661 measurements from 6412 chips. Table 2 lists some results. Kappa values can be improved further by leaving out data of all chips already classified as S1 or S2 fails.

In mode `[S]` (or `[A]`), only the sample chips (or all chips) are used for deriving the Cics. See also Figures 4 and 5.

See Olschewski (2021a) for results of different tasks by Algorithm 3.

Fig. 3. S2≠"0" for product D with dimred sharpness 10: kappa value over cutoff.



Input: $X \in \mathbb{R}^{m\times n}$ column-wise auto-scaled, $I^+ \sqcup I^- = \{1,\ldots,m\}$
Input: sample sets $T^+ \subseteq I^+, T^- \subseteq I^-$
Input: t = #Cics to be used, cutoff $c > 0$, $\mathsf{nb} \in \mathbb{N}_{\geqq 3}$
Output: $P(i)$ for $i \in \{1,\ldots,m\} \setminus (T^+ \cup T^-)$

for $j = 1 \ldots n$ **do**
  Compute histogram (nb bins) of j-th column, limited to rows $i \in T^+$
  $I^{(j)} :=$ most frequent interval
  $n_{\text{pos}}(j) := |\{i \in T^+ : x_{i,j} \in I^{(j)}\}|$
  $n_{\text{neg}}(j) := |\{i \in T^- : x_{i,j} \in I^{(j)}\}|$
  $n_{\text{diff}}(j) := n_{\text{pos}}(j) - n_{\text{neg}}(j)$
end
$(j_1, j_2, \ldots, j_n) :=$ unique permutation of $\{1,2,\ldots,n\}$ satisfying:
$n_{\text{diff}}(j_1) \geqq n_{\text{diff}}(j_2) \geqq \cdots \geqq n_{\text{diff}}(j_n)$
$\wedge\, \forall k \in \{1,\ldots,n-1\}$:
$[n_{\text{diff}}(j_k) = n_{\text{diff}}(j_{k+1}) \Rightarrow j_k < j_{k+1}]$
$\mathsf{C} := \{j_1,\ldots,j_t\}$
for $i \in \{1,\ldots,m\} \setminus (T^+ \cup T^-)$ **do**
  $\mathsf{S_C}(i) := \left|\{j \in \mathsf{C} : x_{i,j} \in I^{(j)}\}\right|$
  $P(i) := \begin{cases} 1, & \mathsf{S_C}(i) \geqq c \\ 0, & \mathsf{S_C}(i) < c \end{cases}$
end

Algorithm 3: MatchMode($X, I^+, I^-, T^+, T^-, t, c, \mathsf{nb}$)

Table 2. Finding true S3 fails while reducing feature count.

Mode	#Feat. used	Train Pos%	Train Neg%	Accu %	Kappa	#Samples Pos/Neg
`[S]`	top:950	50	50	98.4	0.579	75/3131
`[S]`	83	75	0.01	99.7	0.653	113/1
`[S]`	63	50	0.01	99.3	0.595	75/1
`[S]`	43	50	50	95.2	0.024	75/3131
`[A]`	top:950	50	50	99.9	0.986	75/3131

4. Some Stochastical Analysis of $z_\square$ in Algorithm 1

Assume $n, r \in \mathbb{N}_{\geqq 1}$, $\mathbb{B}^{n*} = \mathbb{B}^n \setminus \{0^n\}$, $\langle a, b\rangle = \sum_{i=1}^{r} a_i b_i$ and $H(a) = \sum_{i=1}^{r} a_i$. Let $\mathsf{x} \in \mathbb{B}^n$ be

a random vector of independent, identically distributed (i.i.d.) coordinates $\mathsf{x}_1, \ldots, \mathsf{x}_n$ and $p = \Pr[\mathsf{x}_i = 1]$. Set $X_i = \frac{\mathsf{x}_i}{H(\mathsf{x})}$ if $\mathsf{x} \in \mathbb{B}^{n*}$ and $X_i = 0$ if $\mathsf{x} = 0^n$ $(i = 1, \ldots, n)$. Set $e_1(n,p) = \frac{1-(1-p)^n}{n}$ (if clear from context: e_1). Clearly, $\mathrm{E}[X_i] = e_1(n,p)$.

