# Generative Multinomial Models

Sandro Cumani

sandro.cumani@polito.it

Politecnico di Torino

## Modeling discrete values

We consider a problem characterized by discrete features

For the moment, we assume we have a single categorical feature $x \in \{1 \ldots m\}$.

For example, we may want to predict a cat gender from the fur color

## Modeling discrete values

The categorical feature is the fur color, for example

$$x \in \{white, black, orange, gray, bi\text{-}color, calico, ...\}$$

Also in this case, we have a set of labeled training samples

$$\mathcal{D} = \{(x_1, c_1) \dots (x_n, c_n)\}$$

Each sample $x_i$ is simply the fur color of the $i$-th cat sample, and $c_i$ is the gender label

## Modeling discrete values

We also assume that samples are i.i.d. (as in the Gaussian case)

We want to compute the probabilities

$$P(X_t = x_t | C_t = c) = \pi_{c,x_t}$$

for the different hypotheses $c$, for the observed test sample $x_t$

In our example, these are the probabilities of observing fur color $x_t$ under the different gender hypotheses $c \in \{female, male\}$ (e.g. $P(X_t = white | C_t = female)$)

$\boldsymbol{\pi}_c = (\pi_{c,1} \dots \pi_{c,m})$ are the model parameters for class $c$, with

$$\sum_{i=1}^{m} \pi_{c,i} = 1$$

## Modeling discrete values

Again, we can adopt a frequentist approach and estimate the ML solution for $\mathbf{\Pi} = (\boldsymbol{\pi}_1 \dots \boldsymbol{\pi}_k)$ ($k$ is again the number of classes)

We can express the likelihood for the training set as

$$\mathcal{L}(\mathbf{\Pi}) = \prod_{i=1}^{n} P(X_i = x_i | C_i = c_i) P(C_i = c_i)$$

where $c_i \in \{1 \dots k\}$ is the class of the $i$-th sample

## Modeling discrete values

For example, if our dataset consists of[1]

(black, male), (orange, male), (black, female), (orange, male),

(white, male), (white, female), (white, male), (white, female),

(black, female), (calico, female)

the likelihood is simply the product

$$\begin{aligned}\mathcal{L}(\boldsymbol{\Pi}) =& P(X_1 = black|C_1 = male)P(C_1 = male)\times \\ & P(X_2 = orange|C_2 = male)P(C_2 = male)\times \\ & P(X_3 = black|C_3 = female)P(C_3 = female)\times \\ & \vdots \\ & P(X_{10} = calico|C_{10} = female)P(C_{10} = female)\end{aligned}$$

[1] We assume we have associated each color to an integer value $\{1 \ldots m\}$, and that the gender labels are associated to $\{0, 1\}$

## Modeling discrete values

The log-likelihood is given by

$$\begin{aligned}
\ell(\boldsymbol{\Pi}) &= \sum_{i=1}^{n} \log P(X_i = x_i | C_i = c_i) + \xi \\
&= \sum_{i=1}^{n} \log \pi_{c_i, x_i} + \xi \\
&= \sum_{c=1}^{k} \sum_{i|c_i=c} \log \pi_{c, x_i} + \xi \\
&= \sum_{c=1}^{k} \ell_c(\boldsymbol{\pi}_c) + \xi
\end{aligned}$$

where

$$\ell_c(\boldsymbol{\pi}_c) = \sum_{i|c_i=c} \log \pi_{c, x_i}$$

Again, the log-likelihood can be expressed as a sum of terms, each depending on a separate subset of the parameters (those of class $c$)

## Modeling discrete values

We can thus estimate the parameters by independently optimizing $\ell_c(\boldsymbol{\pi}_c)$

We have

$$\ell_c(\boldsymbol{\pi}_c) = \sum_{i|c_i=c} \log \pi_{c,x_i} \\ = \sum_{j=1}^{m} N_{c,j} \log \pi_{c,j}$$

where $N_{c,j}$ is the number of times we observed $x_i = j$ in the dataset for class $c$ (i.e. the number of times the features of class $c$ were equal to $j$)

