Speech Processing Overview & Supporting Concepts

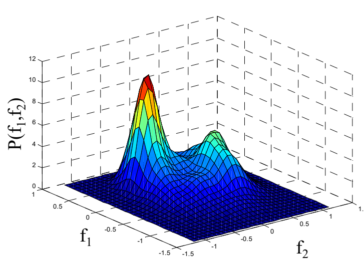
Professor Marie Roch

Readings are listed in the schedule

SAN DIEGO STATE UNIVERSITY

What is speech processing?

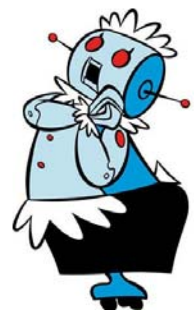
- Specialized branch of machine learning
- Interdisciplinary
  - signal processing
  - statistics
  - linear algebra
  - linguistics
  - optimization
  - perception & production



SAN DIEGO STATE UNIVERSITY

Basic ideas

- Acquire an audio signal
- Extract features
- Classify features to labels depending on goals, e.g.
  - text to speech
  - identify a person
  - recognize directives



Rosie the robot, The Jetsons © Hannah Barbera

SAN DIEGO STATE UNIVERSITY

Probability

- A measure of uncertainty.
- Sources of uncertainty:
  - Process is stochastic (nondeterministic).
  - We cannot observe all aspects of process.
  - Incomplete modeling of process.

SAN DIEGO STATE UNIVERSITY

Probability

Two ways to think about:

- frequentist – Related to the proportion of events occurring
- Bayesian – Related to degree of belief

Both types of probability are treated using the same rules.

Random variables

- Variables that can be bound to values non-deterministically.
- Common to use notation to distinguish between*
  - a random variable X (capitalized)
  - and an observed value x (lower case, e.g. x=5)

*Goodfellow et al. use bold in place of capitalization.

Discrete random variables

- Values from a discrete set of labels:

  $X \in \{\text{red, orange, yellow, blue, indigo, violet}\}, Y \in \{0,1,2\}$

- Probability density (mass) function (pdf/pmf) is the probability of something occurring, P(X=x)*:

  $P(X = red) = 0.8, P(Y = 2) = \frac{1}{3}$

- $\forall x \in \text{domain}(X), 0 \le P(X = x) \le 1 \text{ and } \sum_{x \in \text{domain}(X)} P(X = x) = 1$

*Goodfellow et al. use PMF for discrete distributions. P(X=x) is commonly abbreviated to P(x)

Distribution

- Set of probabilities associated with the values of a random variable.
- Uniform distribution – Special distribution where all probabilities are equal.

Example

- Let X be a die roll
  - P(X=3) denotes the probability of rolling a 3
  - Fair die → P(X=3) = 1/6

- If we don't know if the die is fair, how would we approximate P(X=x)?

- Estimates of probability are denoted with a hat:

$$\hat{P}(X = x)$$

SAN DIEGO STATE UNIVERSITY

Continuous random variables

- Values from a continuous domain
- Satisfies the following:

$$\forall x \in \text{domain}(X), 0 \le P(X = x)$$

$$P(a \le X \le b) = \int_a^b P(X = x)dx$$

$$\int_x P(X = x)dx = 1$$

- Worth noting:
  - no max on P(X=x)
  - What is? $P(a \le X \le a) = \int_a^a P(X = x)dx$

SAN DIEGO STATE UNIVERSITY

Continuous uniform distribution

$$X \sim U(a,b)$$

$$\text{implies } P(X = x) = \begin{cases} 0 & x < a \text{ or } x > b \\ \dfrac{1}{b-a} & a \le x \le b \end{cases}$$

In general, ~ means "has a distribution of."
U(a,b) is a uniform distribution between values and b

Expectation

discrete: $E[u(X)] = \sum_x u(x)\,\mathrm{P}(x)$

continuous: $E[u(X)] = \int_x u(x)\,\mathrm{P}(x)dx$

When u(X) = X, we call this the mean, or average value:

$$\mu = E[X]$$

Sample mean $\hat{\mu}$

- What if we don't know P(X=x)?
- If we roll a die many times, we can estimate P(X=x) by counting the number of times each x occurs and dividing by the number of rolls.

- To think about: How does this relate to our traditional idea of an average?

Variance

- Answers the question: What is the expected squared deviation from the mean?

