Statistical Learning Problems

Prerequisites

Course Overview

Probability Review

Python, Conda and Jupyter Notebook

Lecture 1 Introduction

Probability Review

Shiwei Lan[1]

[1]School of Mathematical and Statistical Sciences
Arizona State University

STP598 Machine Learning and Deep Learning
Fall 2021

Overview

S.Lan

Statistical Learning Problems

Prerequisites

Course Overview

Probability Review

Python, Conda and Jupyter Notebook

Machine Learning and Deep Learning

- Instructor: Shiwei Lan <slan@asu.edu>
  - Office: WXLR 544
  - Office hours: WF 11a - 12p @ https://asu.zoom.us/j/8055899886
- Teaching Assistants:
  - Shuyi Li <shuyili3@asu.edu>
  - Office hours: TBA

Course Info

- Course Schedule, Lecture Notes, etc.

  https://slan-teaching.github.io/STP598mldl/

- Discussion and Questions

  Slack channel **STP 598mldl**

- Homework and Grades on **Canvas**

- Coding Assignments on **Nbgrader**

Course Info

- Textbook:
  - **Required**: [ESL] Hastie, T., Tibshirani, R. Friedman, J. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction". Springer. (free online PDF)
  - **Required**: [DL] Goodfellow, I., Bengio, Y. Courville, A. "Deep Learning" (free online)
  - **Supplemental**: [DLP] Stevens, E., Antiga, L., Viehmann, T. "Deep Learning with PyTorch". (Link)
- Programming language:
  - Python is mandatary.
  - All homework reports must be submitted in either Word of PDF format, no other formats accepted.
  - R Markdown is recommended to generate PDF report.

Table of Contents

1. Statistical Learning Problems
2. Prerequisites
3. Course Overview
4. Probability Review
5. Python, Conda and Jupyter Notebook

Examples

Statistical learning is the process of extracting statistical regularities from datasets. They are motivated from real world problems. A few examples:

- Email Spam (classification)
- Handwritten Digits (classification)
- DNA microarray (clustering)

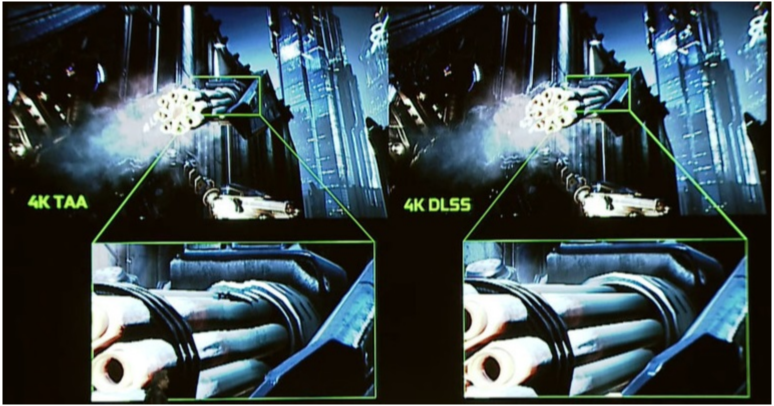
Examples

A compressive sensing method is used to decompose a surveillance video image into background and moving objects. Matrix decomposition and ℓ_1 penalization are used in this method.

Examples from Jiang, Hong, Wei Deng, and Zuowei Shen. "Surveillance video processing using compressive sensing." arXiv preprint arXiv:1302.1942 (2013).

Examples

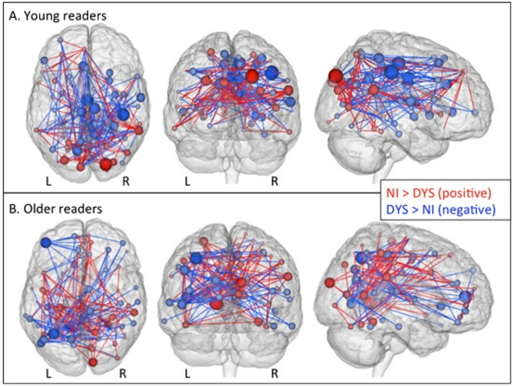
Single image super-resolution is used for sharping an image, or recovering a high-resolution image from a low-resolution one. Deep learning can be used to recognize the objects.



Example by NVIDA using deep learning super-sampling in computer games.

Overview
S.Lan
Statistical Learning Problems
Prerequisites
Course Overview
Probability Review
Python, Conda and Jupyter Notebook

Graphical models can be used to understand the network connectivity of multiple brain regions. This information can help diagnose diseases such as parkinson and dyslexia (difficulty reading).



