RESEARCH UPDATE V

10-620 | Independent Study: Research | Andrea Klein

OVERVIEW OF CHANGES

- Recall that I have been attempting to learn the following two functions (separately) via SVR:
    - [<M200a, R200a, Mcmax, Rcmax>] -> average net velocity
    - [<M200a, R200a, Mcmax, Rcmax>] -> std. dev. net velocity
- This breakdown essentially implies that I am trying to learn a Gaussian distribution. At Hy's suggestion, I will re-parametrize the velocities as a Maxwell velocity distribution, which has a single parameter **a**:
    - [<Mass, R200a, Mcmax, Rcmax>] -> **a**
- I will show that the Maxwell distribution is a better fit to the halo velocities. **From now on, all SVR results will refer to a**. The essential machine learning task has not really changed, since we're just learning a quantity proportional to the mean (see next slide), but physically speaking it is a better-motivated way to characterize the problem.
- Recall that when using 1,000 data points, I was concerned that SVR was not learning the desired functions in a meaningful way (more details in later slides). I will present those results again in more detail, this time while trying to learn the Maxwell parameter.

MAXWELL DISTRIBUTION

A **Maxwell speed distribution** is meant to describe the magnitudes of the velocities of particles within a gas. The speeds follow a distribution of the form

$$f(v) = \sqrt{\left(\frac{m}{2\pi kT}\right)^3} 4\pi v^2 \exp\left(-\frac{mv^2}{2kT}\right)$$

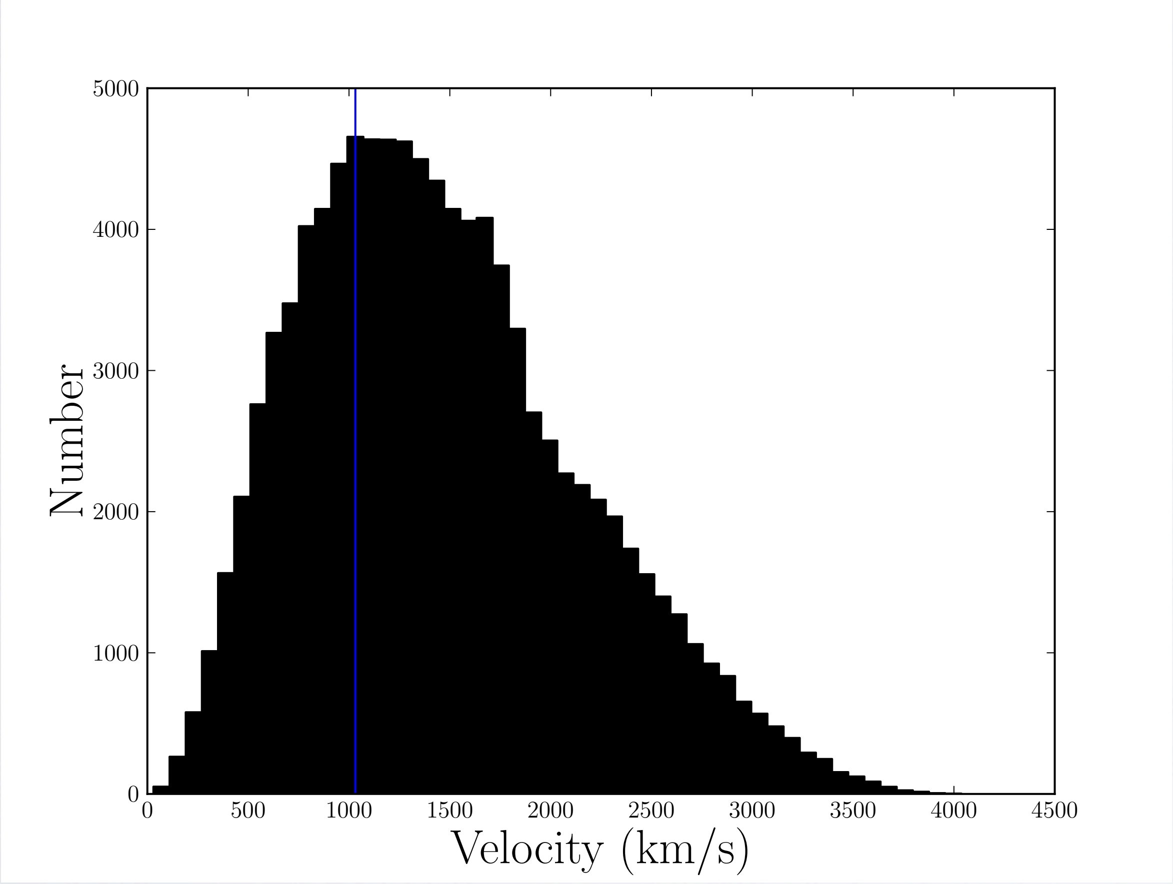
which is a Maxwell-Boltzmann distribution with its characteristic parameter **a** set to: $a = \sqrt{\frac{kT}{m}}$

