PROF. FRANCESCO A.N. PALMIERI
UNIV. DELLA CAMPANIA

APPENDIX
DISCRETE VARIABLES REPRESENTATIONS

In manipulating classifier data, as we will see in the following, it's convenient to use continuous mathematics because we can compute gradients and define smooth functional transformations.
One of the problem that we have to face to train a parametric model $f(\underline{x}, \underline{\theta})$ is that our teaching pairs may include a desired output $\underline{s}$ that belongs to a discrete, non necessarily numerical set.
More specifically, if the objective of $f(\underline{x}, \underline{\theta})$ is to produce a decision, or a posterior distribution, what should be the representation of $\underline{s}$ in our use of a loss function?
For example, when the classifier alphabet is $\mathcal{A} = \{$ HOME, GRASS, SKY, HORSE $\}$, how can we define a loss function to be used for training in comparing $f(\underline{x}, \underline{\theta})$ to each element of $\mathcal{A}$?
Furthermore a classification, or a regression problem, may be based on discrete inputs.
We have so far always assumed that $\underline{x} \in \mathcal{X} \subset \mathbb{R}^n$, but there are many problems in which $\underline{x}$ may be discrete.
To translate discrete variables into continuous representations there are various strategies:

Assume that each symbol $s \in \mathcal{A} = \{a_1, \dots, a_M\}$ is translated into a real vector $\underline{v}$ belonging to a finite set

$$\underline{v} \in \mathcal{V} = \{\underline{v}_1, \underline{v}_2, \dots, \underline{v}_M\}$$

$$s \in \mathcal{A} \longrightarrow \underline{v} \in \mathbb{R}^N$$

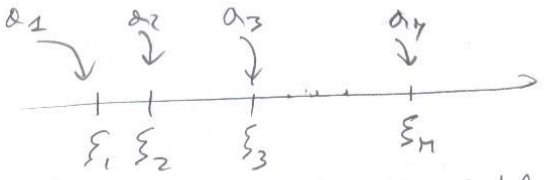
$$a_1 \longrightarrow \underline{v}_1$$
$$a_2 \longrightarrow \underline{v}_2$$
$$\vdots$$
$$a_M \longrightarrow \underline{v}_M$$

MAGNITUDE ENCODING

The most direct way of encoding the symbols may be to use scalar values ($N = 1$)

The symbols in $\mathcal{A}=\{a_1, a_2 \dots a_M\}$ are translated into a finite discrete set of values $\{\xi_1, \dots, \xi_M\}$ on the real line.

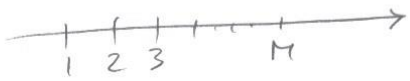
$s \in \mathcal{A}=\{a_1 \dots a_M\} \rightarrow v \in \{\xi_1 \dots, \xi_M\}$



(similar to PAM modulation)

If the original set $\mathcal{A}$ is not ordered the association may be arbitrary.

For example $\xi_1, \xi_2, \xi_3, \dots, \xi_M$ may be equally spaced and/or be the integers $1, \dots, M$.



This representation in neural networks, is rarely used because it lacks robustness with respect to misclassification.

TEMPERATURE CODING

$s \in \mathcal{A}=\{a_1, \dots, a_M\} \longrightarrow \underline{v}$ M dimensional vector of 0s and 1s

Ex $M=5$

$a_1 \rightarrow \underline{v}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$; $a_2 \rightarrow \underline{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$; $a_3 \rightarrow \underline{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$; $a_4 \rightarrow \underline{v}_4 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ $a_5 \rightarrow \underline{v}_5 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

ONE-HOT ENCODING

$s \in \mathcal{A}=\{a_1, \dots, a_M\} \longrightarrow \underline{v}$ M dimensional vector with all zeros except one equal to one.

Ex $M=5$

$a_1 \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ $\underline{v}_1$ $\quad a_2 \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ $\underline{v}_2$ $\quad a_3 \rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ $\underline{v}_3$ $\quad a_4 \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ $\underline{v}_4$ $\quad a_5 \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $\underline{v}_5$

ONE-HOT ε CODING

$S \in \mathcal{X} = \{a_1 \dots a_M\} \rightarrow \underline{V}$ ($0 < \varepsilon < 1$ small)

$M = 5$

$a_1 \rightarrow \underline{V_1} = \begin{bmatrix} 1-\varepsilon \\ \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \end{bmatrix}$ $a_2 \rightarrow \underline{V_2} = \begin{bmatrix} \varepsilon/4 \\ 1-\varepsilon \\ \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \end{bmatrix}$ $a_3 \rightarrow \underline{V_3} = \begin{bmatrix} \varepsilon/4 \\ \varepsilon/4 \\ 1-\varepsilon \\ \varepsilon/4 \\ \varepsilon/4 \end{bmatrix}$ $a_4 \rightarrow \underline{V_4} = \begin{bmatrix} \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \\ 1-\varepsilon \\ \varepsilon/4 \end{bmatrix}$ $a_5 \rightarrow \underline{V_5} = \begin{bmatrix} \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \\ \varepsilon/4 \\ 1-\varepsilon \end{bmatrix}$

In both ONE-HOT and ONE-HOT ε REPRESENTATIONS $\underline{V}$ is a distribution

ONE-HOT ε representation is used to avoid the presence of zeros that may create numerical instabilities during learning.