In our example we would have

$$N_{female,black} = 2 \quad N_{female,orange} = 0 \quad N_{female,white} = 2 \quad N_{female,calico} = 1$$
$$N_{male,black} = 1 \quad N_{male,orange} = 2 \quad N_{male,white} = 2 \quad N_{male,calico} = 0$$

## Modeling discrete values

Solving for $\pi_{c,x_i}$ is a bit more complex than in the Gaussian case, since we have the constraint that $\sum_{j=1}^{m} \pi_{c,j} = 1$

In the following we drop subscript $c$, thus we look for the maximizer of

$$f(\boldsymbol{\pi}) = \sum_{j=1}^{m} N_j \log \pi_j$$

subject to

$$\sum_{j=1}^{m} \pi_j = 1$$

A solution can be obtained by means of Lagrange multipliers

We look for stationary points of

$$L(\boldsymbol{\pi}) = f(\boldsymbol{\pi}) - \lambda \left( \sum_{j=1}^{m} \pi_j - 1 \right) = \sum_{j=1}^{m} N_j \log \pi_j - \lambda \left( \sum_{j=1}^{m} \pi_j - 1 \right)$$

## Modeling discrete values

We need to solve

$$\begin{cases} \frac{\partial L}{\partial \pi_i} = 0 \,, & \forall i = 1 \dots m \\ \frac{\partial L}{\partial \lambda} = 0 \end{cases}$$

We have

$$\frac{\partial L}{\partial \pi_i} = \frac{N_i}{\pi_i} - \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = 1 - \sum_{j=1}^{m} \pi_j = 0$$

We can verify that $\lambda \neq 0$, so that

$$\pi_i = \frac{N_i}{\lambda}$$

## Modeling discrete values

From the derivative with respect to $\lambda$ we thus have

$$\sum_{j=1}^{m} \pi_j = \sum_{j=1}^{m} \frac{N_j}{\lambda} = \frac{1}{\lambda} \sum_{j=1}^{m} N_j = 1$$

thus

$$\lambda = \sum_{j=1}^{m} N_j = N$$

where $N = \sum_{j=1}^{m} N_j$ is the number of samples for the considered class

We finally have

$$\pi_i = \frac{N_i}{N}$$

i.e., the frequency with which we observed value $i$ in the class

## Modeling discrete values

The ML solution for each class is thus

$$\pi^*_{c,i} = \frac{N_{c,i}}{N_c}$$

We can compute the predictive distribution for $x_t$ as

$$P(X_t = x_t | C_t = c) = \pi^*_{c,x_t}$$

In our example, we would have

$$N_{male} = 5 \quad N_{female} = 5$$

thus

$$\pi^*_{female,black} = 0.4 \quad \pi^*_{female,orange} = 0 \quad \pi^*_{female,white} = 0.4 \quad \pi^*_{female,calico} = 0.2$$
$$\pi^*_{male,black} = 0.2 \quad \pi^*_{male,orange} = 0.4 \quad \pi^*_{male,white} = 0.4 \quad \pi^*_{male,calico} = 0$$

## Modeling discrete values

If we have more than one categorical attribute, we may model their joint probability as a categorical R.V. with values given by all possible combinations of the attributes

In practice, the number of elements would quickly become intractable

We can adopt again a naive Bayes approximation and assume that features are independent

We can obtain ML estimates for each feature, and then combine them:

$$P(\boldsymbol{X}_t = \boldsymbol{x}_t | C_t = c) = \prod_j \pi^j_{c, x_{t,[j]}}$$

where $j$ denotes the feature index

## Modeling discrete values

We now consider an extended version of the problem, where features represent occurrences of events

Typical examples are topic or language modeling

We may, for example, represent a text in terms of the words that appear inside

Different topics will result in different sequences of words

Modeling all possible combinations of words is impractical, since the number of categories would grow exponentially with the number of words