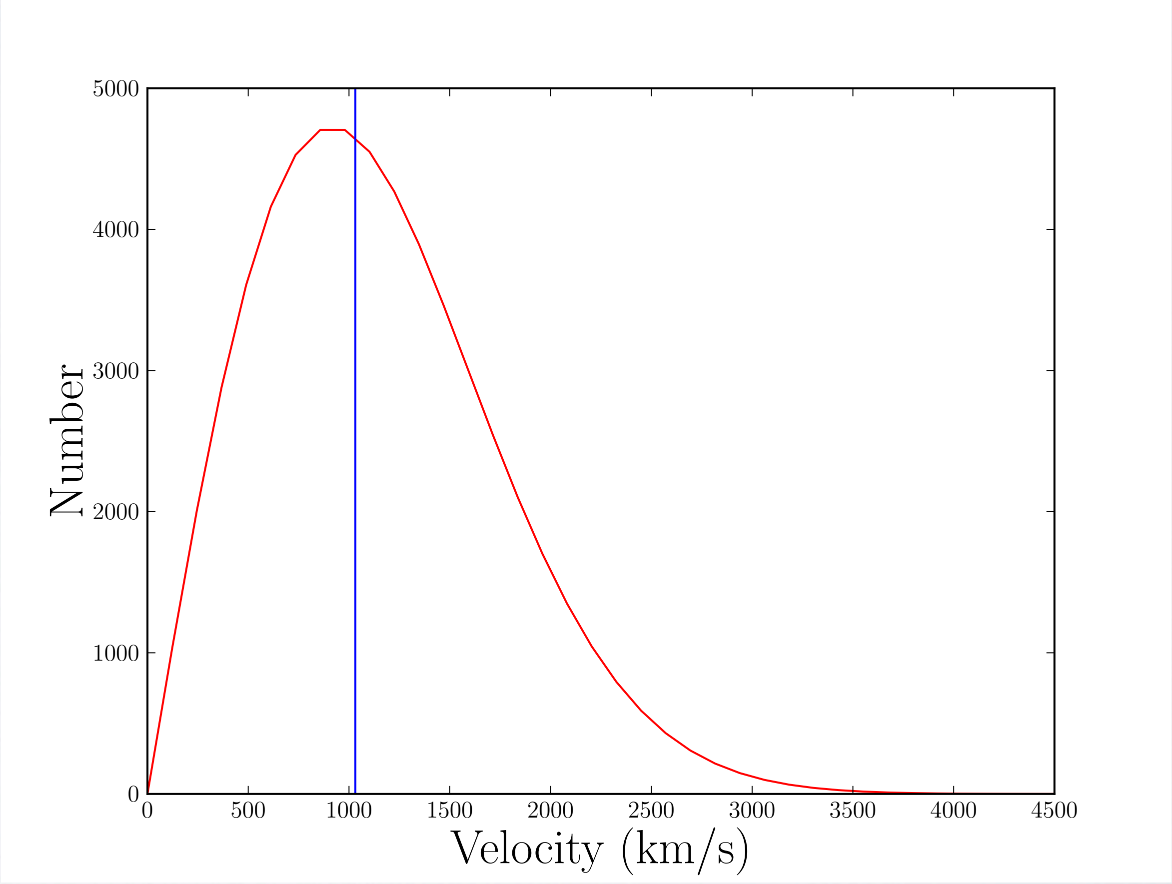
Since the average velocity is given by $\bar{v} = \left(\frac{8kT}{\pi m}\right)^{1/2}$,

