Fundamentals and tools

Applied machine learning (EDAN95)
Lecture 02 — Fundamentals and tools
2019–11–06
Elin A. Topp

Goodfellow chapter 2
various sources

Today's agenda

- Concept learning
- (Recap) Linear Algebra (Goodfellow chapter 2), example: PCA to construct a classifier for localisation
- Some tips on Python / Jupyter notebooks and Numpy, example: image filtering (convolution)

Today's agenda

- Concept learning
- (Recap) Linear Algebra (Goodfellow chapter 2)
- Some tips on Python / Jupyter notebooks and Numpy

Concept learning

- A central issue in learning is the acquistion of general concepts from examples, e.g., finding the descriptive features for deciding, whether something one observes is a bird or not
- Question: Given some descriptive features for the weather on a certain day, and a list of already rated weather conditions, is this particular day a good day for enjoying some sport activity?
- Assume a set of attributes (features) and possible values for them, which express the constraints for classifying a day as "good" (TRUE):
  Sky (Sunny, Rainy, Cloudy), AirTemp (Warm, Cold), Humidity (Normal, High), Wind (Strong, Weak), Water (Warm, Cool), Forecast (Same, Change)
- Represent hypotheses for the concept by a set of values for the attributes, where
  - ? - any value is acceptable
  - Ø - no value is acceptable
  - <value> - a specific value is acceptable
- Find the most specific hypothesis that matches the data (examples) in a training set

Enjoy sports - when?

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport?
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- Most general hypothesis: EnjoySport = yes on every day is h = <?, ?, ?, ?, ?, ?>
- Most specific hypothesis: EnjoySport = yes on no day is h = < Ø, Ø, Ø, Ø, Ø, Ø >

- A fundamental assumption (The Inductive Learning Hypothesis):

  Any hypothesis approximating the target function well over a sufficiently large training set will also approximate the target function well for unseen data

- So - let's find this hypothesis by search…

Search space for EnjoySport

- # of hypotheses h for the task: 3*2*2*2*2*2 = 96 (# of values per attribute)
- or actually, including ? and Ø: 5*4*4*4*4*4 = 5120 *syntactically distinct* h
- but, given that any Ø makes all other attributes obsolete (render the outcome as "no", we have 4*3*3*3*3*3 +1 = 973 *semantically distinct* hypotheses h

- Hence, the search needs to be organised somehow (we cannot test all h, if the problem grows more complex…)

Find-S

- Start with the most specific hypothesis

  $h \leftarrow < \emptyset , \emptyset , \emptyset , \emptyset , \emptyset , \emptyset >$ (no day is a good day for sports)

- observe first example (it is a positive one)

  $h \leftarrow < Sunny, Warm, Normal, Strong, Warm, Same >$

  (a day with exactly these values in its attributes is a good day for sports, all others are not)

- observe second example (it is a positive one)

  $h \leftarrow < Sunny, Warm, ?, Strong, Warm, Same >$

  (since both “Normal” and “High” for the third attribute produce “yes”, the third attribute can obviously have any value, while the others need to be fixed …)

- ignore the third example (it is negative)

- observe the fourth example and get

  $h \leftarrow < Sunny, Warm, ?, Strong, ?, ? >$

Find-S

- Find-S finds the most specific hypothesis that matches the training data (positive examples) and it is correct regarding the negative examples it excludes.
- But there is no guarantee that the found h is the ONLY one that covers the concept fully.
- Also, maybe it would be better to also look at the most general hypothesis that still fits the training data?

Version Space and Candidate Elimination

- Candidate Elimination finds ALL hypotheses consistent with the training data, the Version Space of the hypothesis space H.

- The algorithm:
  Initialize G to the set of maximally general hypotheses in H: $G = \{ < ?, ?, ?, ?, ?, ? > \}$
  Initialize S to the set of maximally specific hypotheses in H: $S = \{ < \emptyset , \emptyset , \emptyset , \emptyset , \emptyset , \emptyset > \}$
  For each training example d, do

If d is a positive example
- Remove from G any hypothesis inconsistent with d
- For each hypothesis s in S that is not consistent with d
  - Remove s from S
  - Add to S all minimal generalisations h of s such that
    - h is consistent with d, and some g in G is more general than h
  - Remove from S any hypothesis that is more general than another hypothesis in S

If d is a negative example
- Remove from S any hypothesis inconsistent with d
- For each hypothesis g in G that is not consistent with d
  - Remove g from G
  - Add to G all minimal specialisations h of g such that
    - h is consistent with d, and some s in S is more specific than h
  - Remove from G any hypothesis that is less general than another hypothesis in G

