041_Folien

2018 年 11 月 23 日

```
In [1]: import numpy as np                      # mathematical methods
        from matplotlib import pyplot as plt     # plotting methods
        %matplotlib inline
```

0.1 Diskrete Wahrscheinlichkeitstheorie - Überblick

1. Zufall

- Zufallsvariable, Zufallsexperiment, WahrscheinlichkeitsmaSS, Wahrscheinlichkeitsverteilung, Axiome von Kolmogorov

2. Zusammengesetzte Wahrscheinlichkeiten

- bedingte Wahrscheinlichkeit, gemeinsame Wahrscheinlichkeit, totale Wahrscheinlichkeit, Satz von Bayes

3. KenngröSSen

- Erwartungswert & Varianz

4. Modellverteilungen

- Laplace-Experiment
- Bernoulli-Experiment
- Binomiale Verteilung
- Geometrische Verteilung
- Poissonverteilung

1 Laplace Experiment

- endlich viele (M) Elementarereignisse 最终 M 个元素事件发生
- alle Elementarereignisse sind gleich wahrscheinlich 所有的元素事件是等概率的

$$P(\omega_i) = p = \frac{1}{M} = \frac{1}{|\Omega|} \quad i \in \{1 \dots M\}$$

- Wahrscheinlichkeit $= \frac{\text{Anzahl für A günstige Ergebnisse}}{\text{Anzahl für A mögliche Ergebnisse}}$

$$p(A) = \frac{|A|}{|\Omega|} = \frac{N}{M}$$

Beispiele:

- Münzwurf, Würfel, Urne mit farbigen Kugeln, Roulette, Kartenspiel (Spieltheorie)
- Ziffern von π (Mathematik)

1.1 Kennzahlen

Erwartungswert

$$\mathcal{E}(X) = \mu_X = \overline{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$$

Varianz

$$\mathrm{Var}(X) = \frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x})^2$$

[ÜA] Bitte nachrechnen!

2 Bernoulli-Experiment

Ereignis A *tritt ein* oder *tritt nicht ein*

$$\Omega = \{A, \overline{A}\}$$

Binäre Zufallsvariable X (*Indikatorvariable*) kann nur Werte $\omega \in \{0, 1\}$ annehmen.

$$X = \begin{cases} 1 & \text{wenn A zutrifft} \\ 0 & \text{wenn A nicht zutrifft} \end{cases}$$

Beispiele

- Münzwurf: Kopf / Zahl
- Produktion: innerhalb Toleranz / AusschuSS
- Geburten: Mädchen / Jungen
- Psychophysik: gesehen / nicht gesehen

Unterscheidung

- MuSS nicht *Laplace* sein, es darf $p(A) \neq p(\overline{A})$

2.1 Bernoulli-Verteilung

Die Wahrscheinlickeit für A sei $\pi = \text{const.}$

$$\begin{aligned} P(A) &= P(X{=}1) &= \pi \\ P(\overline{A}) &= P(X{=}0) &= 1 - \pi \end{aligned}$$

2.2 Kennzahlen

Der Erwartungswert einer Bernoulli-Zufallsvariablen ist

$$\mathcal{E}(X) = \pi$$

Die Varianz ist

$$\mathrm{Var}(X) = \pi\,(1 - \pi)$$

[ÜA] Bitte nachrechnen!

3 Mehrfache identische, unabhängige Wiederholung eines Zufallsexperiments - *i.i.d.*

Beispiel

- Mehrfaches unabhängiges Werfen einer Münze
- Gleichzeitiges unabhängiges Werfen mehrerer identischer Münzen

Identischer Wahrscheinlichkeitsraum: Zufallsvariable, Wahrscheinlichkeistsverteilung

- *i.i.d*: **i**dentically **i**ndependent **d**istributed

4 Binomial-Verteilung

n-malige unabhängige Wiederholung des gleichen Bernoulli-Zufallsexperiments ("*i.i.d.*"): Zufallsvariable X ist jetzt die **Anzahl** der (positiven) Einzel-Ereignisse $A_i \in \{0,1\}$

$$P(A_i = 1) = \pi \quad \forall i$$

$$X = \sum_{i=1}^{n} A_i$$

4.0.1 Beispiel:

drei Münzwürfe mit ein-und-derselben fairen Münze, wie oft kommt Kopf?

```
X : Mögliche Ergebnisse           Wahrscheinlichkeit                    p=1/2
-------------------------------------------------------------------------------------
0 : {(0,0,0)}                     (1-p)(1-p)(1-p)                       p = 1/8
1 : {(1,0,0), (0,1,0), (0,0,1)}   p(1-p)(1-p)+(1-p)p(1-p)+(1-p)(1-p)p   p = 3/8
2 : {(1,1,0), (1,0,1), (0,1,1)}   p*p*(1-p)+p*(1-p)*p+(1-p)*p*p         p = 3/8
3 : {(1,1,1)}                     p^3                                   p = 1/8
```