Example from Finn, Emily S., et al. "Disruption of functional networks in dyslexia: a whole-brain, data-driven analysis of connectivity." Biological psychiatry 76.5 (2014): 397-404.

Examples

Auto piloting is made possible by recognizing objects such as lines, cars, pedestrians, etc. in real time images. These are essentially classification problems, carried out through deep learning.

Example from Tesla: https://www.tesla.com/autopilot; And the video of accident that killed a woman in Tempe, Arizona.

Examples

OpenAI Five is an AI system that plays Dota 2 against real human professionals. "... OpenAI Five sees the world as a list of 20,000 numbers which encode the visible game state, and chooses an action ...". They lose to human professional team 0:2 in TI 2018.


Bill Gates
@BillGates

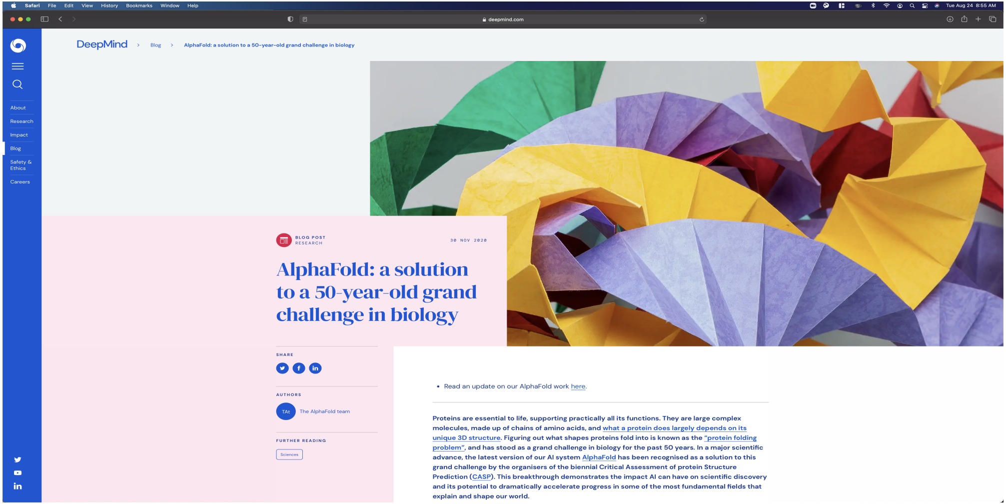
#AI bots just beat humans at the video game Dota 2. That's a big deal, because their victory required teamwork and collaboration – a huge milestone in advancing artificial intelligence. b-gat.es/2KqAlzU

6:25 PM - Jun 26, 2018

14.9K 6,442 people are talking about this

OpenAI Five: https://openai.com/five/;

Examples



AlphaFold: https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology;

I was like...

Table of Contents

1. Statistical Learning Problems
2. Prerequisites
3. Course Overview
4. Probability Review
5. Python, Conda and Jupyter Notebook

Prerequisites

- Probability: probability and random variables, distributions
- Statistics: estimators, likelihood, linear regressions
- Mathematics: linear algebra and calculus
- Programming and software:
  - program in Python
  - introduce Jupyter Notebook / R Markdown for homework
  - optimization basics

Table of Contents

1. Statistical Learning Problems
2. Prerequisites
3. Course Overview
4. Probability Review
5. Python, Conda and Jupyter Notebook

Learn pattern from data

- Suppose we have a set of training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$.
- Each x_i is a p-dimensional covariate vector that may represent
  - Gray scale of pixels in an image
  - Frequency of a particular word in an email
  - Concentration of a gene expression in blood
  - Brain electrical impulses
  - ...
- Each y_i is an outcome variable (or vector) that may represent
  - Cancer status (binary classification)
  - Type of object shown in the image (multicategory classification)
  - Surface temperature of a planet (regression)
  - Weight and volume of a tumor (multivariate regression)
  - ...
- Goal: learn patterns

Supervised vs. Unsupervised Learning

Oftentimes, there are two types of problems. And the goals are different:

- Supervised learning: data contains both "input" variables (covaraites) and "output" variable(s). And we want to use the inputs to predict the values of the outputs.
- Unsupervised learning: data contains only the "input" variables, and we want to know what is the underlying mechanism that generates the data.