## Modeling discrete values

As an approximation, we may represent documents in terms of occurrences of single words

Note that such model ignores the order in which words appear — we may improve the model by considering pairs or triplets of words

Thus, we have feature vectors $\boldsymbol{x} = (x_{[1]} \dots x_{[m]})$, where each $x_{[i]}$ represents the number of times we observed word $i$ in the document

## Modeling discrete values

We have seen that occurrences can be modeled by multinomial distributions

Thus, for each class $c$, we have a set of parameters

$$\boldsymbol{\pi}_c = (\pi_{c,1} \dots \pi_{c,m})$$

that represents the probability of observing a single instance of word $i$

The probability for feature vector $\boldsymbol{x}$ is given by the multinomial density

$$P(\boldsymbol{X} = \boldsymbol{x}|C = c) = \frac{\left(\sum_{j=1}^{m} x_{[j]}\right)!}{\prod_{j=1}^{m} x_{[j]}!} \prod_{j=1}^{m} \pi_{c,j}^{x_{[j]}} \propto \prod_{j=1}^{m} \pi_{c,j}^{x_{[j]}}$$

## Modeling discrete values

Again, we can write the likelihood of the model — as in the previous case, the likelihood factorizes over classes, thus

$$\ell(\mathbf{\Pi}) = \sum_c \ell_c(\boldsymbol{\pi}_c) + \xi$$

with

$$\ell_c(\boldsymbol{\pi}_c) = \sum_{i|c_i=c} \sum_{j=1}^{m} x_{i,[j]} \log \pi_{c,j}$$

Note that we did not write the multinomial coefficient since it's constant with respect to $\mathbf{\Pi}$, and thus can be absorbed in $\xi$.

## Modeling discrete values

Also in this case we can express the log-likelihood as

$$\ell_c(\boldsymbol{\pi}_c) = \sum_{j=1}^{m} N_{c,j} \log \pi_{c,j}$$

where $N_{c,j}$ is the total number of occurrences of event $j$ (word $j$) in the samples of class $c$:

$$N_{c,j} = \sum_{i|c_i=c} x_{i,[j]}$$

To solve the problem, we notice that it has exactly the same form as the one we solved for the categorical case

## Modeling discrete values

Thus, the ML solution is again

$$\pi_{c,j} = \frac{N_{c,j}}{N_c}$$

where, in this case, $N_c$ is the total number of words for class $c$

$$N_c = \sum_j N_{c,j} = \sum_{i|c_i=c} \sum_{j=1}^{m} x_{i,[j]}$$

$\pi_{c,j}$ is therefore again the relative frequency of word $j$ in class $c$

## Modeling discrete values

Again, we write the log-likelihood ratio for a two class problem (we use the expression of the multinomial log-probability — notice that the binomial coefficient can be simplified and thus disappears from the final expression):

$$
\begin{aligned}
llr(\boldsymbol{x}) &= \log \frac{P(\boldsymbol{X} = \boldsymbol{x} | C = h_1)}{P(\boldsymbol{X} = \boldsymbol{x} | C = h_0)} \\
&= \sum_{j=1}^{m} x_{[j]} \log \pi_{h_1,j} - \sum_{j=1}^{m} x_{[j]} \log \pi_{h_0,j} \\
&= \boldsymbol{x}^T \boldsymbol{b}
\end{aligned}
$$

with

$$
\boldsymbol{b} = (\log \pi_{h_1,1} - \log \pi_{h_0,1}, \ldots \log \pi_{h_1,m} - \log \pi_{h_0,m})
$$

Again, we have linear decision functions

## Modeling discrete values

An example: inferring programming language

We consider the problem of automatically inferring whether a file is written in C or Python

We consider a simple model based on the occurrences of punctuation characters: `{} [] () : ; . ,`