For every fixed $\mathsf{y} \in \mathbb{B}^n$, set $Z_\mathsf{y} = \frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})}$ (or 0) if $\mathsf{x} \in \mathbb{B}^{n*}$ (or $\mathsf{x} = 0^n$). Then $Z_\mathsf{y} = \sum_{\substack{i=1 \\ \mathsf{y}_i=1}}^{n} X_i$ and $\mathrm{E}[Z_\mathsf{y}] = H(\mathsf{y})e_1(n,p)$.

Definition 4.1.

$$e_2(n,p) = \frac{1}{n}\sum_{k=1}^{n} \frac{(1-p)^{n-k} - (1-p)^n}{k}$$

(if clear from context: e_2).

Theorem 4.1. $\mathrm{E}\left[X_i^2\right] = e_2(n,p)\ \forall i,\ p \in (0,1)$.

Proof. Let $b(x,y) = \sum_{k=1}^{n}\binom{n}{k}\frac{1}{k}x^k y^{n-k}$ for $x, y \geqq 0$. Then by linearity of $\int_0^x \cdots$ and by continuity at its lower limit: $b(x,y) = \sum_{k=1}^{n}\binom{n}{k} y^{n-k} \cdot \int_0^x t^{k-1}\mathrm{d}t = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} \int_\varepsilon^x \frac{(t+y)^n - y^n}{t}\,\mathrm{d}t$. Now assume $x, y > 0$. Let $u(t) = 1 + \frac{t}{y}$. Then $b(x,y) = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} \int_\varepsilon^x \frac{\left(y\cdot u(t)\right)^n - y^n}{y\cdot\left(u(t)-1\right)} \cdot y \cdot u'(t)\,\mathrm{d}t = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} \int_{u(\varepsilon)}^{u(x)} \frac{(y\cdot w)^n - y^n}{y\cdot(w-1)} \cdot y\,\mathrm{d}w = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} y^n \cdot \int_{1+\frac{\varepsilon}{y}}^{1+\frac{x}{y}} \frac{w^n-1}{w-1}\,\mathrm{d}w$. Note that $u(t) - 1 \neq 0$ in the denominator when t varies from ε to $x > 0$. Now set $x = p > 0$ and $y = 1 - p > 0$. Then $1 + \frac{x}{y} = \frac{1}{1-p}$. Thus, by swapping lim and finite sum, $b(p, 1-p) = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} (1-p)^n \cdot \int_{1+\frac{\varepsilon}{1-p}}^{\frac{1}{1-p}} \frac{w^n-1}{w-1}\,\mathrm{d}w = (1-p)^n \cdot \sum_{k=0}^{n-1} \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} \int_{1+\frac{\varepsilon}{1-p}}^{\frac{1}{1-p}} w^k\,\mathrm{d}w = (1-p)^n \cdot \sum_{k=0}^{n-1} \lim_{\substack{\varepsilon\to 0 \\ \varepsilon>0}} \frac{1}{k+1} \cdot \left[\left(\frac{1}{1-p}\right)^{k+1} - \left(1+\frac{\varepsilon}{1-p}\right)^{k+1}\right]$. By continuity, $\frac{b(p,1-p)}{n} = \frac{(1-p)^n}{n} \sum_{k=1}^{n} \frac{1}{k} \cdot \left[\left(\frac{1}{1-p}\right)^k - 1\right] = e_2(n,p)$ for $p \in (0,1)$. But $\mathrm{E}\left[X_i^2\right] = \frac{b(p,1-p)}{n}$ for $n \geqq 1$, because $\mathrm{E}\left[X_i^2\right] = 0 \cdot \Pr[\mathsf{x}_i = 0] + \sum_{k=1}^{n} \frac{1}{k^2} \Pr[\mathsf{x}_i = 1 \wedge H(\mathsf{x}) = k] = \sum_{k=1}^{n} \frac{1}{k^2} \cdot p \cdot \binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} = \frac{1}{n} \cdot \sum_{k=1}^{n}\binom{n}{k}\frac{1}{k} p^k (1-p)^{n-k} = \frac{b(p,1-p)}{n}$. Case $\mathsf{x} = 0^n$ is covered by the summand $0 \cdot \Pr[\mathsf{x}_i = 0]$. Together, $\mathrm{E}\left[X_i^2\right] = \frac{b(p,1-p)}{n} = e_2(n,p)$. □