Supervised Learning

- Response variable or outcome Y (a random variable)
  - Regression: Y is quantitative/continuous:

$$y_i \in \mathbf{R}$$

  - Classification: Y is categorical/discrete:

$$y_i \in C = \{0, 1\} \quad \text{or} \quad C = \{-1, +1\}$$

- Based on the training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$, we aim to learn/estimate a function f, which describes the relationship between (variable) X and Y:

$$Y \longleftarrow f(X)$$

- Goals:
  1. achieve small prediction error
  2. (and/or) assess the effect of each features on Y

Unsupervised Learning

- No Y, just a set of features $X \in \mathbf{R}^p$
- Training data $\mathcal{D}_n = \{x_i\}_{i=1}^n$ where each $\mathbf{x}_i$ is a random draw of X
- Goal: find patterns in the data (such as clusters), understand the data generating process of X, etc.
- Sometimes, it is difficult to measure the performance of an unsupervised leaning method.
- The goal is fuzzy, but nevertheless it is an important problem.

Types of Learning Problems

- Supervised learning with labeled data: regression or classification
- Unsupervised learning with unlabeled data: clustering, network, graphical model
- There is also a semi-supervised learning that use a large set of unlabeled data to learn the distribution of independent variables and utilize that information for analyzing a smaller set of labeled data. However, we will not cover that in this course.

Course Overview

- Supervised Learning:
  - Linear models and penalization
  - Classification, generalized linear regression
  - K nearest neighbor, tree, random forests
  - Support vector machine, neural networks
- Unsupervised Learning:
  - PCA, K-mean and hierarchical clustering
- Other Concepts / Topics
  - Bias-variance trade-off
  - Variable selection
  - Cross-validation
  - Deep neural networks

Table of Contents

1. Statistical Learning Problems
2. Prerequisites
3. Course Overview
4. Probability Review
5. Python, Conda and Jupyter Notebook

Continuous distributions:

- Normal distribution $\mathcal{N}(\mu, \sigma^2)$:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

- Student's t-distribution with d.f. r:

$$f(x) = \frac{\Gamma(\frac{r+1}{2})}{\sqrt{\pi r}\,\Gamma(\frac{r}{2})} \left(1 + \frac{x^2}{r}\right)^{-\frac{r+1}{2}}$$

- F-distribution with d.f. d_1 and d_2:

$$f(x) = \frac{1}{B\left(\frac{d_1}{2}, \frac{d_2}{2}\right)} \left(\frac{d_1}{d_2}\right)^{\frac{d_1}{2}} x^{\frac{d_1}{2}-1} \left(1 + \frac{d_1}{d_2}x\right)^{-\frac{d_1+d_2}{2}}$$

Relationships among distributions

- $X \sim \mathcal{N}(0,1) \quad \longrightarrow \quad X^2 \sim \chi^2(1)$
- $Z \sim \mathcal{N}(0,1), X \sim \chi^2(r) \quad \longrightarrow \quad \frac{Z}{\sqrt{X/r}} \sim t(r)$
- $X_1 \sim \chi^2(a), X_2 \sim \chi^2(b) \quad \longrightarrow \quad \frac{X_1/a}{X_2/b} \sim F(a,b)$

! Review the properties of Normal, χ^2, t, and F.

Discrete distributions:

- Bernoulli distribution $\mathrm{Bern}(p)$:

$$f(k) = p^k(1-p)^{1-k}, \quad k \in \{0, 1\}$$

- Binomial distribution $\mathcal{B}(n, p)$:

$$f(k) = \binom{n}{k} p^k(1-p)^{n-k}, \quad k \in \{0, 1, \cdots, n\}$$

- Poisson distribution with rate λ:

$$f(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Overview
S.Lan
Statistical Learning Problems
Prerequisites
Course Overview
Probability Review
Python, Conda and Jupyter Notebook

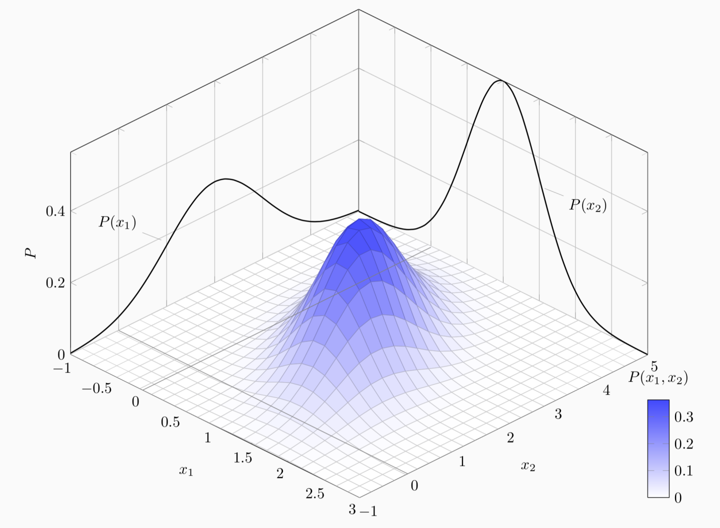
Multivariate normal distribution

- Normal (Gaussian) distribution is the most frequently used distribution in statistics
- By the central limit theory, sample means will converge to Gaussian as sample size increases
- In many cases, we will concern about two or many normally distributed random variables
- Lets consider two random variables X and Y that are jointly normally distributed with density function