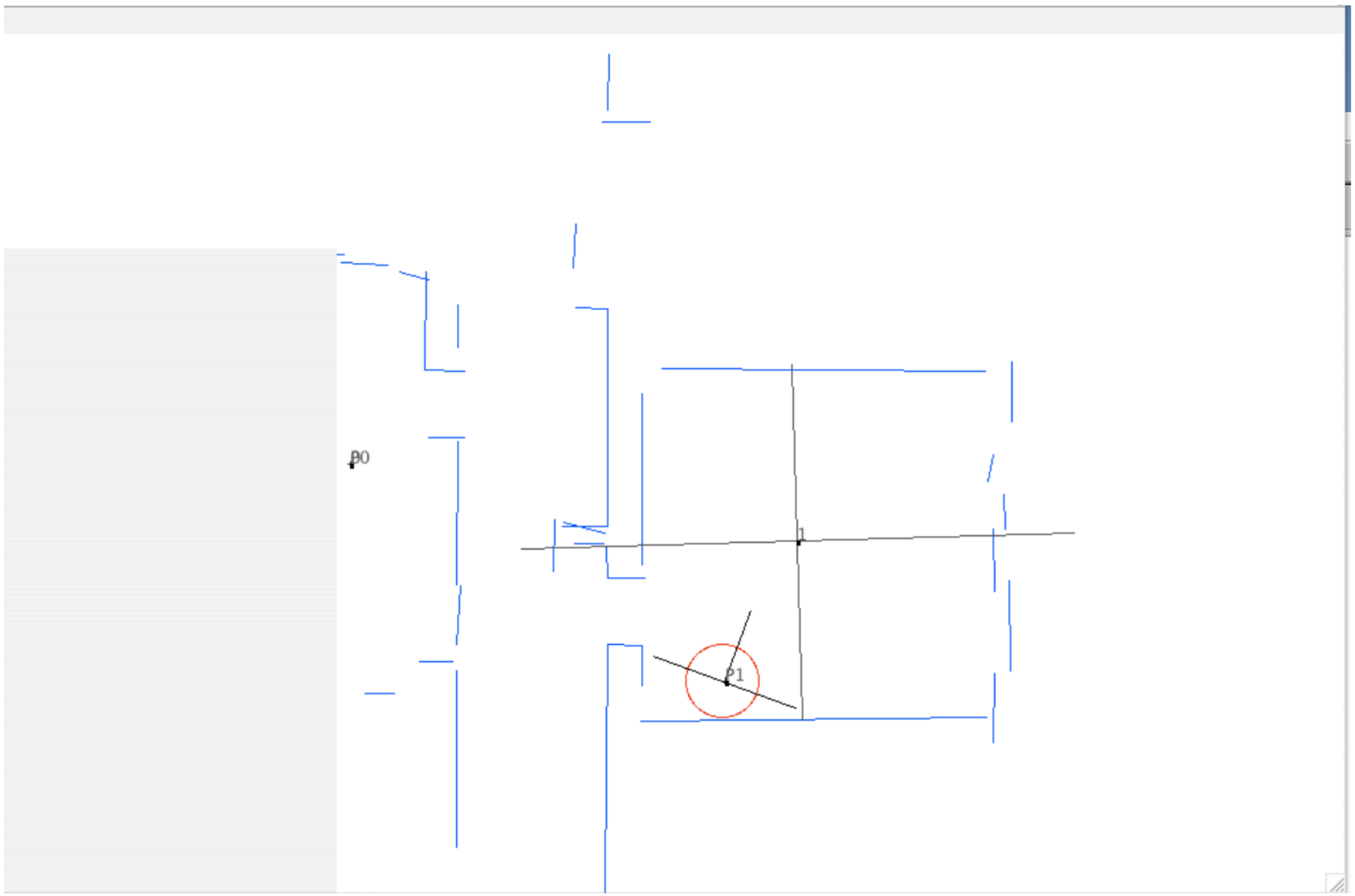
Problem solved?

- Candidate Elimination converges if the training data are correct and there is actually a correct hypothesis in H (if it converges after sufficiently many positive AND negative examples to one single hypothesis, this is the optimal and correct one).
- Extremely sensitive to noise - one single "false negative" in the training data can eliminate the correct hypothesis, and it will never come back …
- Not exactly practical with an image of a bird or not a bird, if attributes are not conceptual but correspond to pixels with a much larger value range than "high" vs "normal".