Then $\mathrm{Var}[X_i] = \mathrm{E}\left[X_i^2\right] - \mathrm{E}[X_i]^2 = e_2 - e_1^2$. A similar calculation shows:

Lemma 4.1. $\mathrm{E}[X_i X_j] = \frac{e_1 - e_2}{n-1}$ *(*$i \neq j$, $n \geqq 2$*) and* $\mathrm{Cov}[X_i, X_j] = \frac{e_1 - e_2}{n-1} - e_1^2$.

For $\mathsf{y} \in \mathbb{B}^{n*}$ with $H(\mathsf{y}) = h$, $n \geqq 2$ and $p \in (0,1)$:

$$\mathrm{E}[Z_\mathsf{y}] = \mathrm{E}[X_1 + \cdots + X_h] = h \cdot e_1 \quad (1)$$

$$\mathrm{Var}[Z_\mathsf{y}] = e_1 \tfrac{h(h-1)}{n-1} + e_2 \tfrac{h(n-h)}{n-1} - e_1^2 h^2 \quad (2)$$

By abbreviating $E_h = \mathrm{E}[X_1 + \cdots + X_h]$ and $V_h = \mathrm{Var}[X_1 + \cdots + X_h]$, $\frac{\mathrm{Var}[Z_\mathsf{y}]}{\mathrm{E}[Z_\mathsf{y}]}$ can be expressed in an especially regular form as a convex combination: $\frac{\mathrm{Var}[Z_\mathsf{y}]}{\mathrm{E}[Z_\mathsf{y}]} = \lambda \cdot \frac{V_n}{E_n} + (1-\lambda)\cdot\frac{V_1}{E_1}$ with $\lambda = \frac{h-1}{n-1} \in [0,1]$.

Fig. 4. Finding true S3 fails with 950 Cics for product G: accuracy and kappa value over cutoff.

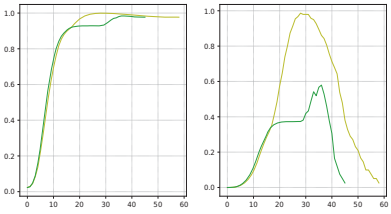
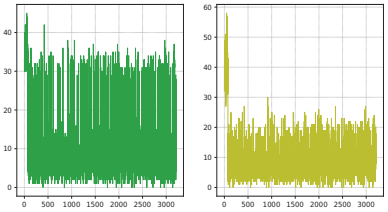


Fig. 5. Finding true S3 fails with 950 Cics for product G: number of satisfied conditions when deriving Cics from sample chips only or from all chips.



4.1. *The variant X_i^* of X_i*

For fixed $\mathsf{y} \in \mathbb{B}^n$, define $Z_\mathsf{y}^* = \frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})}$ (or $\frac{H(\mathsf{y})}{n}$) if $\mathsf{x} \in \mathbb{B}^{n*}$ (or $\mathsf{x} = 0^n$). Then several of the above formulae become simpler:

Lemma 4.2.