- Can be defined with expectation operator

$$E[(X-\mu)^2] = \sum_x (x-\mu)^2 P(X=x)$$

- Sample variance (unbiased)

$$\hat{E}[(x-\mu)^2] = \frac{1}{N-1}\sum_x (x-\mu)^2$$

Normal distribution

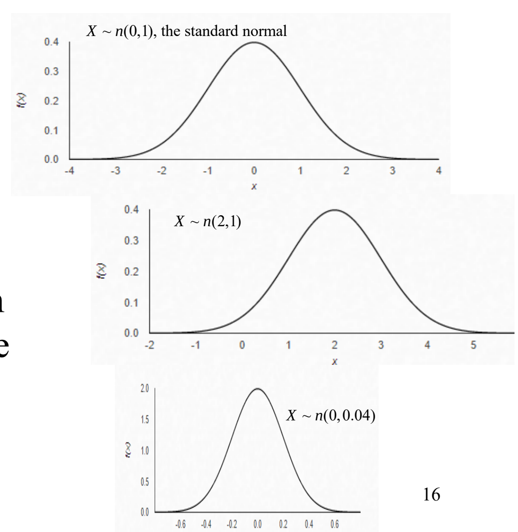
- Normal or Gaussian distributions are the so-called “bell curve” distributions
- Parameters: mean & variance: $X \sim n(\mu, \sigma^2)$

$$P(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Normal distribution

- Mean controls the center, variance the breadth of the distribution.

- A normal distribution can be fit with sample mean & variance



Application: Speech/noise segmentation

How can we determine if someone is talking?

1. Record: Acquire pressure signal
2. Frame: Break into pieces
3. Compute features: root mean square intensity
4. Train: Fit speech/noise distribution models
5. Classify: See which distribution matches new samples

SAN DIEGO STATE UNIVERSITY

17

Acquisition: pressure sensors

Basic ideas for microphones:

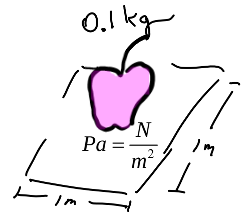
- Deformable membrane
  - pushed by compression
  - pulled by rarefaction
- Deformation is converted to voltage
- Samples: voltage measured N times/second and discretized
  - called sample rate (Fs) and
  - measured in Hertz (Hz)

SAN DIEGO STATE UNIVERSITY

18

Pressure and Intensity

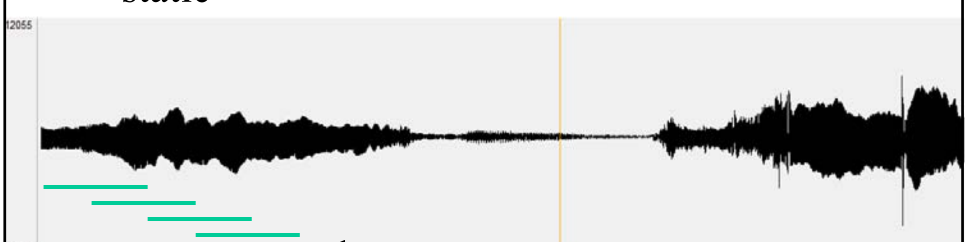
- Pressure
  - Amount of force per area
  - Typically measured in Pascals (Newtons/meter squared) if the microphone is calibrated otherwise in counts



- Intensity
  - Product of sound pressure and particle velocity per area
  - Pressure$^2 \alpha$ Intensity

Framing

- Most audio signals vary over time
- Analyze small segments that are roughly static



and so on…

chimpanzee pant hoot – courtesy Cat Hobaiter

decibel (dB) scale

- Human auditory sensitivity to *pressure*:

  20 μPa – 200 Pa (10[7] range!)
- The <u>deci</u>bel scale allows us to compare the *intensity* between two sounds in a compressed range:
  - Intensity $10\log_{10}\left(\dfrac{I}{I_0}\right)$
  - ~ with pressure $10\log_{10}\left(\dfrac{P}{P_0}\right)^2 = 20\log_{10}\left(\dfrac{P}{P_0}\right)$

What value of P_0?

calibrated terrestrial systems:

$P_0 = 20\mu\text{Pa}$ (threshold of hearing)

denoted dB re 20 μPa

uncalibrated systems:

$P_0 = 1$ (for convenience)

denoted dB rel.