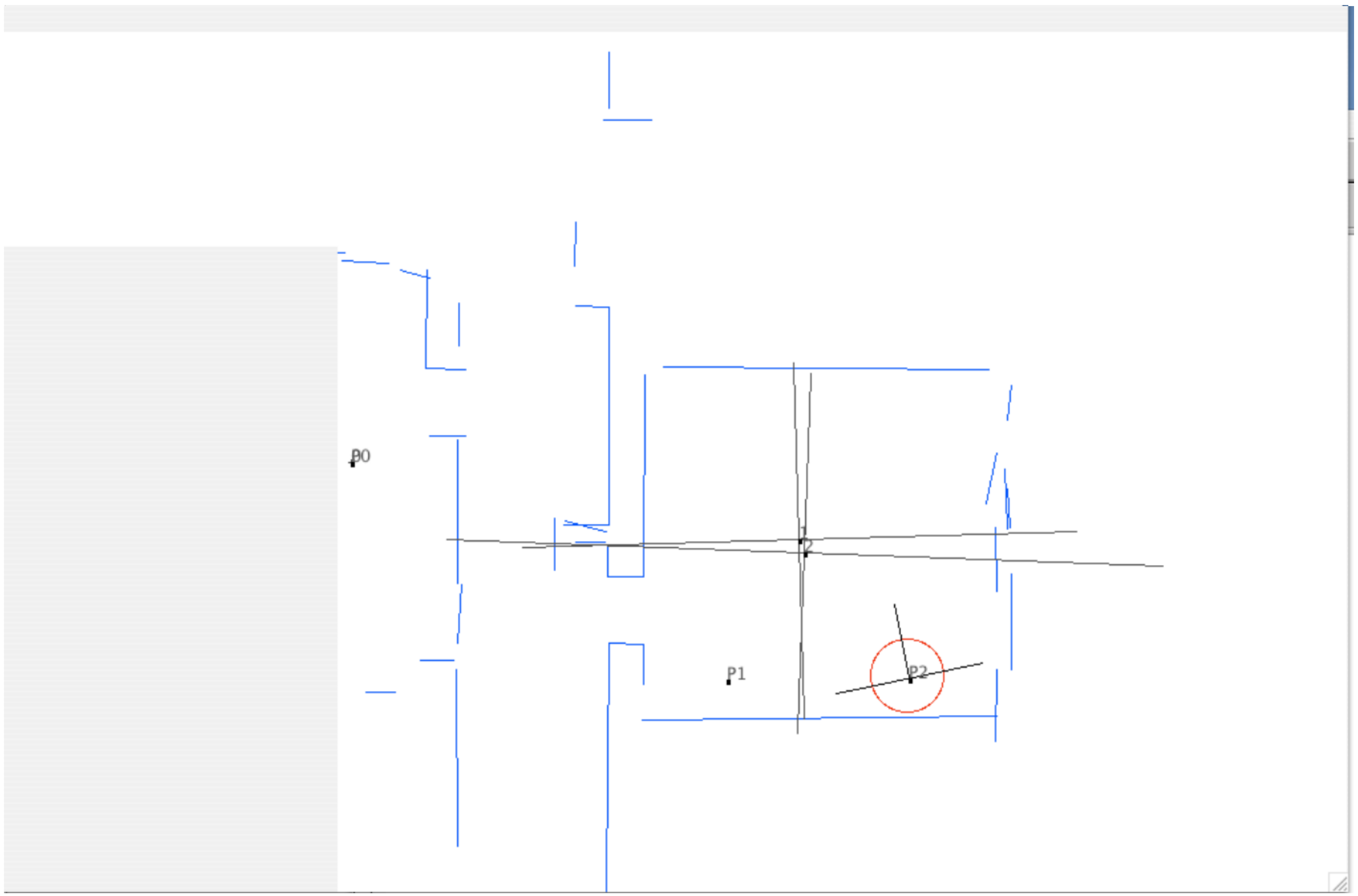
Today's agenda

- Concept learning
- (Recap) Linear Algebra (Goodfellow chapter 2)
- Some tips on Python / Jupyter notebooks and Numpy

