On the Gaussian Distribution of Strings

Sébastien REBECCHI and Jean-Michel JOLION
Université de Lyon, F-69361 Lyon
INSA Lyon, F-69621 Villeurbanne
CNRS, LIRIS, UMR 5205
{*sebastien.rebecchi, jean-michel.jolion*}*@liris.cnrs.fr*

Abstract

We propose in this paper a definition of the normal law on the space of strings. This definition allows generations of sets of strings that have controled statistical features. Moreover, we apply with success a method of estimation of the mean and standard deviation of a normal string distribution.

1. Introduction

For many years, several research teams have tried to blend the two main approaches of pattern recognition, namely the statistical and structural ones [3, 2]. The wish is to be able to benefit from the advantages of the two approaches, while being detached from their respective drawbacks.

The statistical pattern recognition is based on a coding of the data in the form of numerical vectors, often unable to accurately reproduce the complexity of the data. But this choice is justified by the broad pallet of statistical algorithms published in the literature and recognized as powerful with numerical data [1].

In the structural pattern recognition paradigm, the coding part is rich because of being based on data structures of great expressivity (graphs, strings...). However, the tools related to the classification of structures are too restrictive (isomorphism, edit distance[4]...) and not robust enough for some applications specific to pattern recognition.

We propose in this paper to translate the concept of distribution to spaces of structures. We concentrate our attention to strings for which we propose the definition of a distribution controled by a normal probability density function (pdf), together with algorithms that respectively permit to generate sets of such strings, and estimate the mean and standard deviation of a normal distribution, from a sample of strings. Before going into details in section 3, we introduce some necessary notations and definitions.

2. Preliminary notations and definitions

Let A be an alphabet, *i.e.* a finite non-empty set whose elements are called letters. We denote by $|A|$ the number of letters of A.

A string X over A is a finite-length sequence of letters of A. We denote by $|X|$ the length of X, X_i the i-th letter of X ($i \in \{1, \dots, |X|\}$), A^* the set of strings over A, and λ the empty string, *i.e.* of length 0.

The concatenation operator, denoted by ., often omitted for simplicity, is defined over couples (X, Y) of strings or letters (as strings of length 1), as follows: $[X.Y(=_{\text{notation}} XY) = Z] \iff [|Z| = |X| + |Y|] \ \wedge \ [\forall i \in \{1, \dots, |X|\}, Z_i = X_i] \ \wedge \ \big[\forall i \in \{1, \dots, |Y|\}, Z_{|X|+i} = Y_i\big]$.

The prefix X' of length n of a string X is the string made of the concatenation of the n-th first letters of X: $[|X'| = n] \wedge [\forall i \in \{1, \dots, n\}, X'_i = X_i]$.

An edit operation over A is a couple (a, b) of elements of $A \cup \{\lambda\}$, denoted by $a \to b$. We say that such an operation transforms a into b, or consumes a to produce b. Moreover, if $a = \lambda$ (resp. $b = \lambda$), it is the insertion of b (resp. deletion of a), and if $a \neq \lambda$ and $b \neq \lambda$, it is the substitution of a by b.

An edit string over A is a string of edit operations over A (string over the alphabet $(A \cup \{\lambda\})^2$). Let $Z = X_1 \to Y_1 \dots X_n \to Y_n$ be such a string. We say that Z consumes (resp. produces) the string (over A) made of the concatenation of the letters consumed (resp. produced) by its operations, *i.e.* $X_1 \dots X_n$ (resp. $Y_1 \dots Y_n$).

Let c be an edit cost function over A: $\forall (a, b) \in (A \cup \{\lambda\})^2$, $a \neq b$, $c(a \to a) = 0$, and $c(a \to b) > 0$. c can translate a knowledge related to the dissimilarity of a and b, or to the likelihood of the transformation of a into b.

3. The string gaussian distribution

3.1. Definition

As initialy proposed in [2], a deviation in the letter domain is an edit operation consuming the mean letter (except λ if we expect no letter at all), and so on in the string domain an edit string consuming the mean string.