$$
\begin{aligned}
& f(x,y) \\
= & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \times \\
& \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}-\frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right]\right\}
\end{aligned}
$$

where μ_x and μ_y are the means, σ_x and σ_y are the standard deviations, and ρ is the correlation coefficient.

Multivariate normal distribution



Multivariate normal distribution

- The sum of two random normal variables are also normally distributed.
- Suppose that $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$, $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ and the correlation coefficient between X and Y is ρ, then the sum

$$X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2 + 2\rho\sigma_x\sigma_y)$$

  and the linear combination

$$aX + bY \sim \mathcal{N}(a\mu_x + b\mu_y, a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\rho\sigma_x\sigma_y)$$

- From the previous example, what is the probably that a randomly selected student has a combined score over 150, i.e., $\mathrm{P}(X + Y > 150)$?
- Find $\mathrm{P}(2X + 3Y > 350)$.
- Find that the student did better on Exam 1 than on Exam 2, i.e., $\mathrm{P}(X - Y > 0)$.

Multivariate normal distribution

- We usually represent a multivariate normal (MVN) distribution in a matrix form:
- Let $X = (X_1, X_2, \ldots, X_p)^\mathsf{T}$ be a p-dimensional random vector that follows the distribution $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean $\boldsymbol{\mu}_{p\times 1}$ is a p-dimensional vector, and the covariance $\boldsymbol{\Sigma}_{p\times p}$ is a $p \times p$ dimensional symmetric and positive-definite matrix.
- The i-th element, μ_i, and (i,j)-th element, σ_{ij} are defined as follows respectively

$$\mu_i = \mathrm{E}(X_i), \quad \sigma_{ij} = \mathrm{Cov}(X_i, X_j), \quad , i,j = 1, 2, \cdots, p.$$

- In particular, $\sigma_{ii} = \sigma_i^2 = \mathrm{Var}(X_i)$. And $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$, where ρ_{ij} is the correlation between X_i and X_j. In this general case, we have correlation matrix $\mathbf{P}_{p\times p}$ with ρ_{ij} as its (i,j)-th element.
- When $p = 1$, $\mathbf{P}_{p\times p} = \rho$ becomes correlation coefficient as mentioned before.

Multivariate normal distribution

- The pdf of X is

$$\frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Let Z be a q-dimensional vector of linear combinations of X such that $Z = \mathbf{A}_{q\times p}X + \mathbf{b}_{q\times 1}$, then Z follows a MVN distribution:

$$Z \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\mathsf{T})$$

- A special case: if $Z = \mathbf{\Sigma}^{-1/2}(X - \boldsymbol{\mu})$, then entries in Z follow iid normal (We call this transformation *standardization*):

$$Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{p\times p})$$

- For example, we could let $Z = \mathbf{L}^{-1}(X - \mu)$ where $\mathbf{L}$ is the Cholesky factor of $\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^\mathsf{T}$, or $Z = \mathbf{D}^{-1/2}\mathbf{\Gamma}^\mathsf{T}(X - \mu)$, where we use the eigen-decomposition $\mathbf{\Sigma} = \mathbf{\Gamma}\mathbf{D}\mathbf{\Gamma}^\mathsf{T}$.

Multivariate normal distribution

- Conditional distribution of multivariate normal is also frequently used
- Let the random vector $(X^\mathsf{T}, Y^\mathsf{T})^\mathsf{T}$ be jointly distributed as

$$\begin{pmatrix} X_{p\times 1} \\ Y_{q\times 1} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{xy}^\mathsf{T} & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \right)$$

  where $\boldsymbol{\mu}_x$ is p-dimensional, $\boldsymbol{\mu}_y$ is q-dimensional. $\boldsymbol{\Sigma}_{xx}$ is $p \times p$ dimensional covariance matrix of X, $\boldsymbol{\Sigma}_{yy}$ is $q \times q$ dimensional covariance matrix of Y, and $\boldsymbol{\Sigma}_{xy}$ is $p \times q$ dimensional (cross) covariance matrix between X and Y with (i, j)-th element $\mathrm{Cov}(X_i, Y_j)$.
- The conditional distribution of $X|Y = y$ is

$$X|Y = y \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(y - \boldsymbol{\mu}_y), \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^\mathsf{T} \right)$$