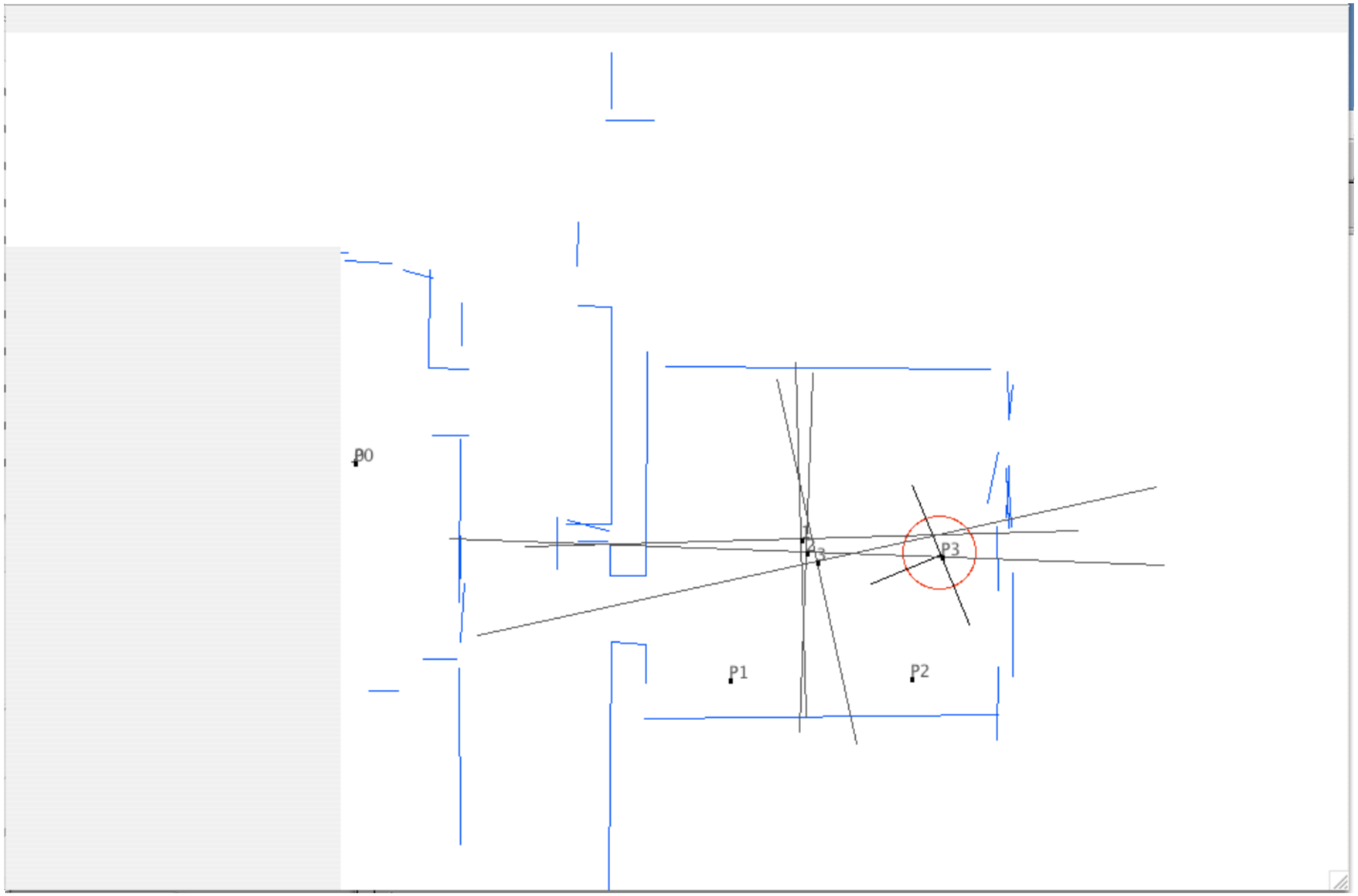
Building up an ellipsoid from different viewpoints