i.e. **v** = sqrt(8/Pi) x **a**, we can trivially "fit" a sample of velocities by setting **a** = **v** x sqrt(Pi/8), where **v** is the sample mean.

MAXWELL DISTRIBUTION

- Example (subhalo 25):

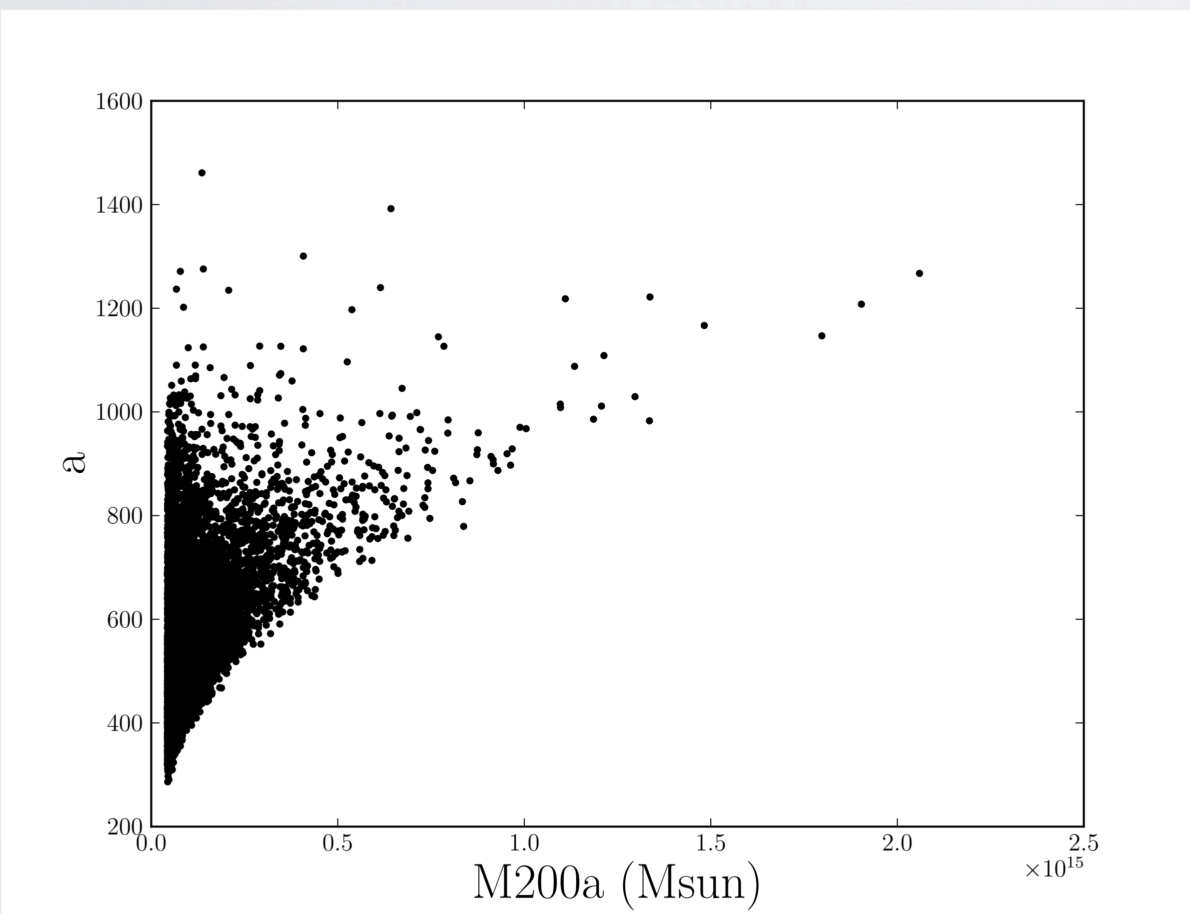