4.1 Wahrscheinlichkeitsverteilung

$$P(X = x) = \binom{n}{x} \pi^x (1-\pi)^{(n-x)} \quad x \in \{0...n\}$$

$\binom{n}{k}$ (sprich: "*n über k*") Anzahl der Variationen von k aus n Elementen.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n \cdot (n-1) \cdot ... \cdot (n-k+1)}{k \cdot (k-1) \cdot ... \cdot 1}$$

Python:

```
In [3]: from scipy.special import comb
        n = 49
        k = 6
        nck = comb(n, k, exact=False)
        print('(n={} choose k={}) yields {}'.format(n, k, nck))
```

```
(n=49 choose k=6) yields 13983816.0
```

```
In [4]: print('binomial triangle')
        for n in range(10):
            nf = np.math.factorial(n)
            tr = [int(nf/(np.math.factorial(k)*np.math.factorial(n-k))) for k in range(n+1)]
            print('{:2d}: '.format(n), tr)
        print('...')
```

```
binomial triangle
 0:  [1]
 1:  [1, 1]
 2:  [1, 2, 1]
 3:  [1, 3, 3, 1]
 4:  [1, 4, 6, 4, 1]
 5:  [1, 5, 10, 10, 5, 1]
 6:  [1, 6, 15, 20, 15, 6, 1]
 7:  [1, 7, 21, 35, 35, 21, 7, 1]
 8:  [1, 8, 28, 56, 70, 56, 28, 8, 1]
 9:  [1, 9, 36, 84, 126, 126, 84, 36, 9, 1]
...
```

4.1.1 Normierung

$$\sum_{i=1}^{N=n+1} p_i = \sum_{x=0}^{n} \binom{n}{x} \pi^x (1-\pi)^{(n-x)} = 1$$

4.1.2 Parameter

Die Binomialverteilung $\mathcal{B}$ läSSt sich durch zwei Parameter

- n: Anzahl Versuche
- π: Wahrscheinlichkeit für das Eintreten des positiven Falles

eindeutig beschreiben: $\mathcal{B}_{(n,\pi)}$
Sie ist dann eine Funktion der Zufallsvariablen X, der Anzahl der positiven Ergebnisse:

$$\mathcal{B}_{(n,\pi)}(x) = P(X{=}x) = \binom{n}{x} \pi^x (1-\pi)^{(n-x)} \quad x \in \{0...n\}$$

4.2 Kennzahlen

Der **Erwartungswert** einer Binomial-verteilten Zufallsvariablen X ist

$$\mathcal{E}(X) = n \cdot \pi$$

Die **Varianz** ist

$$\mathrm{Var}(X) = n \cdot \pi \cdot (1-\pi)$$

4.3 Python

Statistik-Paket `scipy.stats`

- Referenz http://docs.scipy.org/doc/scipy/reference/stats.html
- open source unter BSD Lizenz https://www.scipy.org/scipylib/license.html
- gehostet auf https://github.com/scipy/scipy
- Version `SciPy 1.1.0 released 2018-05-05`
- verwenden:

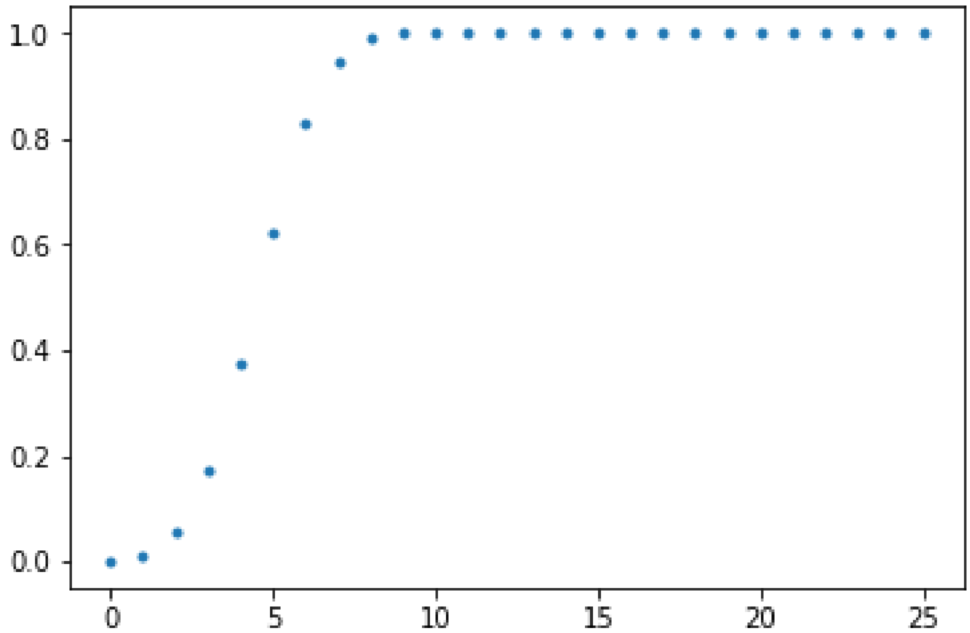
```
In [5]: '''statistical methods'''
        from scipy import stats

        stats.binom?
```

```
In [6]: '''binomial distribution - an example'''
        n = 10         # number of repetitions
        p = .5         # probability pi of single event
        x = 5          # one example result: number of successes
        print('probability of x={} as number of "positive" events is Binomial({}, {}, {}) = {:.5f}'.
              format(x, x, n, p, stats.binom(n, p).pmf(x)))
        print('expectation value of Binomial({}, {}) = {:.5f}'.
              format(n, p, stats.binom(n, p).expect()))
```

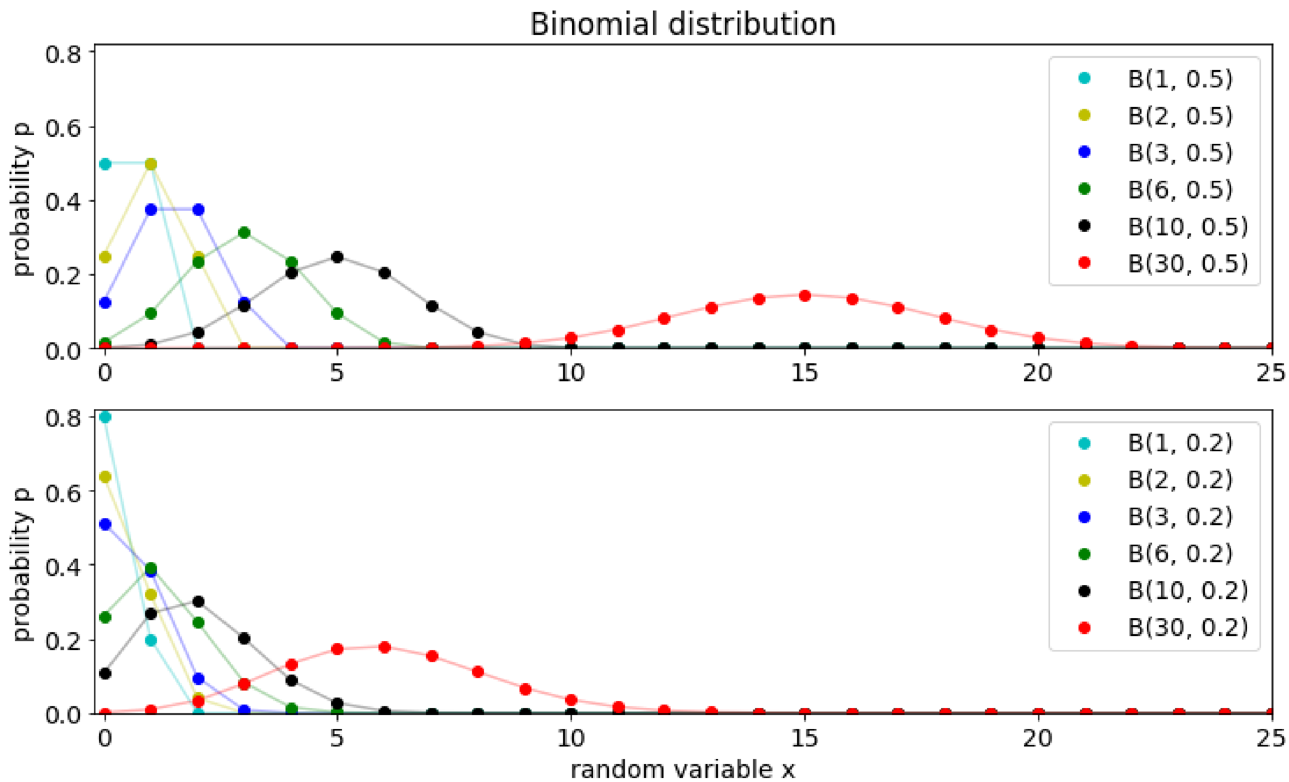
```
probability of x=5 as number of "positive" events is Binomial(5, 10, 0.5) = 0.24609
expectation value of Binomial(10, 0.5) = 5.00000
```

```
In [3]: from scipy import stats
        from matplotlib import pyplot as plt
        import numpy as np

        x = np.arange(26)

        plt.plot(stats.binom(10, .5).cdf(x), '.')
```

```
Out[3]: [<matplotlib.lines.Line2D at 0x7f0f2c6392b0>]
```




```
In [7]: '''Binomial probability distributions'''
        xs = np.arange(26)                          # number of successes x
        p = 0.5                                     # probability pi of 1st examples
        colors = ['c', 'y', 'b', 'g', 'k', 'r']     # cycle through colors
        ns = [1, 2, 3, 6, 10, 30]                   # variety of n to show

        f = plt.figure(figsize=(12, 7))             # big canvas
        f.add_subplot(2, 1, 1)                      # top figure for 1st examples
        plt.title('Binomial distribution')
        plt.ylabel('probability p')
        plt.axis((-.2, 25., 0., 0.82))
        for i, n in enumerate([1, 2, 3, 6, 10, 30]):
            ps = stats.binom(n, p).pmf(xs)
            plt.plot(xs, ps, colors[i]+'o', label='B({}, {})'.format(n,p))
            plt.plot(xs, ps, colors[i]+'-', alpha=.3)
        plt.legend()
        f.add_subplot(2, 1, 2)                      # bottom figure for 2nd examples
        plt.ylabel('probability p')
        plt.xlabel('random variable x')
        p = 0.2                                     # probability pi of 2nd examples
        plt.axis((-.2, 25., 0., 0.82))
        for i, n in enumerate([1, 2, 3, 6, 10, 30]):
            ps = stats.binom(n, p).pmf(xs)
            plt.plot(xs, ps, colors[i]+'o', label='B({}, {})'.format(n,p))
            plt.plot(xs, ps, colors[i]+'-', alpha=.3)
        plt.legend();
```