Refering to a multivariate paradigm, and with an assumption of mutual independance between the letter distributions composing a string, we define the gaussian distribution of strings over A^*, of standard deviation the edit string D over A, by the concatenation of $|D|$ gaussian distributions over $A \cup \{\lambda\}$, where the i-th one is of standard deviation D_i, $i \in \{1, \ldots, |D|\}$.

A gaussian distribution over $A \cup \{\lambda\}$, with standard deviation $m \to d$, is the distribution that assigns to each letter $a \in A \cup \{\lambda\}$, a probability equals to the density of probability of $c(m \to a)$ assigned by a normal pdf over $\mathbb{R}$ of mean 0 ($c(m \to m)$) and standard deviation $c(m \to d)$, normalized by the sum over $A \cup \{\lambda\}$ of all of these densities. That is, we use the cost function c to be able to define the concept of gaussian letter deviation, by means of a discretization process of a numerical normal law.

The generative and predictive processes are respectively detailed by the algorithms 1 and 2. The generative algorithm merely applies the iterative process implied by the reasoning above, of complexity $O(|D|)$. The complexity of the predictive algorithm is higher. Indeed, a string X is unaffected by insertions of as much of λ as desired at any position where it is possible, and so the number of possibilities of production of X is equal to the number of combinations of size $|X|$ from a set of size $|D|$. To solve this problem, we use a dynamic programming method of temporal complexity $O(|X| \times |D|^3)$ and space complexity $O(|D|)$. At each iteration j of the main central loop, PP[i] contains the probability of production of the prefix of length j of X, according to a normal law of standard deviation the prefix of length i of D. The special case $X = \lambda$ does not enter in this scheme, and is thus treated separately.

3.2. Estimation

Given a set of strings S, we want to estimate the parameters of the underlying normal distribution G.

The estimate of the standard deviation D is the edit string $\widehat{D_{\max}}$ obtained by the maximization of the likelihood of S: $\widehat{D_{\max}} = \arg\max\{\widehat{D} \in ((A \cup \{\lambda\})^2)^*\} \prod_{X \in S} \mathrm{P}(X|\widehat{G})$, with $\widehat{G}$ the normal law of standard deviation $\widehat{D}$, and $\mathrm{P}(X|\widehat{G})$ the probability of X according to $\widehat{G}$. We think that this optimization problem

Input: An edit string $D = m_1 \to d_1 \ldots m_n \to d_n$ over A
Output: A string X generated according to a normal law over A^*, of standard deviation D
begin
  $X \longleftarrow \lambda$;
  for $i \longleftarrow 1$ **to** n **do**
    $g \longleftarrow \mathcal{N}(0, c(m_i \to d_i)^2)$;
    $l \longleftarrow$ random choice according to the following distribution: $\forall a \in A \cup \{\lambda\}$:

$$\mathrm{P}(a) = \frac{g\left(c(m_i \to a)\right)}{\sum_{b \in A \cup \{\lambda\}} g\left(c(m_i \to b)\right)};$$

    $X \longleftarrow X.l$;
  end
  return X;
end

Algorithm 1: String generation according to a normal law

Input: An edit string $D = m_1 \to d_1 \ldots m_n \to d_n$ over A
Output: The probability of λ according to a normal law over A^*, of standard deviation D
begin
  $P \longleftarrow 1.0$;
  for $i \longleftarrow 1$ **to** n **do**
    $P \longleftarrow P \times \mathrm{P}(\lambda$ in position $i)$;
    // algorithm 4
  end
  return P;
end