(i) $\mathrm{E}\left[Z^*_{\mathsf{y}}\right] = \frac{H(\mathsf{y})}{n}$ *and*
(ii) $\mathrm{Var}\left[Z^*_{\mathsf{y}}\right] = (e^*_2 - {e^*_1}^2) \cdot \frac{H(\mathsf{y})(n-H(\mathsf{y}))}{n-1}$,

where $e^*_1 = e^*_1(n,p) = \frac{1}{n}$ *and* $e^*_2 = e^*_2(n,p) = \frac{1}{n^2} + \frac{1}{n} \cdot \sum_{k=1}^{n-1} \frac{(1-p)^{n-k} - (1-p)^n}{k}$.

4.2. *More on distributional properties with fixed* y

In this Section—as in Section 4.1—the distribution of $\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})}$ with fixed $\mathsf{y} \in \mathbb{B}^n$ will be examined. In terms of Algorithm 1, $\frac{\langle x^*_i, x^*_k\rangle}{H(x^*_i)}$ is computed for some fixed training chip $x^*_k \in \mathbb{B}^n$, while the chip under test $x^*_i \in \mathbb{B}^n$ is varying.

Let $\mathrm{hyp}(k;n,h,\ell) = \frac{\binom{h}{k}\binom{n-h}{\ell-k}}{\binom{n}{\ell}}$ be the probability of k successes when drawing without replacement ℓ objects from a population of n objects, h of which are of a special type (hypergeometric distribution). If $p = \Pr[\mathsf{x}_i = 1]$ is the same for all i, then for unrestricted $H(\mathsf{x})$, $\Pr[\langle \mathsf{x},\mathsf{y}\rangle = k] = \mathrm{bin}(k; H(\mathsf{y}), p)$. If $H(\mathsf{x})$ is fixed, then

$$\begin{aligned}&\Pr[\langle \mathsf{x},\mathsf{y}\rangle = k \wedge H(\mathsf{x}) = \ell]\\ &= \Pr[\langle \mathsf{x},\mathsf{y}\rangle = k \mid H(\mathsf{x}) = \ell] \Pr[H(\mathsf{x}) = \ell]\\ &= \mathrm{hyp}(k;n,H(\mathsf{y}),\ell)\,\mathrm{bin}(\ell;n,p),\end{aligned} \tag{3}$$

and by some short calculation,

$$\begin{aligned}&\mathrm{hyp}(k;n,h,\ell)\,\mathrm{bin}(\ell;n,p)\\ &\quad = \mathrm{bin}(k;h,p)\,\mathrm{bin}(\ell-k;n-h,p).\end{aligned} \tag{4}$$

Remark 4.1. Under suitable conditions on the integers a,b,c,h,n and by setting $\Pi = \frac{h}{n}$, $\mu = b\cdot\Pi$, $\sigma^2 = b\cdot\Pi(1-\Pi)$, $\mathrm{hyp}(ac;n,h,bc) \approx \frac{1}{\sqrt{c}}\frac{1}{\sqrt{2\pi}\sigma}\left(\sqrt{2\pi}\sigma\,\mathrm{hyp}(a;n,h,b)\right)^c$, and for $c|a \wedge c|b$: $\mathrm{hyp}\left(\frac{a}{c};n,h,\frac{b}{c}\right) \approx \frac{\sqrt{c}}{\sqrt{2\pi}\sigma}\left(\sqrt{2\pi}\sigma\,\mathrm{hyp}(a;n,h,b)\right)^{\frac{1}{c}}$.