4.4 Eigenschaften

Addition Sind $X \sim \mathcal{B}(n, \pi)$ und $Y \sim \mathcal{B}(m, \pi)$ Binomial-verteilt und voneinander unabhängig, so ist

$$X + Y \sim \mathcal{B}(n + m, \pi)$$

Symmetrie Sei $X \sim \mathcal{B}(n, \pi)$, dann gilt für $Z = n - X$

$$Z \sim \mathcal{B}(n, 1 - \pi)$$

5 Geometrische Verteilung

n-malige unabhängige Wiederholung (*i.i.d.*) des gleichen Bernoulli-Zufallsexperiments: Zufallsvariable X ist Anzahl der Versuche mit Wahrscheinlichkeit π, die nötig sind **bis das erste Mal** das positive Ereignis $A_i \in \{0, 1\}$ eintritt.

$$P(A_i = 1) = \pi \quad \forall i$$

$$X = n \quad \text{für} \quad A_{i<n} = 0\,,\; A_n = 1$$

Beispiel Rechnerabstürze Sei p die Wahrscheinlichkeit, daSS ein PC abstürzt. Wie lange kann man im Mittel arbeiten?

Das Ereignis könnte so aussehen

```
Stunden   1  2  3  4  5  6  7  8  9
Absturz   0  0  0  0  0  1
```