Algorithm 3: Prediction of the probability of λ according to a normal law

Input: A letter $l \in A \cup \{\lambda\}$, an edit operation $m \to d$ over A
Output: The probability of l according to a normal law over $A \cup \{\lambda\}$, of standard deviation $m \to d$
begin
  $g \longleftarrow \mathcal{N}(0, c(m \to d)^2)$;
  return

$$\frac{g\left(c(m \to l)\right)}{\sum_{b \in A \cup \{\lambda\}} g\left(c(m \to b)\right)};$$

end

Algorithm 4: Prediction of the probability of a letter according to a normal law

Input: A string X over A, an edit string $D = m_1 \rightarrow d_1 \ldots m_n \rightarrow d_n$ over A
Output: The probability of X according to a normal law over A^*, of standard deviation D
begin
  /* Special case $X = \lambda$ */
  if $X = \lambda$ **then return** $\mathrm{P}(\lambda)$; // algorithm 3
  /* Inizialisation : production of the first letter of X */
  for $i \longleftarrow 1$ **to** $|D| - |X| + 1$ **do**
    // Production of X_1 in position i of D
    PP[i] $\longleftarrow \mathrm{P}(X_1$ in position $i)$; // algorithm 4
    // Leftmost completion with productions of λ
    for $l \longleftarrow 1$ **to** $i - 1$ **do** PP[i] $\longleftarrow$ PP[i] $\times \mathrm{P}(\lambda$ in position $l)$; // algorithm 4
  end
  for $i \longleftarrow |D| - |X| + 2$ **to** $|D|$ **do**
    // Impossible to begin the production in this position i of D
    PP[i] $\longleftarrow 0.0$;
  end
  /* Iterations : production of the other letters of X */
  for $j \longleftarrow 2$ **to** $|X|$ **do**
    // Possibilities of production of X_j in position i of D
    for $i \longleftarrow |D|$ **to** 1 **do**
      PP[i] $\longleftarrow 0$; // initialization
      // Possibilities of production of X_{j-1} in position $k < i$ of D
      for $k \longleftarrow 1$ **to** $i - 1$ **do**
        // Production of X_{j-1} in position k of D
        PLUS $\longleftarrow$ PP[k];
        // Completion with productions of λ in positions $(k + 1) \leqslant l \leqslant (i - 1)$
        for $l \longleftarrow k + 1$ **to** $i - 1$ **do** PLUS $\longleftarrow$ PLUS $\times \mathrm{P}(\lambda$ in position $l)$; // algorithm 4
        // Addition of the probabilities of production of X_{j-1}
        PP[i] $\longleftarrow$ PP[i] + PLUS;
      end
      // Production of X_j in position i of D
      PP[i] $\longleftarrow$ PP[i] $\times \mathrm{P}(X_j$ in position $i)$; // algorithm 4
    end
  end
  /* Finalization of the possibilities of production of X */
  $P \longleftarrow 0.0$;
  for $i \longleftarrow 1$ **to** $|D|$ **do**
    // The production of X was stopped position i of D
    PLUS $\longleftarrow$ PP[i];
    // Rightmost completion with productions of λ
    for $l \longleftarrow i + 1$ **to** $|D|$ **do** PLUS $\longleftarrow$ PLUS $\times \mathrm{P}(\lambda$ in position $l)$; // algorithm 4
    // Addition of the probabilities of production of X
    $P \longleftarrow P +$ PLUS;
  end
  return P;
end