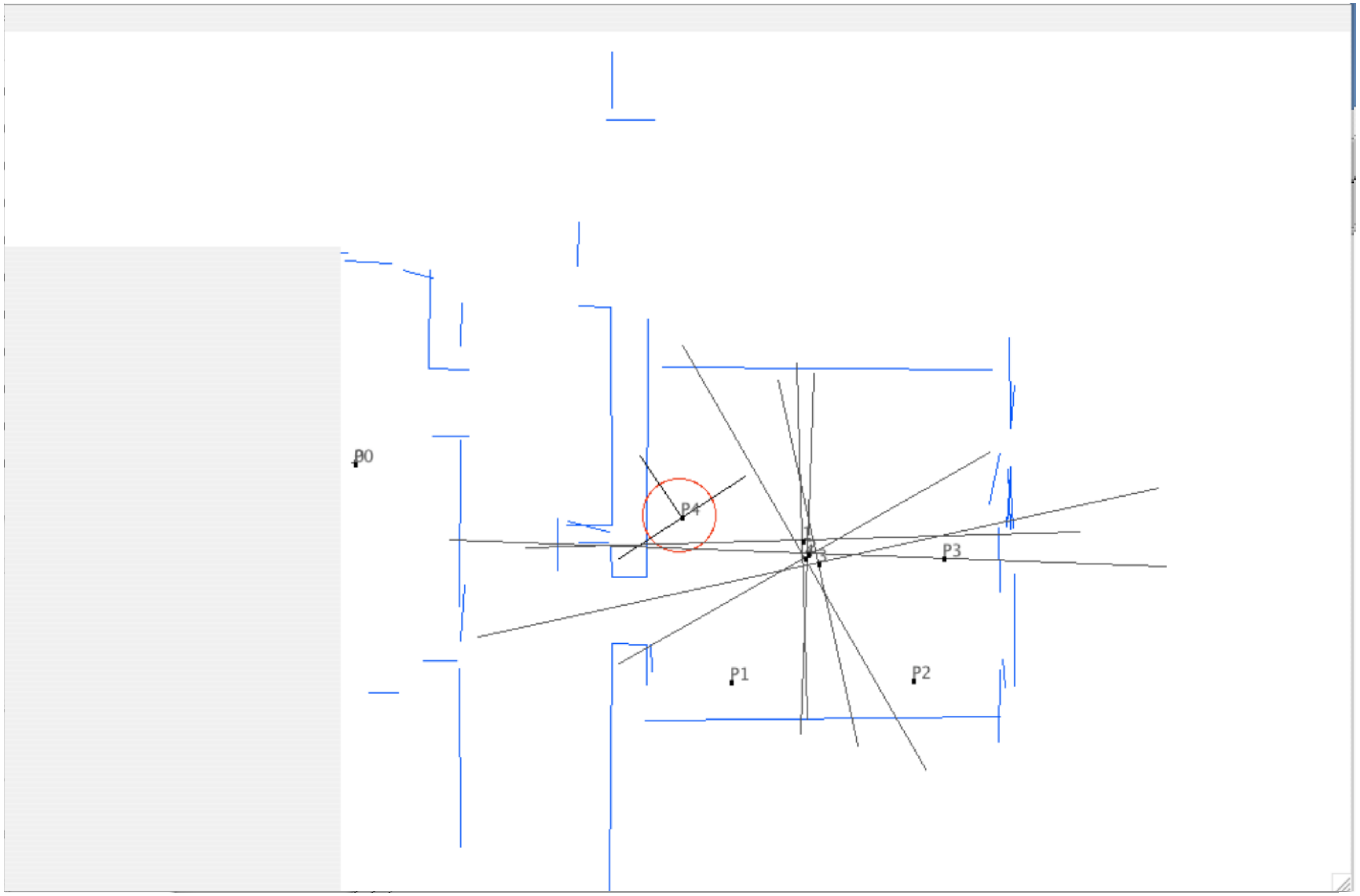
Building up an ellipsoid from different viewpoints



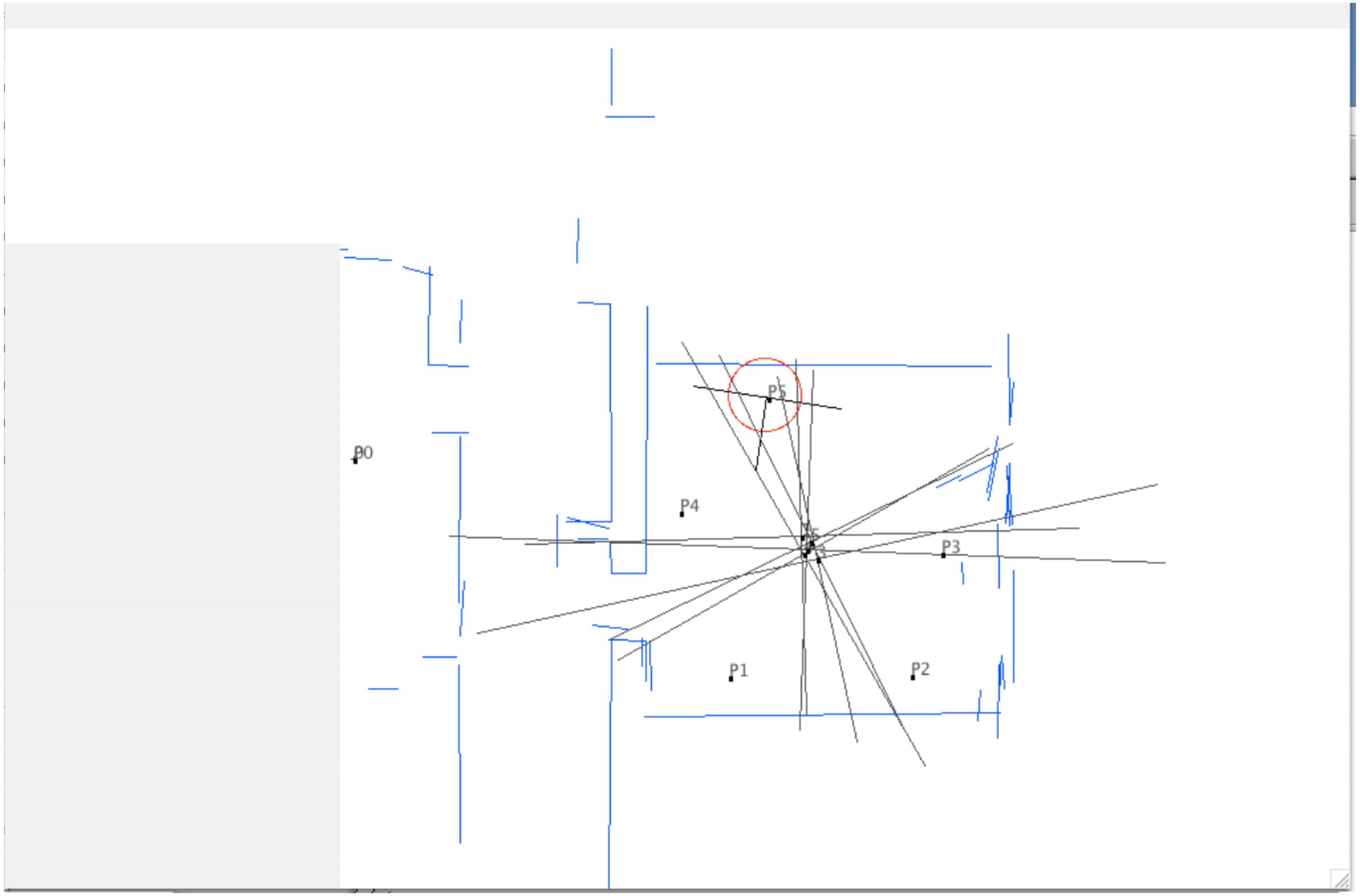
Building up an ellipsoid from different viewpoints