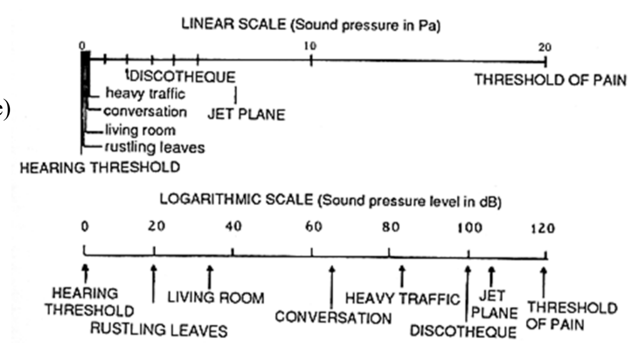


Figure 2.4 Intensities of some different sounds given in linear and logarithmic measures. (From Spens, 1970.)

Rabiner/Juang 1993

Computing intensity

- Computed for each frame
- Root-mean-square (RMS) pressure
  - squared
  - averaged
  - square root

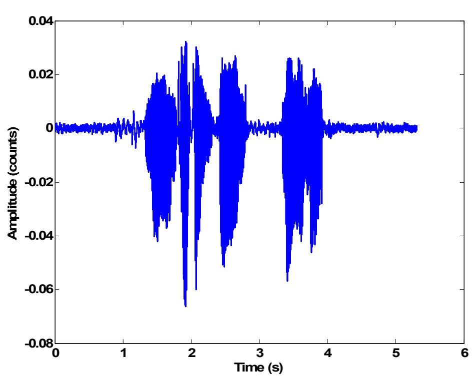
$$p_{RMS} = \sqrt{\frac{1}{|\text{frame}|} \sum_{i \in \text{frame}} x_i^2}$$

- Pressure converted to intensity in dB:

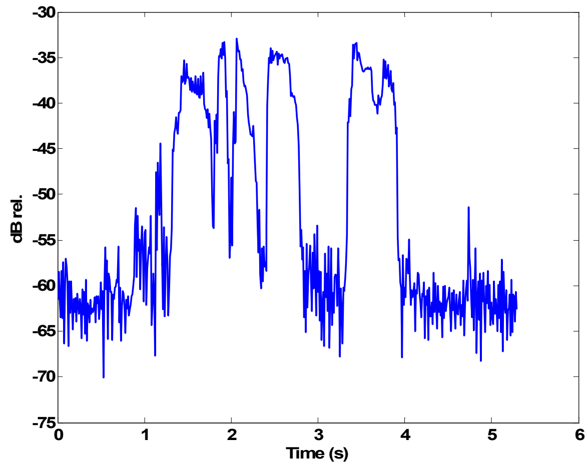
$$20 \log_{10}(p_{RMS}) = 10 \log_{10}\left(\frac{1}{|\text{frame}|} \sum_{i \in \text{frame}} x_i^2\right) \text{ dB}$$

Our first speech problem

- Given a recording, separate speech from constant background noise

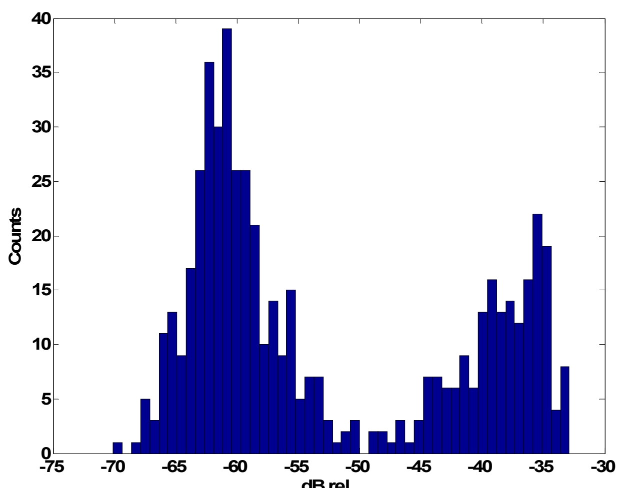


Relative Intensity



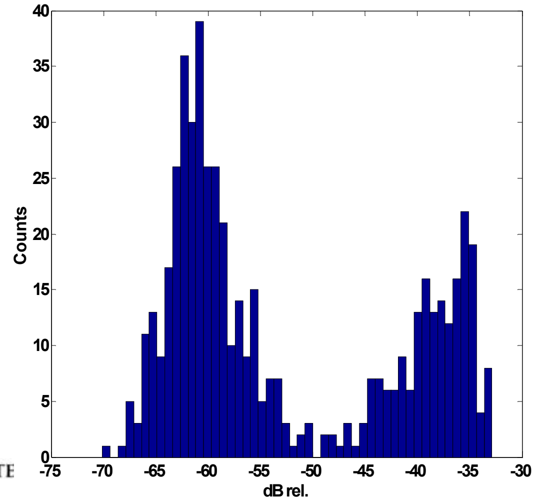
SAN DIEGO STATE UNIVERSITY

Intensity histogram



SAN DIEGO STATE UNIVERSITY

So how can we characterize speech and silence?