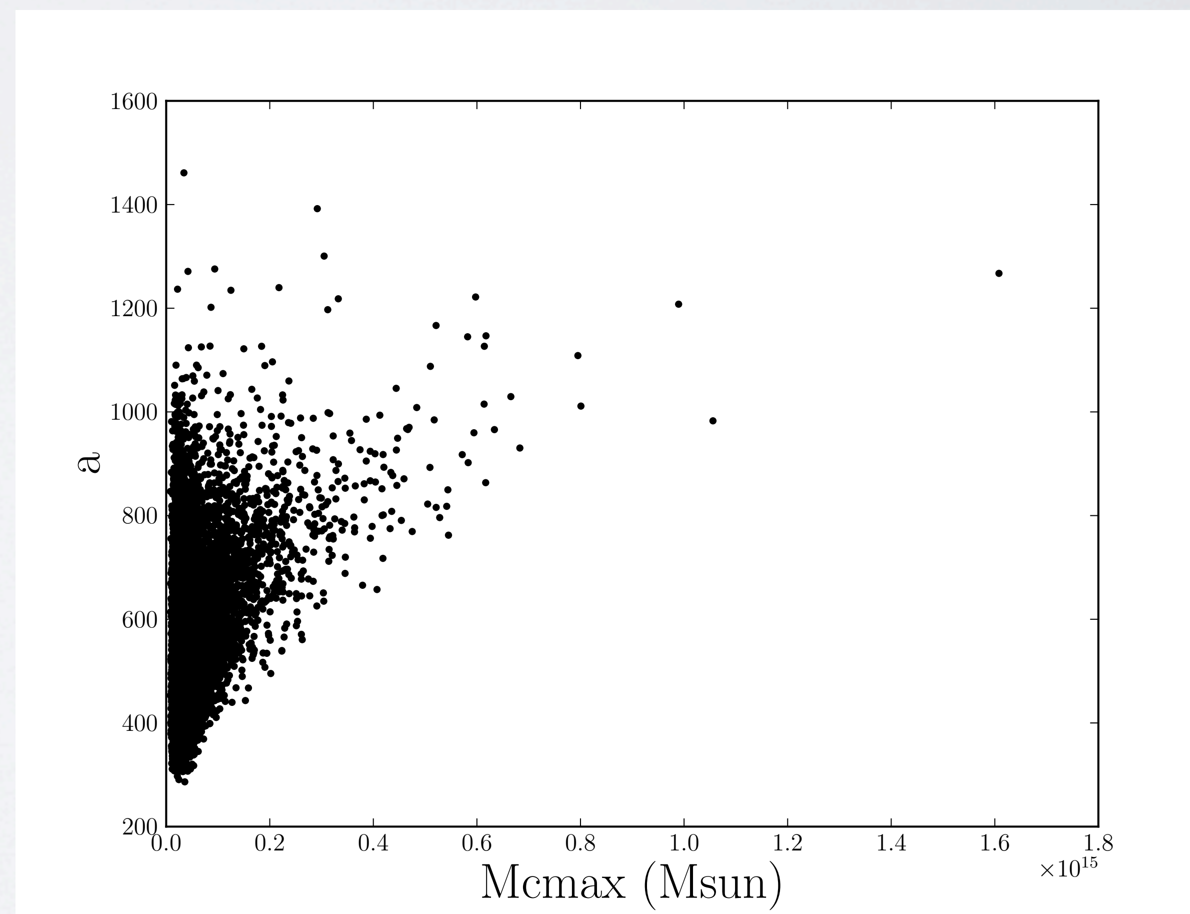


Note that this doesn't really need to be a perfect parametrization of the velocities. In particular, for a subhalo-based study, you'd presumably try to learn the velocity directly.