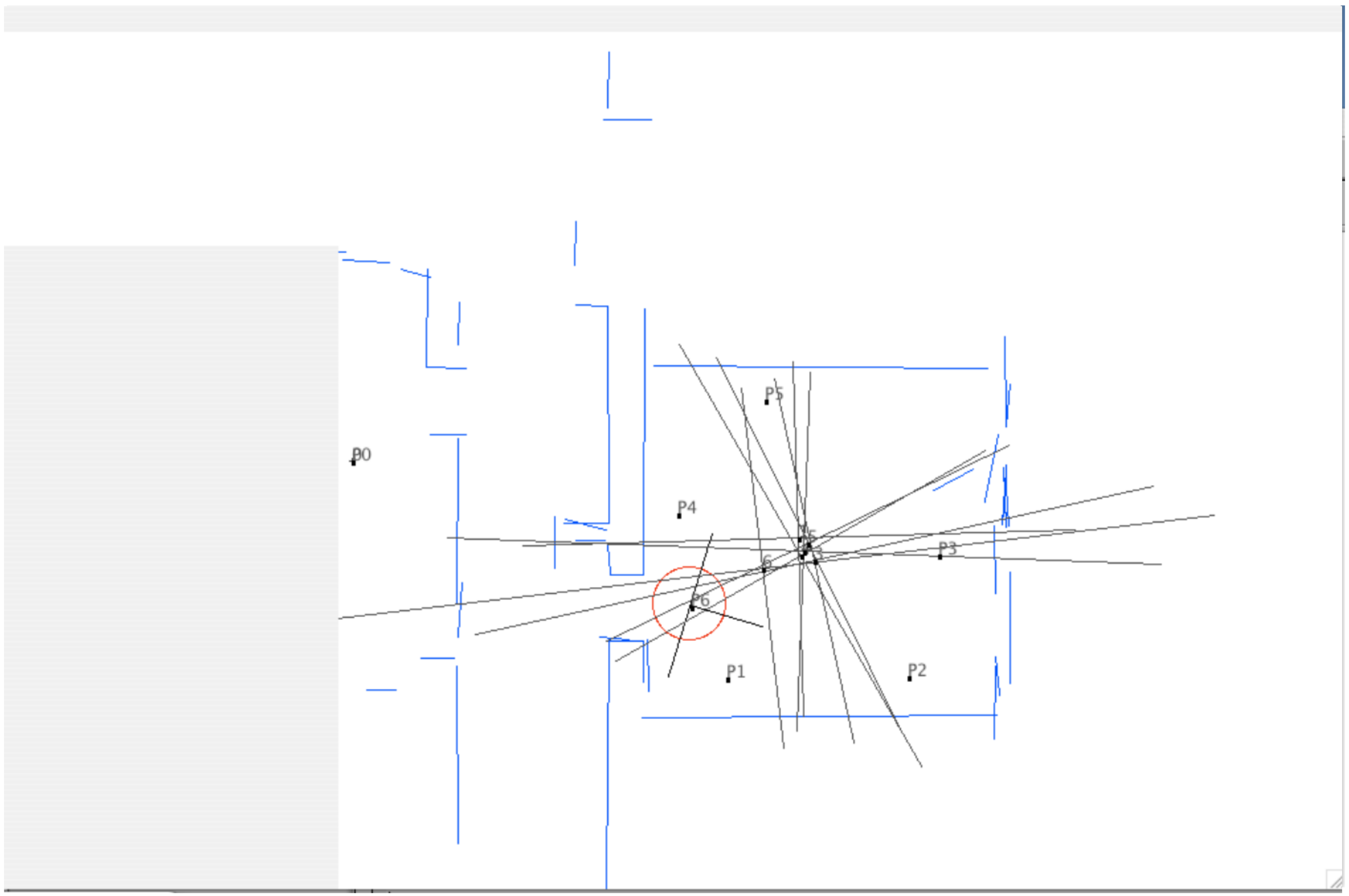
Building up an ellipsoid from different viewpoints