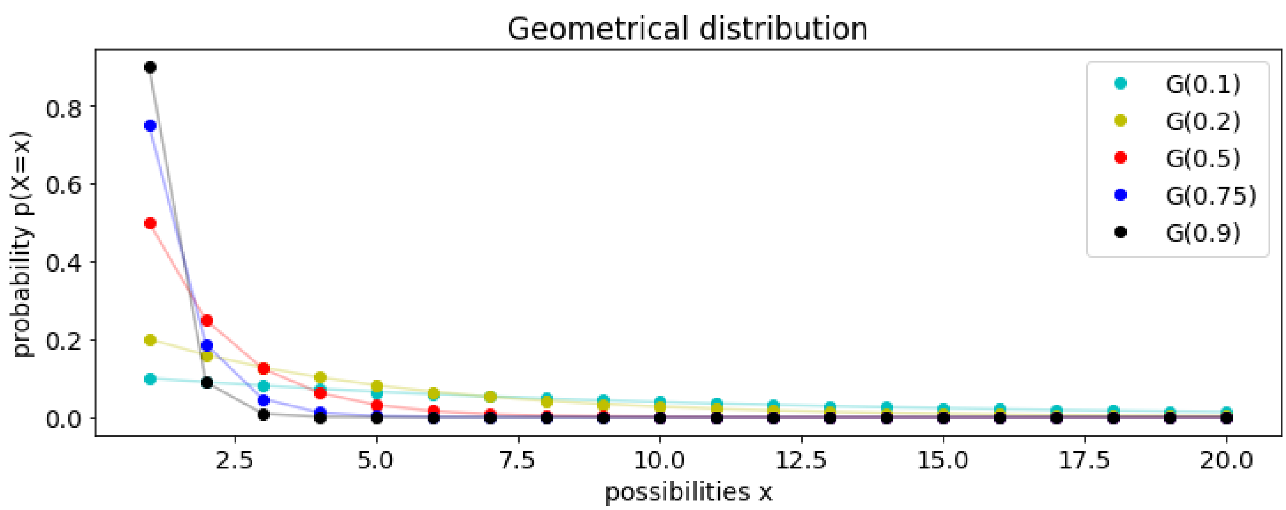
- x = 6

- 5h kein Absturz mit jeweils Wahrscheinlichkeit $(1-\pi)$ also $p = (1-\pi)^5$
- 1h der Absturz mit Wahrscheinlichkeit π
- Gesamtwahrscheinlichkeit $p_6 = (1-\pi)^5 \cdot \pi$

5.1 Wahrscheinlichkeitsverteilung

$$P(X{=}x) = (1-\pi)^{x-1} \cdot \pi \quad x \in \{1...n\}$$

In [8]:
```
'''Geometric probability distributions'''
f = plt.figure(figsize=(12, 4))
plt.title('Geometrical distribution')
plt.xlabel('possibilities x')
plt.ylabel('probability p(X=x)')
colors = ['c', 'y', 'r', 'b', 'k', 'r']
xs = np.arange(1., 21)
for i, pi in enumerate([.1, .2, .5, .75, .9]):
    ps = stats.geom(pi).pmf(xs)
    plt.plot(xs, ps, colors[i]+'o', label='G({})'.format(pi))
    plt.plot(xs, ps, colors[i]+'-', alpha=.3)
plt.legend();
```



5.1.1 Normierung:

Die Geometrische Wahrscheinlichkeitsverteilung ist normiert:

$$\sum_{i=1}^{\infty} p_i = \sum_{x=0}^{\infty} (1-\pi)^{(x-1)} \cdot \pi = 1$$

In [9]:
```
'''is geometrical distribution normalized? Not as a proof but ... '''
pi = 0.4                        # probability as an example
nmax = 24                       #
p_sum = 0.                      # sum all probabilities
for x in np.arange(1, nmax):
```

```
        p = pi * (1-pi)**(x-1)
        p_sum += p
        print('x={:2d} has p={:.5f} which sums up to {:.5f}'
              .format(x, p, p_sum))
```

```
x= 1 has p=0.40000 which sums up to 0.40000
x= 2 has p=0.24000 which sums up to 0.64000
x= 3 has p=0.14400 which sums up to 0.78400
x= 4 has p=0.08640 which sums up to 0.87040
x= 5 has p=0.05184 which sums up to 0.92224
x= 6 has p=0.03110 which sums up to 0.95334
x= 7 has p=0.01866 which sums up to 0.97201
x= 8 has p=0.01120 which sums up to 0.98320
x= 9 has p=0.00672 which sums up to 0.98992
x=10 has p=0.00403 which sums up to 0.99395
x=11 has p=0.00242 which sums up to 0.99637
x=12 has p=0.00145 which sums up to 0.99782
x=13 has p=0.00087 which sums up to 0.99869
x=14 has p=0.00052 which sums up to 0.99922
x=15 has p=0.00031 which sums up to 0.99953
x=16 has p=0.00019 which sums up to 0.99972
x=17 has p=0.00011 which sums up to 0.99983
x=18 has p=0.00007 which sums up to 0.99990
x=19 has p=0.00004 which sums up to 0.99994
x=20 has p=0.00002 which sums up to 0.99996
x=21 has p=0.00001 which sums up to 0.99998
x=22 has p=0.00001 which sums up to 0.99999
x=23 has p=0.00001 which sums up to 0.99999
```

5.2 Kennzahlen

Der **Erwartungswert** einer geometrisch-verteilten Zufallsvariablen X ist

$$\mathcal{E}(X) = \frac{1}{\pi}$$

Die **Varianz** ist

$$\mathrm{Var}(X) = \frac{1-\pi}{\pi^2}$$

5.2.1 Für das Beispiel der Rechnerabstürze ergibt sich

- ein Erwartungswert von 5 Stunden
- bei einer Streuung von $\sqrt{\frac{0.8}{0.2^2}} = 4,5$ Stunden