Algorithm 2. Prediction of the probability of a string according to a normal law

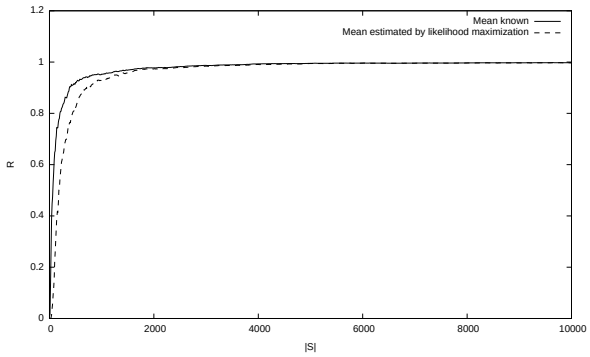

Figure 1. Ratios of perplexities of the estimates

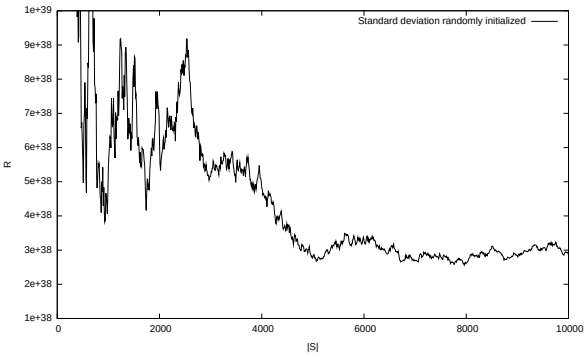

Figure 2. Ratios of perplexities of a normal law of random deviation

is NP-hard. But in the specific case where the cost function makes improbable any insertion or deletion (cost $+\infty$), all strings of S have same length $|D|$, and we know that all letters in the same position in the strings of S have been generated by the same normal law of letters. Therefore, we have: $\widehat{D_{\max}} = \widehat{d_{1\max}} \dots \widehat{d_{|D|\max}}$, with, $\forall\ 1 \leqslant i \leqslant |D|$, $\widehat{d_{i\max}} = \arg\max\{\widehat{d_i} \in A^2\} \prod_{X \in S} \mathrm{P}(X_i|\widehat{G_i})$, with $\widehat{G_i}$ the normal law of standard deviation $\widehat{d_i}$. It is thus effectively possible, in that case, to find $\widehat{D_{\max}}$ in $O(|D| \times |A|^2 \times |S|)$.

Figure 1 shows the result of an estimation process in that specific case. The alphabet is the set of non-special ASCII characters ($|A| = 95$), and the strings are generated with length 100. The cost of a subsitution of 2 characters is the Hamming distance of their repective visual coding by a binary 7×7 matrix. To have a better interpretation, the effectively presented measures are the ratios $R = \widehat{P}/P$, with $\widehat{P}$ the perplexity of S by $\widehat{G_{\max}}$, and P by G. The perplexity is just the reverse of the average likelihood of an element of S, and such a ratio $R > 1$ is interpreted as the fact that a string of S is R times as likeliky by G than by $\widehat{G_{\max}}$, and if $R < 1$, $1/R$ times as likeliky by $\widehat{G_{\max}}$ than by G. Notice that minimizing the perplexity maximizes the likelihood.

We have divided the experimentation in two levels of difficulties: estimating the standard deviation with known or unknown mean. In the first case, the consumed string of the standard deviation estimate is obsiously set to the mean, and it remains to estimate the produced one. In the second case, the consumed string of the estimate provides an estimate of the mean. In all cases, we can see that the method leads to good results, $\widehat{G_{\max}}$ converging very quickly to G, due to the fact that we don't need to generate a great number of strings (relatively to the 95^{100} non improbable strings) to reach a rather good representativy of S regarding to G. The cases where S is more likely by $\widehat{G_{\max}}$ than by G appears when S is too small to be representative of G, leading to the existence of normal laws different of G for which S is more representative. Finally, this good conclusion is emphasized by the very bad results obtained by a random estimate, as presented in figure 2.

4. Conclusion

We have presented a new gaussian string distribution definition, with a method to estimate its (mean and) standard deviation, leading to encouraging results.

This work is a prelude and it would be interesting to study the possible properties of the distribution, particularly those that could arise when combining, in a manner to be specified yet, independent executions of the same string distribution, as told by the central limit theorem. A necessary condition is to define the concept of random variable in a measurable vector space for strings. It could be sufficient to order the strings to keep the real line as the state set of such random variables, and thus take advantage of the classical (numerical) probabilistic statistical theory.

References

[1] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[2] J.-M. Jolion. The deviation of a set of strings. *Pattern Analysis and Applications*, 6(3):224–231, 2003.

[3] T. Kohonen. Median strings. *Pattern Recognition Letters*, 3(5):309–313, 1985.

[4] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.