We treat open and closed brackets of the same kind as a single symbol

We model scripts in terms of occurrences of the aforementioned symbols

## Modeling discrete values

Let's consider the following training set

| | {} | [] | () | : | ; | . | , |
|---|---|---|---|---|---|---|---|
| C script 1 | 6 | 8 | 14 | 1 | 10 | 1 | 7 |
| C script 2 | 8 | 10 | 14 | 0 | 11 | 1 | 7 |
| C script 3 | 12 | 22 | 34 | 1 | 21 | 2 | 13 |
| C script 4 | 4 | 6 | 10 | 1 | 6 | 1 | 4 |
| C total | 30 | 46 | 72 | 3 | 48 | 5 | 31 |
| Py script 1 | 6 | 14 | 30 | 6 | 2 | 16 | 16 |
| Py script 2 | 2 | 8 | 14 | 3 | 1 | 9 | 8 |
| Py script 3 | 4 | 14 | 26 | 7 | 2 | 15 | 14 |
| Py total | 12 | 36 | 70 | 16 | 5 | 40 | 38 |

## Modeling discrete values

The ML estimate for the model parameters are the symbol frequencies for each class :

| | {} | [ ] | ( ) | : | ; | . | , |
|---|---|---|---|---|---|---|---|
| $\pi_C$ | 0.128 | 0.196 | 0.306 | 0.013 | 0.204 | 0.021 | 0.132 |
| $\pi_{Py}$ | 0.055 | 0.166 | 0.323 | 0.074 | 0.023 | 0.184 | 0.175 |

To infer the class of an unknown script, we need to compute class posterior.

In this case, the task is binary so we compute log-likelihood ratios $llr(\boldsymbol{x})$, where $\boldsymbol{x}$ is the vector of occurrences of the symbols in the test script

The log-likelihood ratio can be expressed as $llr(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{b}$

## Modeling discrete values

Each element of $\boldsymbol{b}$ is $b_i = \log \frac{\boldsymbol{\pi}_{C,[i]}}{\boldsymbol{\pi}_{Py,[i]}}$

| | `{}` | `[]` | `()` | `:` | `;` | `.` | `,` |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{b} \approx$ | 0.837, | 0.165, | -0.052, | -1.754, | 2.182, | -2.159, | -0.283 |

The model parameters already tell us what symbols are most discriminant

Observing `;` is much more likely for C scripts

Observing `:` or `.` is much more likely for Python scripts

Round brackets are not very discriminant

## Modeling discrete values

We consider two test scripts:

| | {} | [ ] | ( ) | : | ; | . | , |
|---|---|---|---|---|---|---|---|
| Test script $\boldsymbol{x}_1$ (C) | 2 | 10 | 12 | 0 | 1 | 1 | 0 |
| Test script $\boldsymbol{x}_2$ (Py) | 2 | 18 | 16 | 3 | 0 | 1 | 1 |

The llr for the first script is $llr(\boldsymbol{x}_1) \approx 2.7$, i.e. it's more likely that we observe the occurrences of $\boldsymbol{x}$ in C scripts

For the second script we have $llr(\boldsymbol{x}_2) \approx -3.9$, i.e. it's significantly more likely that we observe the occurrences in $\boldsymbol{x}_2$ in Python scripts

Class posteriors can be computed once we choose class priors

With uniform priors, $\boldsymbol{x}_1$ would be assigned to C class, and $\boldsymbol{x}_2$ to Python class

## Modeling discrete values

Practical considerations:

- Rare words can cause problems: if a word does not appear in a topic we will estimate a probability $\pi_{c,j} = 0$. Any test sample that contains the word will have $0$ probability of being from class $c$
- We can mitigate the issue introducing pseudo-counts, i.e. assuming that each topic contains a sample were all words appear a (fixed) number of times
- In practice, we can add a fixed values to the class occurrences $N_c$ before computing the ML solution
- This corresponds to a Maximum-a-posterior estimate using a Dirichlet prior distribution (we won't see the details in this course)

## Modeling discrete values

Practical considerations:

- Bayesian treatment of the hyperparameters is a more appropriate way to deal with limited sample sizes (but we will not analyze Bayesian models)
- We can consider pairs of words, triplets of words and so on if we want to partially account for correlations
- We can also combine different models (categorical, multinomial, Gaussian, ...) through a naive Bayes assumption

## Modeling discrete values

Some additional comments on the discrete models for categorical events:

- We can model a dataset of categorical samples as $n$ independent categorical R.V.s $X_i \in \{1 \dots m\}$.
- Each variable represents a token.
- The distribution is described by a vector of probabilities $\boldsymbol{\pi}$ that allow computing $P(X = j) = \pi_j$

## Modeling discrete values

Some additional comments on the discrete models for categorical events:

- Alternatively, we can model the dataset in terms of occurrences of events.
- We have a random vector $\boldsymbol{Y} = (Y_1 \dots Y_m)$ whose components are R.V. $Y_i$ corresponding to the number of occurrences of event $i$ in the dataset.
- Again, the distribution is described by a vector of probabilities $\boldsymbol{\pi}$ that represent probabilities of single events

The two models are related by

$$Y_j = \sum_{i=1}^{n} \mathbb{I}\left[X_i = j\right]$$

## Modeling discrete values

As we have seen, in both cases the ML solution for $\boldsymbol{\pi}$ corresponds to the relative frequencies of occurrences. Indeed, we can go further and show that the two models are equivalent

Considering only samples of a single class, in the first case the log-likelihood for a given class is given by

$$\ell_X(\boldsymbol{\pi}) = \sum_{i=1}^{n} \log \pi_{x_i} = \sum_{j=1}^{m} N_j \log \pi_j$$

whereas in the second case

$$\ell_Y(\boldsymbol{\pi}) = \log \frac{\left(\sum_{j=1}^{m} y_{[j]}\right)!}{\prod_{i=1}^{m} y_{[j]}!} + \sum_{j=1}^{m} y_j \log \pi_j = \xi + \sum_{j=1}^{m} N_j \log \pi_j$$

## Modeling discrete values

The corresponding likelihoods $\mathcal{L}_X(\boldsymbol{\pi})$ and $\mathcal{L}_Y(\boldsymbol{\pi})$ are proportional:

$$\mathcal{L}_X(\boldsymbol{\pi}) \propto \mathcal{L}_Y(\boldsymbol{\pi})$$

Therefore:

- Maximum likelihood estimates will be the same
- Bayesian posterior probabilities for model parameters will be the same (again, we will not go into details of Bayesian models)
- Inference will be the same

## Modeling discrete values

The first point is straightforward

Regarding the last point, we can observe that, for a given test sample, the class conditional likelihoods are also proportional

$$f_{\boldsymbol{X}|C}(\boldsymbol{x}_t|c) = \zeta(\boldsymbol{y}_t) \cdot f_{\boldsymbol{Y}|C}(\boldsymbol{y}_t|c)$$

Thus, binary log-likelihood ratios are equal

$$\log \frac{f_{\boldsymbol{X}|C}(\boldsymbol{x}_t|h_1)}{f_{\boldsymbol{X}|C}(\boldsymbol{x}_t|h_0)} = \log \frac{f_{\boldsymbol{Y}|C}(\boldsymbol{y}_t|h_1)}{f_{\boldsymbol{Y}|C}(\boldsymbol{y}_t|h_0)}$$

and similarly for class posteriors:

$$P(C_t = c|\boldsymbol{X} = \boldsymbol{x}) = \frac{f_{X|C}(\boldsymbol{x}_t|c)P(c)}{\sum_{c'} f_{X|C}(\boldsymbol{x}_t|c')P(c')} = \frac{f_{Y|C}(\boldsymbol{y}_t|c)P(c)}{\sum_{c'} f_{Y|C}(\boldsymbol{y}_t|c')P(c')} = P(C_t = c|\boldsymbol{Y} = \boldsymbol{y})$$