Proof. Let $\varphi(x) = \frac{1}{\sqrt{2\pi}}\,e^{-\frac{x^2}{2}}$. Approximating the hypergeometric distribution twice by Gauss gives $\mathrm{hyp}\left(\frac{a}{c};n,h,\frac{b}{c}\right)$

$$\begin{aligned}&\approx \frac{1}{\sqrt{2\pi\,\frac{b}{c}\,\Pi(1-\Pi)}}\,e^{-\frac{(\frac{a}{c}-\frac{b}{c}\cdot\Pi)^2}{2\cdot\frac{b}{c}\cdot\Pi(1-\Pi)}}\\ &= \sqrt{c}\left(\frac{1}{\sqrt{2\pi}\,\sigma}\right)^{1-\frac{1}{c}}\left(\frac{1}{\sigma}\,\varphi\left(\frac{a-\mu}{\sigma}\right)\right)^{\frac{1}{c}}\\ &\approx \sqrt{c}\left(\frac{1}{\sqrt{2\pi}\,\sigma}\right)^{1-\frac{1}{c}}\mathrm{hyp}(a;n,h,b)^{\frac{1}{c}},\end{aligned}$$

which is the second claim. Substituting $\frac{1}{c}$ by c proves the claim for $\mathrm{hyp}(ac;n,h,bc)$ for suitable $c \in \mathbb{N}_{\geqq 1}$. □

Theorem 4.2. *Let* $\mathsf{y} \in \mathbb{B}^n$ *be fixed and* $\mathsf{x} \in \mathbb{B}^n$ *random with* $p = \Pr[\mathsf{x}_i = 1]$ *for all* i. *Then for* $a,b \neq 0$, $\Pr\left[\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})} = \frac{a}{b}\right] = \Pr\left[\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})} = \frac{a'}{b'}\right]$ *with* $a' = \frac{a}{\gcd(a,b)}$, $b' = \frac{b}{\gcd(a,b)}$, *and this equals* $\sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} \mathrm{hyp}(a'c;n,h,b'c)\cdot\mathrm{bin}(b'c;n,p)$.

Proof.
From $a,b \neq 0$, $\left(\langle \mathsf{x},\mathsf{y}\rangle, H(\mathsf{x})\right) \in \{1,\ldots,n\} \times \{1,\ldots,n\}$ and $\langle \mathsf{x},\mathsf{y}\rangle \leqq H(\mathsf{x})$ follows: $\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})} = \frac{a}{b}$ iff $\left(\langle \mathsf{x},\mathsf{y}\rangle, H(\mathsf{x})\right) = (a'c, b'c)$ with some $c \in \{1,\ldots,\lfloor \frac{n}{b'}\rfloor\}$. Thus, $\Pr\left[\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})} = \frac{a}{b}\right] = \sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} \Pr[\langle \mathsf{x},\mathsf{y}\rangle = a'c \wedge H(\mathsf{x}) = b'c]$
$= \sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} \mathrm{hyp}(a'c;n,H(\mathsf{y}),b'c)\,\mathrm{bin}(b'c;n,p)$. □

Remark 4.2. Giving the probabilities of $\langle \mathsf{x},\mathsf{y}\rangle = 0$ or $H(\mathsf{x}) = 0$ for random $\mathsf{x} \in \mathbb{B}^n$ takes a special treatment.

Corollary 4.1. *Let* $h = H(\mathsf{y})$. *The following approximation* P *to* $\Pr\left[\frac{\langle \mathsf{x},\mathsf{y}\rangle}{H(\mathsf{x})} = \frac{a}{b}\right]$ *can be derived under the assumptions of Theorem 4.2:* $\mathsf{P} = \frac{e^{\frac{-np}{2(1-p)}}}{2\pi\cdot p(1-p)\sqrt{h(n-h)}} \sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} e^{c^2\,\frac{-a'^2n-b'^2h+2a'b'h}{2p(1-p)h(n-h)} + c\,\frac{2b'hp(n-h)}{2p(1-p)h(n-h)}}$.

Proof. Let $\varphi(x) = \frac{1}{\sqrt{2\pi}}\cdot e^{-\frac{x^2}{2}}$. By Theorem 4.2 and Eq. (4),

$$\mathsf{P} = \sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} \mathrm{bin}(a'c;h,p)\cdot\mathrm{bin}(b'c-a'c;n-h,p),$$

which can be approximated by:
$\sum_{c=1}^{\lfloor \frac{n}{b'}\rfloor} \frac{1}{\sqrt{2\pi hp(1-p)}}\,\varphi\left(\frac{a'c-hp}{\sqrt{hp(1-p)}}\right)$.