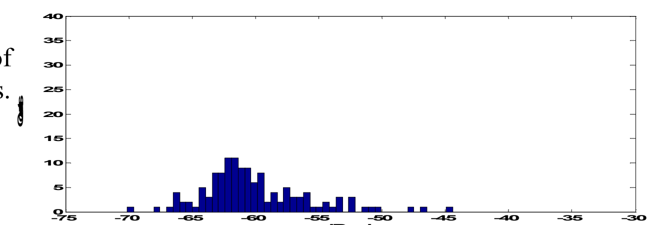


San Diego State University

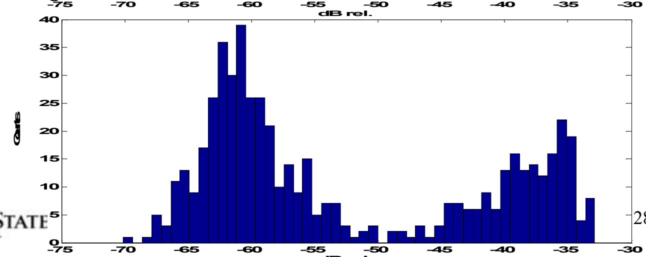
27

Silence estimator

- Beginning of recording vs.



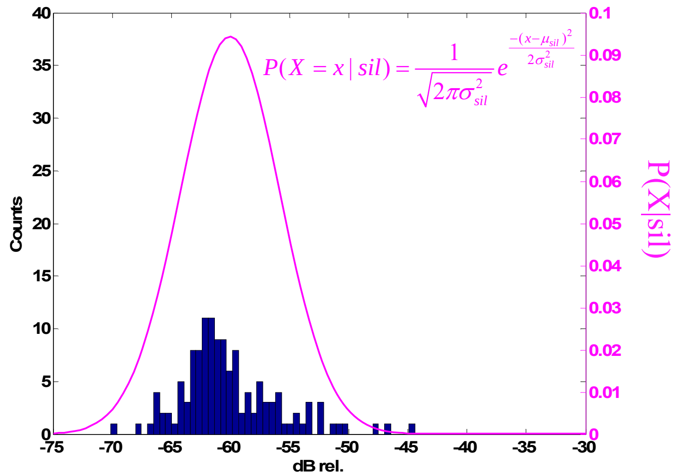
- Entire recording



San Diego State University

28

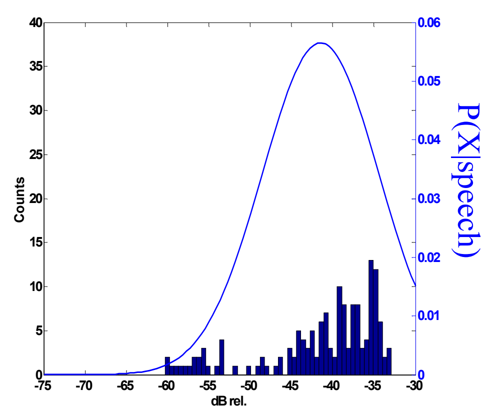
Normal fit of silence



SAN DIEGO STATE UNIVERSITY

29

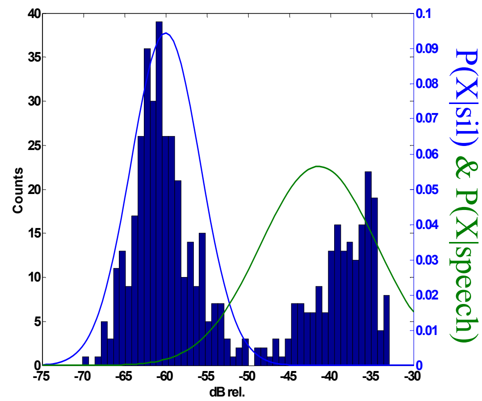
Similar for speech 1.3 – 2.8 s



SAN DIEGO STATE UNIVERSITY

30

Intensity distribution with pdfs



Where should we draw the boundary?

