A possible general Heuristic:

Computing features which are physically-intelligent (thus hoping that the computer will bypass the need to do the “heavy” physical computations)

There are “classically” 2 main challenges:

1. coming up with the best, most physically-intelligent features

2. Picking the best algorithm for the job (esp. considering the “not identified” class appearing only in test set)

1. Features

|  |  |  |
| --- | --- | --- |
| Feature | Logic | Method of computing |
| abs(x-vel), abs(y-vel) | It doesn’t matter in what direction them missile was shot (as opposed to z-vel, see also “going-up” feature). Not surprisingly, also appears in equations ([detailed equations](https://www.google.co.il/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&cad=rja&uact=8&ved=0ahUKEwiF2rLV7YTXAhUpAcAKHUVmAG8QFghMMAg&url=http%3A%2F%2Fweb.mit.edu%2F16.unified%2Fwww%2FFALL%2Fsystems%2FLab_Notes%2Ftraj.pdf&usg=AOvVaw0XRUUrUbvU0ig9xz2rTB92)) |  |
| Other option for the above: Normalized direction  Projection of X,Y and velX,velY on normalized direction | Assuming the direction is the same during the entire measurement. | Least squares? |
| Standard deviation of X,Y,velX,velY | Smaller target rotating target have bigger standard deviation of position.  Rotating target have bigger standard deviation of vel. |  |
| Maximum error X,Y | Assuming there is correlation between errors in X,Y and height (or maximum height) |  |
| vel^2 | Appears in “[drag equation](https://spaceflightsystems.grc.nasa.gov/education/rocket/drageq.html)” |  |
| Acceleration (Vel’) - avg / max / min | obvious |  |
| Drag force | Different rocket types have different aerodynamics features |  |
| Change of accel’ (Vel’’) - avg / max / min | obvious |  |
| Air density - avg / max / min | Affects drag | Atmosisa (matlab) |
| Air pressure | Affects drag | Atmosisa (matlab) |
| Going\_up (binary) |  |  |
| Degree of launch | Probably different types have different launching degrees limits | difficult |
| Momentary Degrees | Info on the degree of the missile may replace the difficult “phase\_of\_flight” & “degree\_of\_launch” below.  Defined according to direction of vel. With respect to the surface. | Not hard |
| Momentary change in degree (1st derivative) | Not hard |
| Degrees - 2nd derivative | Not hard |
| (Max degree of launch) |  |  |
| (Min degree of launch) |  |  |
| Top estimated height |  |  |
| (Min height per type) |  |  |
| (Max vel per type) | Finding a bound to filter into groups of classes |  |
| (Min vel per type) |
| (Max height per type) |
| Is\_engine\_on  (maybe in phases of flight) | Very different behaviour - we want the machine to “know” this fact (and perhaps even give it a strong bias manually in algorithm) |  |
| Phase\_of\_flight | It will be VERY useful If we can figure out, in what stage (say, 1-10) of the flight our sample is from (after engine is off) | difficult |

2. Algorithmic Issues

* Clustering seems to depend heavily on the phase the missile is in (perhaps different missiles have similar segments: for one the segment begins after 20% of its course, for the other after 30%).
  + Let’s say we know how to compute this (=feature “phase\_of\_flight” above). How do we make the machine know it is important? (force this feature)
* Time-series (or “functional”) analysis vs. “regular” analysis
  + Using “regular” features limits us to, e.g. avg\_x\_vel, max\_x\_vel - while functional analysis can create curves and compare them to each other, thus getting a much better comparison - is it smart model or more like deep learning using a lot of “small” features?
* The extra class appearing in test set only:
  + Possible strategy:
    - Choose a model that can give a **grade** **of confidence to its results.** And take the least

3. Technical Issues

* What’s the best manner to convert the strange manner in which the data is given (time series appearing in separately labeled columns such as: posZ\_6, posZ\_7 etc…)
  + See Shay’s proposed solution in git, not necessarily the best

4. Useful info

* <http://kevinkparsons.com/projects/matlab-rocket-trajectory-simulation.html>
* <https://spaceflightsystems.grc.nasa.gov/education/rocket/drageq.html>
* <http://web.mit.edu/16.unified/www/FALL/systems/Lab_Notes/traj.pdf> (more detailed than previous link)

· גישות לפתרון:

1. סיווג דו-שלבי:

1. סינון ראשוני ע"פ פרמטרים לא מעובדים לקלאסים עם false negative =0

2. סינון חכם בתוך הקלאס

2. למצוא כיוון כללי (X,Y) ולהעביר למערכת צירים של כיוון וגובה

3. להראות התפלגות שגיאות סביב כיוון ההתקדמות (מניחים גאוסי)

4. לצמצם את הסטייה סביב כיוון ההתקדמות

5. סיווג של הרעשים

6. לשערך מסלול שלם של מטרה ע"פ חלק המסלול הנתון ולחפש שיא גובה ולמרחק הירי

1. לעשות ע"פ מודל מטרה

2. שיטה טיפשה: להשוות לפרבולה

7. חישוב גובה מקסימלי (חסם) ע"פ הקלט בשלב היורד

1. ההנחות הן שמנוע לא פעל כבר בשלב שיא הגובה ואחריו

8. אולי לסווג רקטות לעוד תתי קבוצות. למשל רקטה 122 בעלייה ורקטה 122 בירידה

· הנחות:

1. מטרה נעה בכיוון X,Y קבוע עם שגיאת מדידה מתפלגת גאוסית

2. מנוע לא פועל משלב שיא גובה ולאחריו

3. שגיאות מהירות ומרחק קורלטיביות לסוג מטרה

4.

· שלבים מקדימים:

1. לסדר את הדאטהסט

2. לאמת הנחות

1. מבצעים הטלה לכיוון ההתקדמות ועושים מבחן השערה לראות אם זה גאוסי או לא

2. עבור כל המדידות המכילות שיא גובה נבדוק תאוצה בכיוון אופקי

3. ביצוע של קלאסטרינג מגודל k משתנה לפי סטיית תקן ובדיקה האם מטרות מאותו סוג משוייכות לאותו קלאסטר

3. מסנן גס וראשוני ע"פ תחום של מהירויות (מינימום, מקסימום) וגובה מקסימאלי לסוג רקטה

4. פילוח של המדידות להבין כמה מדידות יש מכל סוג ובאיזה שלבי מעוף כדי לדעת על מה לשים משקל גדול יותר

5. לחפש נקודות של כיבוי מנוע (תאוצה בציר Z) – אולי ניתן להוסיף למודל של הירידה של הרקטה

6. לחפש דרך חכמה לראות בעיניים את הפיצ'רים בשלב באימון

7. לחפש דרך ליצור מסלול חכם של מטרה מסוג מסויים ולראות איך להלביש את המטרה עליו

8. האם המדידות הללו ממיקום כפי שנמדד? (עזה, לבנון ייתן לנו פילוח של רקטות שבנמצא)

9. יצירת בסיס למצגת (שלד)

· שיטת עבודה:

1. הגשה מהירה המכילה תוכה סינון ראשוני וחכם על מנת לקבל פידבאק ולכייון

2.

· שאלות לרפאל:

1. האם יש מדידות חסרות?

2. האם יש מידע על סטיית התקן של רעש המדידות?

3. מה הייתה צורת המדידה? רדאר, אופטי, כמה מכשירים וכו'?