BINARY UNIPOLAR CODING (FULL)

$M = 2^q$ $S \in \mathcal{X} = \{a_1 \dots a_M\} \rightarrow \underline{V} \in \{0,1\}^q$

Ex $M = 4$

$a_1 \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix}^{\underline{V_1}}$ $a_2 \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}^{\underline{V_2}}$ $a_3 \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\underline{V_3}}$ $a_4 \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}^{\underline{V_4}}$

BINARY BIPOLAR CODING (FULL)

$M = 2^q$ $S \in \mathcal{X} = \{a_1 \dots a_M\} \rightarrow \underline{V} \in \{-1,1\}^q$

Ex $M = 4$

$a_1 \rightarrow \begin{bmatrix} -1 \\ -1 \end{bmatrix}^{\underline{V_1}}$ $a_2 \rightarrow \begin{bmatrix} -1 \\ 1 \end{bmatrix}^{\underline{V_2}}$ $a_3 \rightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix}^{\underline{V_3}}$ $a_4 \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}^{\underline{V_4}}$

BINARY UNIPOLAR CODING (SPARSE)

$M << 2^q$ $S \in \mathcal{X} \in \{a_1, \dots, a_M\} \rightarrow \underline{V} \in \{0,1\}^q$

Ex $M = 5$, $q = 4$

$a_1 \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}^{\underline{V_1}}$ $a_2 \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^{\underline{V_2}}$ $a_3 \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\underline{V_3}}$ $a_4 \rightarrow \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}^{\underline{V_4}}$ $a_5 \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\underline{V_5}}$

BINARY BIPOLAR CODING (SPARSE)

$$a_1 \to \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \underline{v}_1 \qquad a_2 \to \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \underline{v}_2 \qquad a_3 \to \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \underline{v}_3 \qquad a_4 \to \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \underline{v}_4 \qquad a_5 \to \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \underline{v}_5$$

(The transformation from binary $b \in \{0,1\}$ to bipolar is

$b = 0$
$b = 1$ $\Rightarrow x = 2b - 1 \Rightarrow$ -1, 1)

CONTINUOUS EMBEDDING

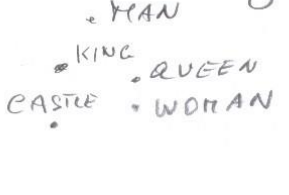
The most general case is where

$a_1 \to \underline{v}_1$
$a_2 \to \underline{v}_2$
$\vdots$
$a_M \to \underline{v}_M$

Each symbol a_i is mapped into an N-dimensional vector $\underline{v}_i \in \mathbb{R}^N$.

Usually the vectors are chosen in a way that reflects some kind of relation among the symbols. For examples, in processing words, we use embeddings that tend to preserve semantic similarities (WORD2VEC)

Ex $\mathcal{A} =$ {KING, QUEEN, CASTLE, MAN, WOMAN, ...}



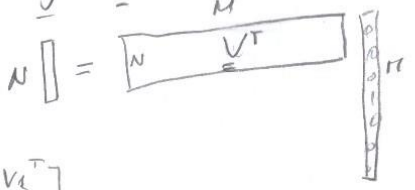
The embeddings are usually represented as the following scheme.

WORDS (ONE HOT ENCODING) $\underline{x}$ → $\underline{\underline{V}}$ → $\underline{v}$ embedding

M (size of the vocabulary)

$\underline{y} = \underline{\underline{V}}^T \underline{x}$

vocabulary



$$\underline{\underline{V}} = \begin{bmatrix} \underline{V}_1^T \\ \underline{V}_2^T \\ \underline{V}_3^T \\ \vdots \\ \underline{V}_M^T \end{bmatrix} \leftarrow \text{each row of } \underline{\underline{V}} \text{ is the vector for a word}$$

The embedding can be inserted in various architectures and be pre-learned, or learned with all the other parameters of the network.

[see the literature for schemes to learn word embeddings]

This topic deserves a separate discussion that will be reported elsewhere.