$\cdot \frac{1}{\sqrt{2\pi(n-h)p(1-p)}}\ \varphi\left(\frac{(b'-a')c-(n-h)p}{\sqrt{(n-h)p(1-p)}}\right)$. Expression P follows by elementary transformations. □

Fig. 6 visualizes $\Pr\left[\frac{\langle \mathrm{x},\mathrm{y}\rangle}{H(\mathrm{x})} = q\right]$ with $n = 50$, $H(\mathrm{y}) = 40$ and $p = 0.5$, from left to right: frequencies and cumulative of a stochastic simulation with 10^6 repetitions, the distribution by Theorem 4.2 and the approximated distribution by Corollary 4.1.

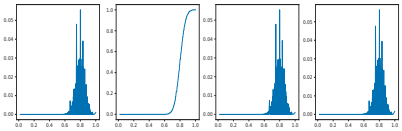
4.3. *Feature-specific distributions*

If $\Pr[\mathrm{x}_j = 1]$ is not the same for all j when applying the same thresholding to all features (as always assumed above), then Algorithm 1 can still be applied and Section 4 is still valid by introducing coordinate-specific thresholds $t_1, \ldots, t_n$ for compensation. If F_j is the cumulative distribution function of the j-th measurement x_j, thresholds t_j are to be chosen such that $\forall j$: $\Pr\left[\theta_{t_j}(x_j) = 1\right] = p$ (iff $\Pr\left[|x_j| > t_j\right] = p$ iff $F_j(-t_j) + 1 - F_j(t_j) = p$). For example, if $F_j(x) = \Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}}\, e^{-\frac{x^2}{2}}\, \mathrm{d}x$—Gauss $\mathrm{N}(0,1)$—, t_j is to be chosen so that $\Phi(-t_j) + 1 - \Phi(t_j) = p$, or $t_j = \Phi^{-1}\left(1 - \frac{p}{2}\right)$.

5. Conclusion

Algorithms 1, 2 and 3 have been implemented and applied to measurement data of more than 100,000 chips of different products. Results tended to be excellent when the overall defect state to be detected is accompanied by measurement data—for example, S2 overall defect state using S1 and S2 measurements. Properties which made the task harder with below-excellent results include cases where the soft bin to be predicted belongs to a measurement step not included in the data base—for example, predicting S3 defect states using S1- and S2-measurements only. In cases where Algorithm 3 reached at least good classification quality, dimensional reduction according to its n_{diff} ranking may lead to considerable reduction of input data demand.

Fig. 6. Distribution of $\frac{\langle \mathrm{x},\mathrm{y}\rangle}{H(\mathrm{x})}$ with $n = 50$, $H(\mathrm{y}) = 40$, $p = 0.5$: simulation with cumulative, two-binomial formula and summed Gauss approximations.



Acknowledgement

I would like to thank Zoltán Sasvári for reading different manuscript versions and for helpful remarks and some corrections. The author is grateful to the reviewers for valuable comments.

Research leading to these results has received funding from the iRel40 project. iRel40 is a European co-funded innovation project that has been granted by the **ECSEL** Joint Undertaking (JU) under grant agreement **n$^{\mathrm{O}}$ 876659**. The funding of the project comes from the Horizon 2020 research programme and participating countries. National funding is provided by Germany, including the Free States of Saxony and Thuringia, Austria, Belgium, Finland, France, Italy, the Netherlands, Slovakia, Spain, Sweden, and Turkey.

Disclaimer: The document reflects only the author's view and the JU is not responsible for any use that may be made of the information it contains.