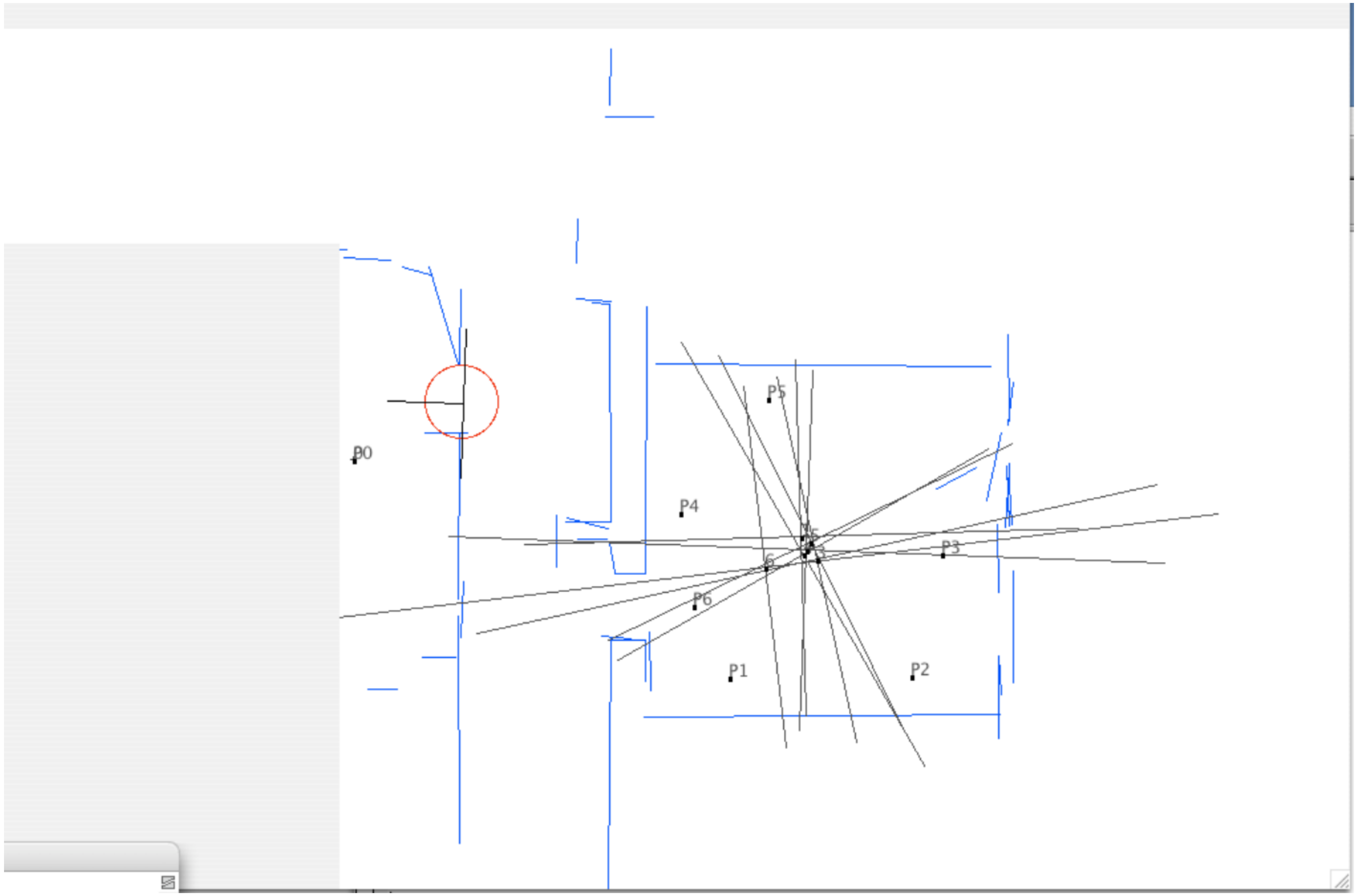
Building up an ellipsoid from different viewpoints