Conditional probability

The probability of something occurring can change when you know something…

$$P(X = \text{"nice to meet you"})$$

vs

$$P(X = \text{"nice to meet you"} \mid \text{meeting new person})$$

Conditional probability

The probability of something occurring can change when you know something…

$$P(X = \text{"nice to meet you"})$$

vs

$$P(X = \text{"nice to meet you"} \mid \text{meeting new person})$$

Formally, P(X|Y) defined as:

$$P(X \mid Y) \triangleq \frac{P(X,Y)}{P(Y)}$$

SAN DIEGO STATE UNIVERSITY

33

A problem…

Our intensity pdfs showed

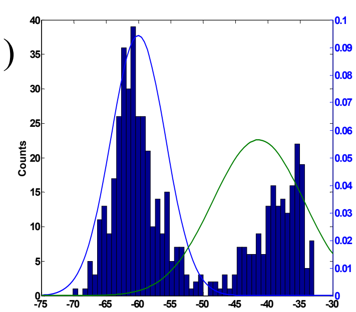
$$P(X \mid speech) \text{ and } P(X \mid noise)$$

but we would like to know

$$P(speech \mid X) \text{ and } P(noise \mid X)$$

which is known as the

posterior distribution



SAN DIEGO STATE UNIVERSITY

Reverend Thomas Bayes
1702-1761

Bayes' decision rule

The optimal classification rule is the one that maximizes the posterior probability:

$$\arg\max_{\omega \in \{speech, noise\}} P(\omega \mid X)$$

Regrettably, we do not know $P(\omega \mid X)$,
but we do know $P(X \mid \omega)$

note: arg returns the argument (ω) rather than the value

Bayes' rule

(of conditional probability)

$$P(A \mid B) = \frac{P(A, B)}{P(B)} \triangleq \text{ conditional probability}$$

note that swapping names A and B

$$\text{P(B|A)}=\frac{P(B, A)}{P(A)} \rightarrow P(B, A) = P(B \mid A) P(A)$$

and $P(A, B) = P(B, A)$

$$\therefore P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

Bayes' Rule
≠Bayes' Decision Rule

also known as Bayes' Theorem/Bayes' Law)

Bayes' decision rule

Bayes' rule to the rescue!

class-conditional probability

prior probability

$$\arg\max_{\omega} P(\omega \mid X) = \frac{P(X \mid \omega)P(\omega)}{P(X)}$$

posterior probability

observation probability

SAN DIEGO STATE UNIVERSITY

37

almost there…

$$P(\omega \mid X) = \frac{P(X \mid \omega)P(\omega)}{P(X)}$$

We know P(X|ω). We don't need to know P(X) as

$$\max\left(\frac{P(X \mid speech)P(speech)}{P(X)}, \frac{P(X \mid noise)P(noise)}{P(X)}\right)$$

$$= \max\left(P(X \mid speech)P(speech), P(X \mid noise)P(noise)\right)$$

SAN DIEGO STATE UNIVERSITY

38

Prior probability

- Probability of an observation without any prior information.
- When we don't know this, we frequently assume a uniform (equally likely) distribution:

$$= \max\left(P(X \mid speech)\frac{1}{2}, P(X \mid noise)\frac{1}{2} \right)$$

SAN DIEGO STATE UNIVERSITY

Bayes decision rule revisited

Assuming a uniform prior, we can now make decisions about speech or noise:

$$\text{decision}(x) = \underset{\omega \in \{speech, noise\}}{\arg\max} \frac{1}{\sqrt{2\pi\sigma_\omega^2}} e^{\frac{-(x-\mu_\omega)^2}{2\sigma_\omega^2}}$$

How can we solve the boundary threshold?