INPUT PARAMETERS

- A new look at the dependence of the target value (now **a**) on the input parameters (this time presented more reasonably with scatter plots), for 10,000 samples

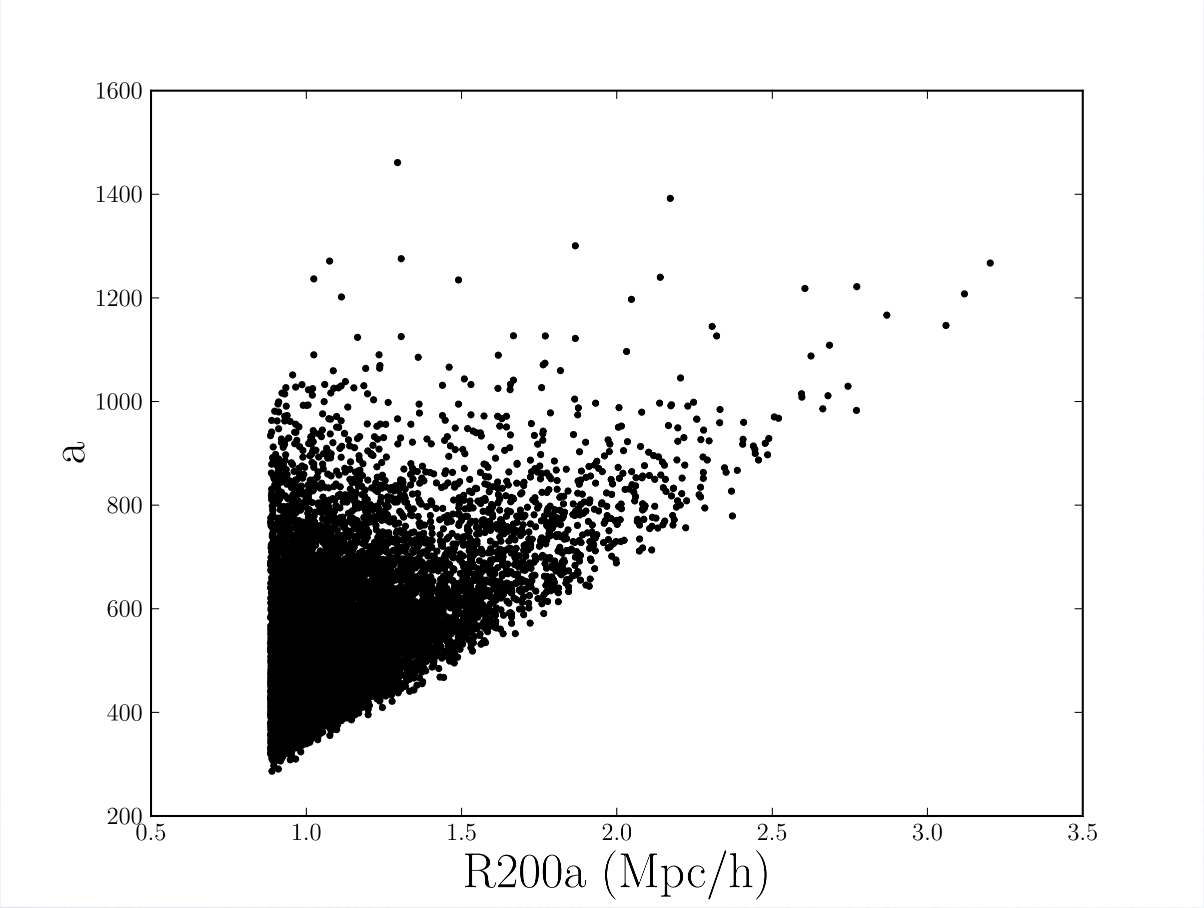