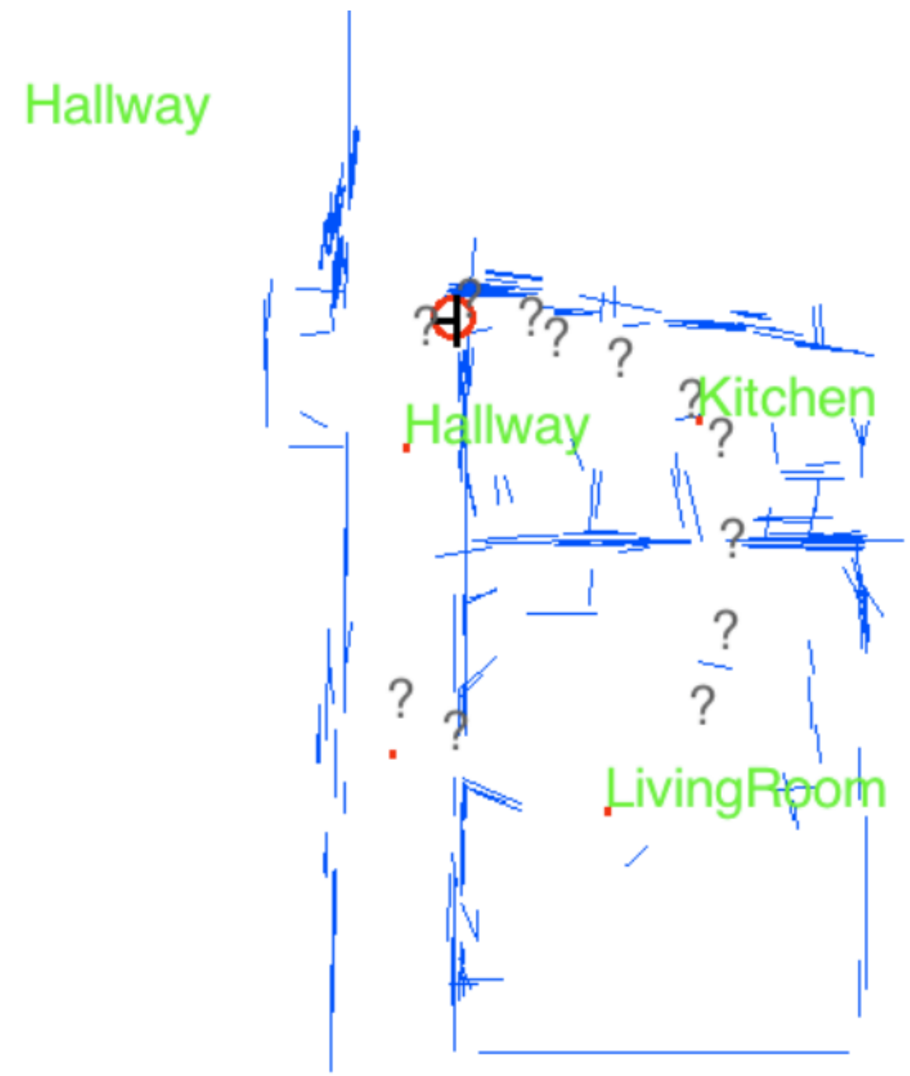
Building up an ellipsoid from different viewpoints



Building up an ellipsoid from different viewpoints



Using the estimate to ask for clarification



Today's agenda

- Concept learning
- (Recap) Linear Algebra (Goodfellow chapter 2)
- Some tips on Python / Jupyter notebooks and Numpy (shown in Jupyter notebook)

Outlook lecture 3

- Decision Trees
- Recap Information Theory and Probability Theory

- Reading advise: Tom Mitchell, chapter 3, Goodfellow chapter 3, online material by Géron on DTs.

Today's summary

- Introduced concept learning as an intuitive (conceptual) approach to machine learning (including its limitations)
- Walked through a recap of Linear Algebra concepts, touching upon EVD / SVD and PCA. Exemplified PCA with categorisation of locations from own research
- Showed some examples of use for Numpy in context of image filtering (convolution)

- Reading advise: Tom Mitchell, chapter 2, Goodfellow chapter 2, Numpy (SciPy) tutorials / reference at https://docs.scipy.org/doc/numpy/reference/index.html, Pierre Nugues introductory chapter on Python (holler, if you did not get it yet)