Statistical methods in NLP
Random variables in Scipy

UNIVERSITY OF GOTHENBURG

Richard Johansson

January 26, 2016

overview

recap: random variables

random variables in Scipy

the cumulative distribution function

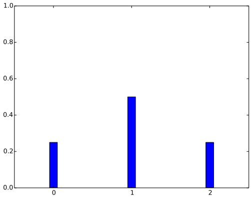
random variables and their distributions

- a **random variable** (r.v.) is a variable that selects its value randomly, like `random.randint` and `random.random`
- to describe the distribution of the r.v. X, we use a function called the **probability mass function** (pmf) of X:

$$p_X(x) = P(X \text{ takes the value } x)$$

- for instance, the number of heads when tossing a coin twice:

$$p_X(0) = P(X = 0) = \frac{1}{4}$$
$$p_X(1) = P(X = 1) = \frac{2}{4}$$
$$p_X(2) = P(X = 2) = \frac{1}{4}$$



the mean value of a random variable

- the notion of **mean** has a natural correspondence for random variables:
  - intuitively, this corresponds to what happens if we take the mean of a very large sample from the random variable

$$\mathrm{E}(X) = \sum_i p_X(i) \cdot i$$

- similarly, we have the **variance**: if $\mathrm{E}(X) = m$, then

$$V(X) = \mathrm{E}[(X - m)^2]$$

- and naturally, there is also a **standard deviation**

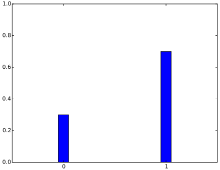
$$D(X) = \sqrt{V(X)}$$

the Bernoulli distribution

- we toss an uneven coin that gives heads ($X = 1$) with the probability p and tails ($X = 0$) with probability $1 - p$:

$$p_X(0) = 1 - p$$

$$p_X(1) = p$$

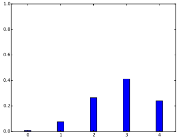


- X is then said to have a **Bernoulli** distribution with a parameter p
- a single experiment that can "succeed" or not

the binomial distribution

- a random variable is said to have a **binomial distribution** with parameters n and p if its pmf is

$$\binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}$$



- the classical use case for the binomial distribution: **repeated experiments**
    - n corresponds to the number of experiments, p to the probability of "success"
- it is the sum of n independent Bernoulli variables

overview

recap: random variables

random variables in Scipy

the cumulative distribution function

random variables in Scipy

- Scipy implements a large number of probability distributions
  - see http://docs.scipy.org/doc/scipy/reference/stats.html
- make a new r.v. representing a die:

```
die = scipy.stats.randint(1, 7)
```

- roll the die 10 times:

```
rolls = die.rvs(10)
```

- what's the probability of a 4?

```
die.pmf(4)
```

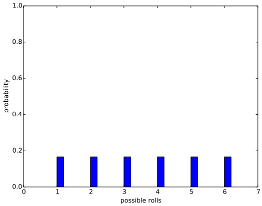
- what's the mean, variance, standard deviation?

```
die.mean()
die.var()
die.std()
```

example: plotting the pmf

- let's plot the pmf of the die roll:

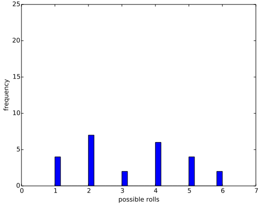
```
import scipy.stats
from matplotlib import pyplot as plt

die = scipy.stats.randint(1, 7)

possible_rolls = [1,2,3,4,5,6]

pmf_values = [ die.pmf(x) for x in possible_rolls ]

# or even shorter:
pmf_values = die.pmf(possible_rolls)

plt.bar(possible_rolls, pmf_values, width=0.2)

# some cosmetics
plt.axis([0, 7, 0, 1])
plt.xlabel('possible rolls')
plt.ylabel('probability')

plt.show()
# or plt.savefig('die_pmf.png')
```



example: plotting a histogram of die rolls

- we generate a sample and plot the histogram:

```
import scipy.stats
from matplotlib import pyplot as plt

die = scipy.stats.randint(1, 7)

n_rolls = 25

sample = die.rvs(n_rolls)

# increase the number of bins if ugly
plt.hist(sample, bins=30)

# some cosmetics
plt.axis([0, 7, 0, n_rolls])
plt.xlabel('possible rolls')
plt.ylabel('frequency')

plt.show()
# or plt.savefig('die_hist.png')
```



overview

recap: random variables

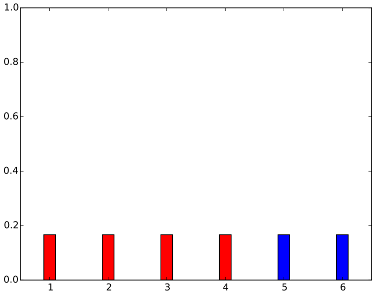
random variables in Scipy

the cumulative distribution function

recap: probabilities of intervals

- what is the probability that we roll a number that is at most 4?

$$p_X(1) + p_X(2) + p_X(3) + p_X(4) = 4 \cdot \frac{1}{6}$$



the cumulative distribution function

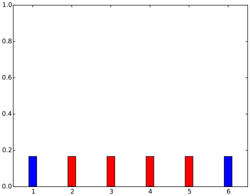
- we define the **cumulative distribution function** (cdf) of a random variable like this:

$$f_X(k) = P(X \leq k) = \sum_{i \geq k} p_X(i)$$

- what is the probability that we roll a number that is at most 4?
  - it is $f_X(4)$

- what is the probability that we roll a number that is greater than 1 but at most 5?

$$P(1 < X \leq 5) = f_X(5) - f_X(1)$$

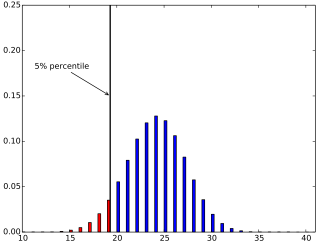


the cdf in Scipy

```
die = scipy.stats.randint(1, 7)

print(die.cdf(5) - die.cdf(1))
```

the percentiles

- just like for a sample, we can speak of **percentiles** for a r.v.
- for instance: the 5% percentile is the k such that 5% of the distribution falls below k



- in Scipy it's called ppf:

```
my_rv.ppf(0.05)
```