SAN DIEGO STATE UNIVERSITY

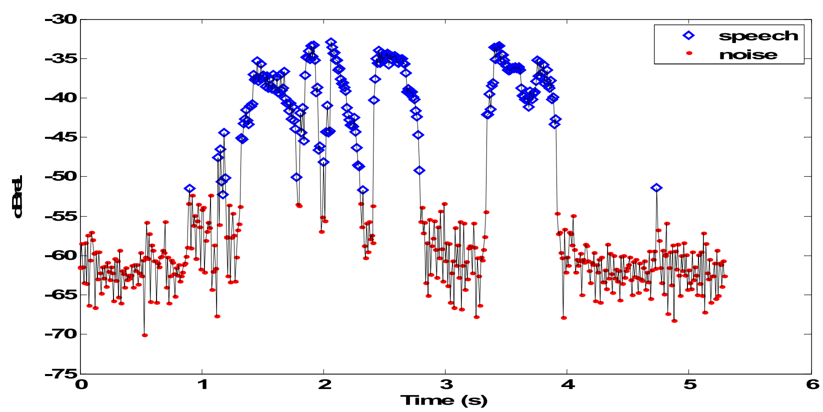
Threshold for our minimal speech detector

Solve for x (2 roots):

$$\frac{1}{\sqrt{2\pi\sigma_{noise}^2}} e^{\frac{-(x-\mu_{noise})^2}{2\sigma_{noise}^2}} = \frac{1}{\sqrt{2\pi\sigma_{speech}^2}} e^{\frac{-(x-\mu_{speech})^2}{2\sigma_{speech}^2}}$$

SAN DIEGO STATE UNIVERSITY

Result



Caveat: Example is for pedagogical purposes, we can do better than this…

SAN DIEGO STATE UNIVERSITY

A brief introduction to Python

- Object oriented language
- Weak typing system, variables do not need to be declared
- Nice language for rapid prototyping
- Rich set of standard libraries and add-ons

San Diego State University

43

Basics

- Indentation is part of the language
- Commands end at line feed unless continuation character present: \
- From # to end of line is a comment

- Be sure to comment your code!

San Diego State University

44

Basic types

- boolean: True/False
- int, float, complex
- string ‘hi there’ or “hi there”
- sequences:
  - lists: [4, 5, 10, “alpha”]
  - tuples: (4, 5, 10, “alpha”)

  Tuples are static, you cannot append, delete, etc.

SAN DIEGO STATE UNIVERSITY

45

Basic types

- Sets: set([3,4,3,5]) → {3, 4, 5}
- Dictionaries (hash tables)
  - >>> mydict = {} # empty dictionary
  - >>> mydict["hello"] = "hola"
  - >>> mydict["goodbye"] = "adios"

- >>> mydict
- {'hello': 'hola', 'goodbye': 'adios'}
- >>> mydict['goodbye']
- 'adios'
- >>> mydict['restroom']
- Traceback (most recent call last):
- File "<stdin>", line 1, in <module>
- KeyError: 'restroom'

SAN DIEGO STATE UNIVERSITY

46

Loops

for x in range(10):
        print(x) # 0:9
for x in range(3,10):
        print(x) # 3:9

for supports iterators, e.g. iterating over a list variable

x = 3
while x < 10:
        print(x)

Loops support:
- continue
- break

SAN DIEGO STATE UNIVERSITY

Functions

```
def classify(model, examples):
      '''classify – Classify each example
      model is classifier, examples are list
      of feature lists/tuples
      '''
      labels = []  # empty label list
      # predict class and store result
      for ex in examples:
            label = model.predict(ex)
            labels.append(label)
      return labels
```

SAN DIEGO STATE UNIVERSITY

Objects

```
class classifier:
      def __init__(self):
            ''' documentation string
            '''
            # constructor code
      def train(self, examples, labels):
            # code to train, e.g.
            self.model = …
      def predict(self, example)
            return self.model.predict(example)
```

Note: self is declared explicitly
Equivalent to this in C++/Java

Objects

- Objects support class and instance variables, we only showed instance variables (look it up)
- Private methods, use __name
- Special methods, use __name__

Objects

- Some special methods
  - __iter__(self) – Returns an iterator object. If the object can do iteration, sometimes just returns self
  - __next__(self) – Returns the next item. When there are no more items, executes "raise StopIteration"

Iterator example

```
class Fib:
    '''iterator that yields numbers in the Fibonacci sequence, series where next number is sum
      of the previous two'''

    def __init__(self, max):
        self.max = max      # stop when next Fibonacci number exceeds this

    def __iter__(self):
        self.a = 0
        self.b = 1
        return self

    def __next__(self):
        fib = self.a
        if fib > self.max:
            raise StopIteration
        self.a, self.b = self.b, self.a + self.b  # evaluate RHS first, then assign pair
        return fib
```

Example from Pilgrim's *Dive Into Python 3*

Exceptions

```
try:
        some code…
except RunTimeError as e:
        e is bound to the exception object
        do what you want…

# Other exceptions are not caught
# Read about finally clause
```