6 Poisson-Verteilung

Die Anzahl von seltenen Ereignissen innerhalb eines festen Intervalls werden durch die Poisson-Statistik beschrieben. $X_i \in \{0, 1, 2, ...\}$

Voraussetzungen:

- kein gleichzeitiges Eintreten zweier Ereignisse
- Wahrscheinlichkeit für Ereignis in kleinem Intervall Δt ist $\lambda \cdot \Delta t$
  - λ ist eine Intensitätsrate
  - Ereignisse sind selten
- Wahrscheinlichkeit für $X_i = x$ ist in jedem beliebigen Intervall gleicher Länge gleich

6.1 Wahrscheinlichkeitsverteilung

$$P(X=x) = \frac{\lambda^x}{x!}e^{-\lambda}$$

- Einziger Parameter der Poissonverteilung ist die Rate λ: $\mathcal{P}\wr(\lambda)$
- Funktion von x, der Anzahl der Ereignisse im betrachteten Intervall

6.2 Kennzahlen

Der **Erwartungswert** einer Poisson-verteilten Zufallsvariablen $X \sim \mathcal{P}\wr(\lambda)$ ist

$$\mathcal{E}(X) = \lambda$$

Die **Varianz** ist

$$\mathrm{Var}(X) = \lambda$$

Beispiel Sternschnuppen Sei $\lambda = 0.2$ die Wahrscheinlichkeit eine Sternschnuppe pro Minute zu sehen. Wie lange muSS man im Mittel warten um eine zu sehen?

Das Ereignis könnte so aussehen

```
Minute    1  2  3  4  5  6  7  8  9 10
Schnuppe  0  0  0  1  0  0  0  0  1  0
```

6.2.1 Für das Beispiel der Sternschnuppen ergibt sich

- ein Erwartungswert von 0.2 je Minute
- bei einer Streuung von 0.2 je Minute

6.3 Eigenschaften

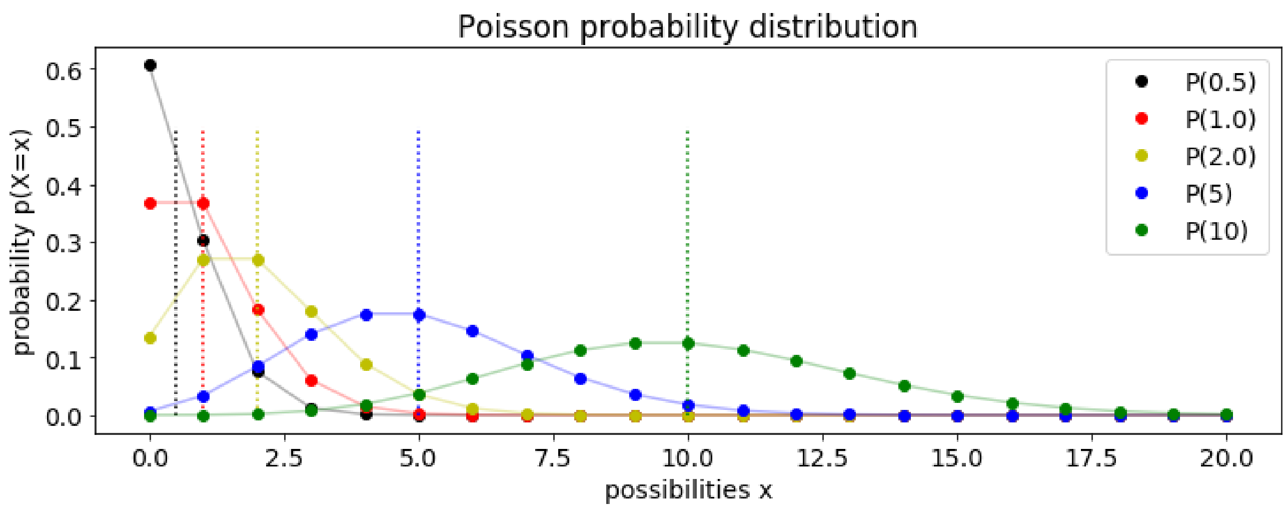
Intervallstreckung Sei $X \sim \mathcal{P}\wr(\lambda)$ auf dem Einheitsintervall, dann ist Z im Intervall der Länge l Poissonverteilt mit