References

Angluin, D. and P. Laird (1988). Learning From Noisy Examples. *Machine Learning 2*, 343–370.

Baker, R. J. (2010). *CMOS: Circuit Design, Layout, and Simulation* (3rd ed.). IEEE Press Series on Microelectronic Systems. John Wiley & Sons, Ltd.

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera (2020). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion 58*, 82–115.

Ben-David, S., N. Eiron, and P. M. Long (2003). On the Difficulty of Approximately Maximizing Agreements. *J. Comput. Syst. Sci. 66*(3), 496–514.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership
SPONSORED BY THE
Federal Ministry of Education and Research
Powered by
Freistaat SACHSEN
iRel4.0 Intelligent Reliability 4.0

COLT'92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. ACM.

Cortes, C. and V. Vapnik (1995). Support-Vector Networks. *Machine Learning 20*, 273–297.

Domingues, R., M. Filippone, P. Michiardi, and J. Zouaoui (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition 74*, 406–421.

Dua, D. and C. Graff (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`. (Date accessed: Oct 9, 2021).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*(2), 179–188.

Gao, L., J. Schulman, and J. Hilton (2022). Scaling Laws for Reward Model Overoptimization. `https://arxiv.org/pdf/2210.10760.pdf`. (Date accessed: Feb 6, 2023).

iRel4.0 (2020). Intelligent Reliability 4.0. `https://www.irel40.eu`. (Date accessed: Feb 5, 2022).

Kearns, M. and M. Li (1993). Learning in the Presence of Malicious Errors. *SIAM Journal on Computing 22*(4), 807–837.

Kiran, B. R., I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez (2022). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems 23*(6), 4909–4926.

Kriegel, H.-P., M. Schubert, and A. Zimek (2008). Angle-based outlier detection in high-dimensional data. In *KDD'08: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 444–452.

Liu, F., K. Ting, and Z. Zhou (2008). Isolation forest. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Mandic, D. P. and J. A. Chambers (2001). *Recurrent Neural Networks for Prediction*. John Wiley & Sons, Ltd.

Minsky, M. and S. Papert (1969). *Perceptrons: An Introduction to Computational Geometry* (1st ed.). The MIT Press.

Olschewski, T. (2021a). Defect Detection on Semiconductor Wafers by Distribution Analysis. `https://arxiv.org/abs/2111.03727v1`. (Date accessed: Feb 5, 2022).

Olschewski, T. (2021b). Fast Accurate Defect Detection in Wafer Fabrication. `https://arxiv.org/abs/2108.11757v1`. (Date accessed: Oct 9, 2021).

Olschewski, T., P. Biele, J. Bartholomäus, Z. Sasvári, and S. Wunderlich (2020). Case study - Evaluation of the ranking results of various outlier detection methods applied to semiconductor measurement data. Technical Report, Technische Universität Dresden.

Pham, N. and R. Pagh (2012). A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 877–885.

Productive4.0 (2017). Productive 4.0. `https://productive40.eu`. (Date accessed: Feb 5, 2022).

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Books.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature 323*(6088), 533–536.

Samek, W., G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller (2021). Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proceedings of the IEEE 109*(3), 247–278.

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science 2*(420).

Sugiyama, M. and K. M. Borgwardt (2013). Rapid distance-based outlier detection via sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 467–475. Curran Associates, Inc.

Sumikawa, N., L.-C. Wang, and M. S. Abadir (2013). A pattern mining framework for inter-wafer abnormality analysis. In *2013 IEEE International Test Conference (ITC)*, pp. 1–10.

Valiant, L. (1984). A Theory of the Learnable. *Communications of the ACM 27*(11), 1134–1142.

Valiant, L. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85) 1*, 560–566.

Zeiler, M. D. and R. Fergus (2014). Visualizing and Understanding Convolutional Networks. In *Computer Vision – ECCV 2014, Proceedings, Part I*, LNCS 8689, pp. 818–833. Springer.