Overview

S.Lan

Statistical Learning Problems

Prerequisites

Course Overview

Probability Review

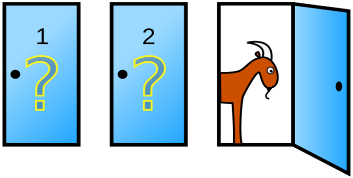
Python, Conda and Jupyter Notebook

- **Independent events** For two independent events A and B, $P(A \cap B) = P(A)P(B)$
- **Mutually exclusive events** For two mutually exclusive events A and B, $P(A \cap B) = 0 \implies P(A \cup B) = P(A) + P(B)$ "sum rule"
- **Conditional probability** The probability of A given B, $P(A|B) = P(A \cap B)/P(B) \implies P(A \cap B) = P(A|B)P(B)$ "multiplication rule"
- **Bayes ("inverse") rule** The probability of A given B can be obtained

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} = \frac{P(A)P(B|A)}{P(A)P(B|A) + P(A^c)P(B|A^c)}$$

- posterior $\propto$ prior $\times$ likelihood

Monty Hall Problem



The Monty Hall problem is a brain teaser, in the form of a probability puzzle, loosely based on the American television game show *Let's Make a Deal* and named after its original host, *Monty Hall*.

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?"

Is it to your advantage to switch your choice?

I was like...

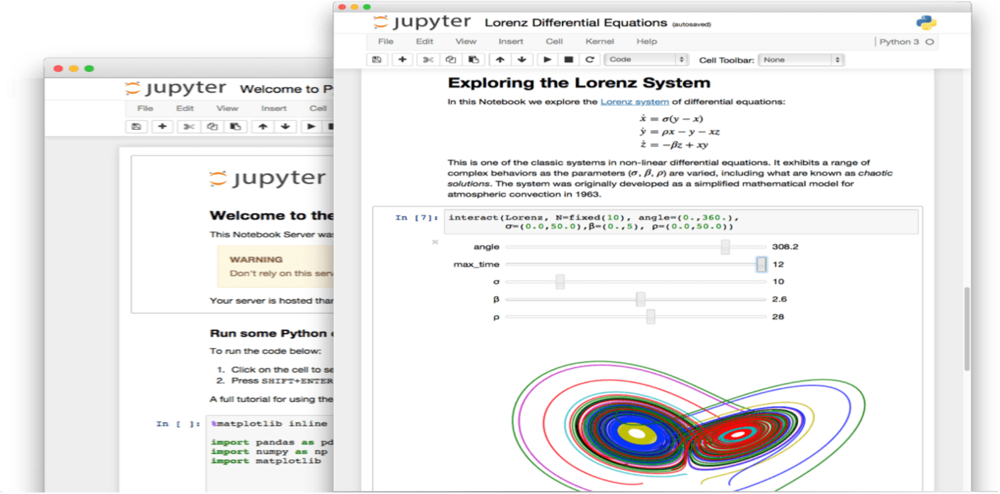
Table of Contents

1. Statistical Learning Problems
2. Prerequisites
3. Course Overview
4. Probability Review
5. Python, Conda and Jupyter Notebook

Python, Conda and Jupyter Notebook

- Python is a free software environment for statistical machine learning and graphics.
  - https://www.python.org
- Conda is an open source package / environment management system for Python.
  - https://anaconda.org
- Jupyter Notebook is a Python package to integrate code, equations, visualizations in document.
  - https://jupyter.org
- R Markdown has also integrated Python. Check link1 link2.

Jupyter



jupyter Welcome to P
File Edit View Insert Cell
jupyter
Welcome to the
This Notebook Server wa
WARNING
Don't rely on this serv
Your server is hosted than
Run some Python
To run the code below:
1. Click on the cell to se
2. Press SHIFT+ENTER
A full tutorial for using the
In [ ]: %matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
jupyter Lorenz Differential Equations (autosaved)
File Edit View Insert Cell Kernel Help
Python 3
Code
Cell Toolbar: None
Exploring the Lorenz System
In this Notebook we explore the Lorenz system of differential equations:
$\dot{x} = \sigma(y - x)$
$\dot{y} = \rho x - y - xz$
$\dot{z} = -\beta z + xy$
This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
angle 308.2
max_time 12
σ 10
β 2.6
ρ 28