$$Z \sim \mathcal{P}\wr(l \cdot \lambda)$$

6.4 Wahrscheinlichkeitsverteilung

In [10]:
```
'''Poisson probability distributions'''
f = plt.figure(figsize=(12, 4))
plt.title('Poisson probability distribution')
plt.xlabel('possibilities x')
plt.ylabel('probability p(X=x)')
```

```
colors = ['k', 'r', 'y', 'b', 'g', 'r']
xs = np.arange(0., 21)
for i, lbda in enumerate([.5, 1., 2., 5, 10]):
    ps = stats.poisson(lbda).pmf(xs)
    plt.plot(xs, ps, colors[i]+'o', label='P({})'.format(lbda))         # dots at p
    plt.plot(xs, ps, colors[i]+'-', alpha=.3)                            # connecting line
    plt.plot(2*[stats.poisson(lbda).expect()], [0, 0.5], colors[i]+':') # expectation
plt.legend();
```



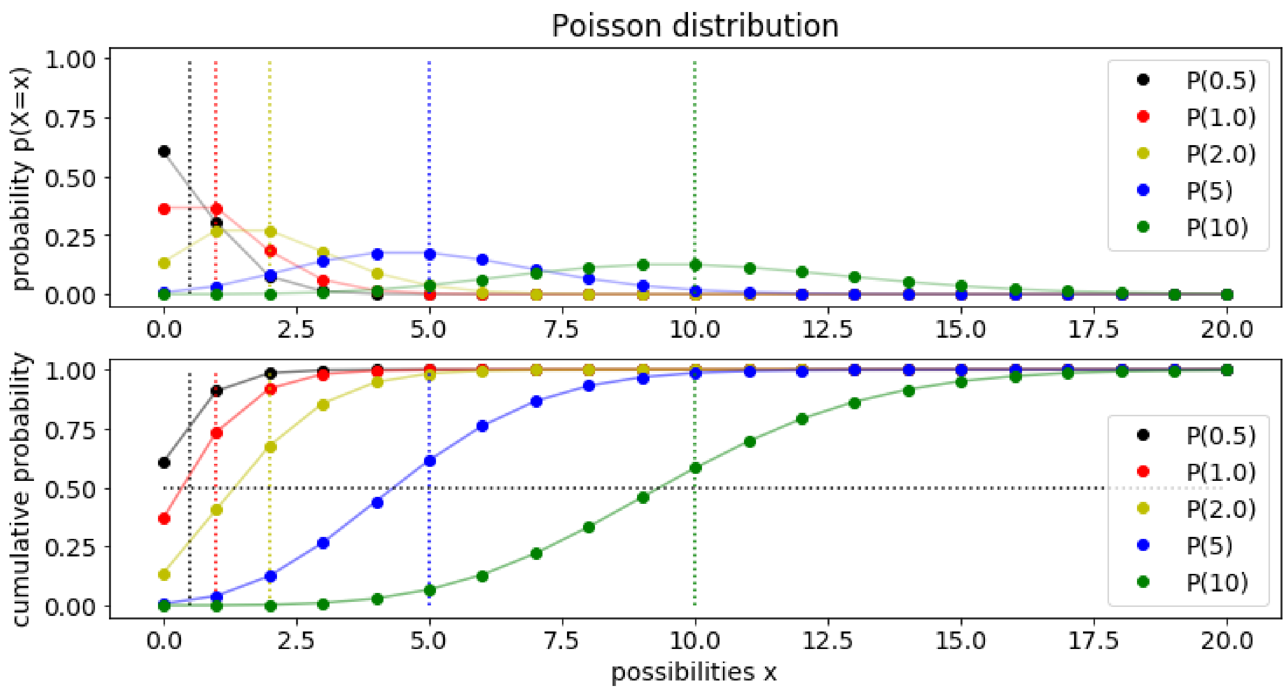
6.5 Kumulierte Verteilungsfunktion

$$F(x) = \sum_{i=1}^{j} p(x_i) \qquad j = max(i \mid x_i <= x)$$

$$F(x) = \sum_{x_i <= x} p(x_i)$$

```
In [11]: '''Poisson probability distributions'''
         f = plt.figure(figsize=(12, 6))
         f.add_subplot(2, 1, 1)
         plt.title('Poisson distribution')
         plt.ylabel('probability p(X=x)')
         colors = ['k', 'r', 'y', 'b', 'g', 'r']
         xs = np.arange(0., 21)
         for i, lbda in enumerate([.5, 1., 2., 5, 10]):
             ps = stats.poisson(lbda).pmf(xs)
             plt.plot(xs, ps, colors[i]+'o', label='P({})'.format(lbda))         # dots at p
             plt.plot(xs, ps, colors[i]+'-', alpha=.3)                            # connecting line
             plt.plot(2*[stats.poisson(lbda).expect()], [0, 1.0], colors[i]+':') # expectation
         plt.legend()
         f.add_subplot(2, 1, 2)
```

```
plt.xlabel('possibilities x')
plt.ylabel('cumulative probability')
colors = ['k', 'r', 'y', 'b', 'g', 'r']
x = np.arange(0., 21)
for i, lbda in enumerate([.5, 1., 2., 5, 10]):
    ps = stats.poisson.cdf(x, lbda)
    plt.plot(x, ps, colors[i]+'o', label='P({})'.format(lbda))
    plt.plot(x, ps, colors[i]+'-', alpha=.5)
    plt.plot(2*[stats.poisson(lbda).expect()], [0, 1.0], colors[i]+':')
plt.plot([0, 20], 2*[.5], 'k:')
plt.legend(loc='lower right');
```



---

7 Diskrete Verteilungen - Zusammenfassung

7.0.1 Voraussetzungen

i.i.d. Unabhängige identische Wiederholung

- der selbe Wahrscheinlichkeitsraum

Bernoulli-Experiment

- zwei (dichotome) Fälle $A, \overline{A}$
- Wahrscheinlichkeiten

$$P(X=A) = \pi \qquad P(X=\overline{A}) = 1 - \pi$$

7.0.2 Binomialverteilung

- n Bernoulli-Experimente
- Zufallsvariable ist Anzahl der *Erfolge* $x \in [0 \dots n]$
- Parameter der Binomialverteilung:
  - n = Anzahl der durchgeführten Wiederholungen
  - π = Wahrscheinlichkeit des Bernoulli-Experiments

7.0.3 Geometrische Verteilung

- Bernoulli-Experimente bis zum ersten Eintreten von *Erfolg*
- Zufallsvariable ist Anzahl der Versuche $x \in [0 \dots]$
- Parameter der Geometrischen Verteilung
  - π = Wahrscheinlichkeit des Bernoulli-Experiments

7.0.4 Poissonverteilung

- Seltene Ereignisse in festen Intervallen
- Zufallsvariable ist Anzahl der Beobachtungen $x \in [0 \dots]$
- Parameter der Poissonverteilung
  - Rate λ

8 Zusammenfassung Python

Statsitik-Bibliothek `scipy.stats` enthält diskrete Verteilungen

```
bernoulli(p)
binom(n, p)
geom(p)
poisson(mu)       # "lambda" ist reserviertes Schlüsselwort

np.random.choice  # wie .rvs() für Laplace-Verteilungen - und beliebig andere
```

Funktionen und Methoden

```
.expect()         # Erwartungswert einer Verteilung
.pmf(x)           # Wahrscheinlichkeitsverteilung "probability mass function"
.cdf(x)           # Verteilungsfunktion "cumulative distribution function"
.rvs()            # Zufallsererergebnis "random variables"
                  # (optional Anzahl der Werte)
```

```
In [12]: '''passing parameters to scipy.stats functions, example binomial'''
         num = 8
         prob = 0.50
         x = [0, 1, 2, 3]
         ps1 = stats.binom(num, prob).pmf(x)         # two parameters, one (list of) value(s)
         ps2 = stats.binom(n=num, p=prob).pmf(x)     # my preferred syntax
         ps3 = stats.binom(p=prob, n=num).pmf(x)     # no problem if exchanged
         ps4 = stats.binom.pmf(x, num, prob)         # often found, need to know order
```

```
    ps5 = stats.binom.pmf(x, n=num, p=prob)      # python-like with key word list
    for psi in [ps2, ps3, ps4, ps5]:
        if (psi == ps1).all():
            print('identical')
        else:
            print('{} not identical with {}'.format(psi, ps1))
```

```
identical
identical
identical
identical
```

```
In [13]: '''freeze a function'''
         bdistrib = stats.binom(n=num, p=prob)    # from now on includes parameters num and prob
         print(bdistrib)                          # object?
         print(bdistrib.rvs(5))                   # get random variables
         print(bdistrib.pmf(x))                   #     or  probability  from binomial(parameters)
```

```
<scipy.stats._distn_infrastructure.rv_frozen object at 0x7fc3199a4cf8>
[4 6 7 1 2]
[0.00390625 0.03125    0.109375   0.21875    ]
```

9 Ausblick

- mehrdimensionale Verteilungen
- Kontinuierliche Verteilungen

10 Fragen?