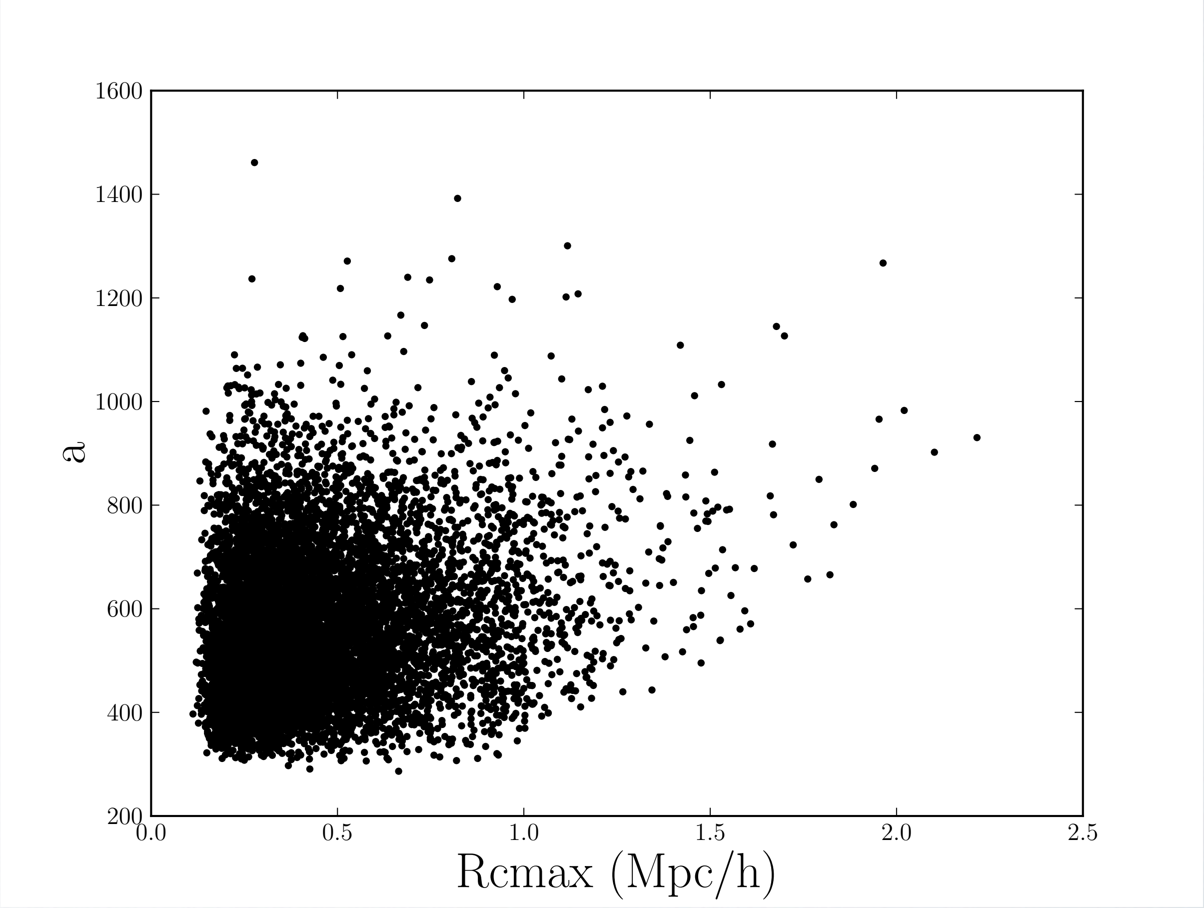


Note: in principle we expect **a** to vary ~mass^1/3, although there are several phenomena contributing to the large scatter (including mergers)

INPUT PARAMETERS

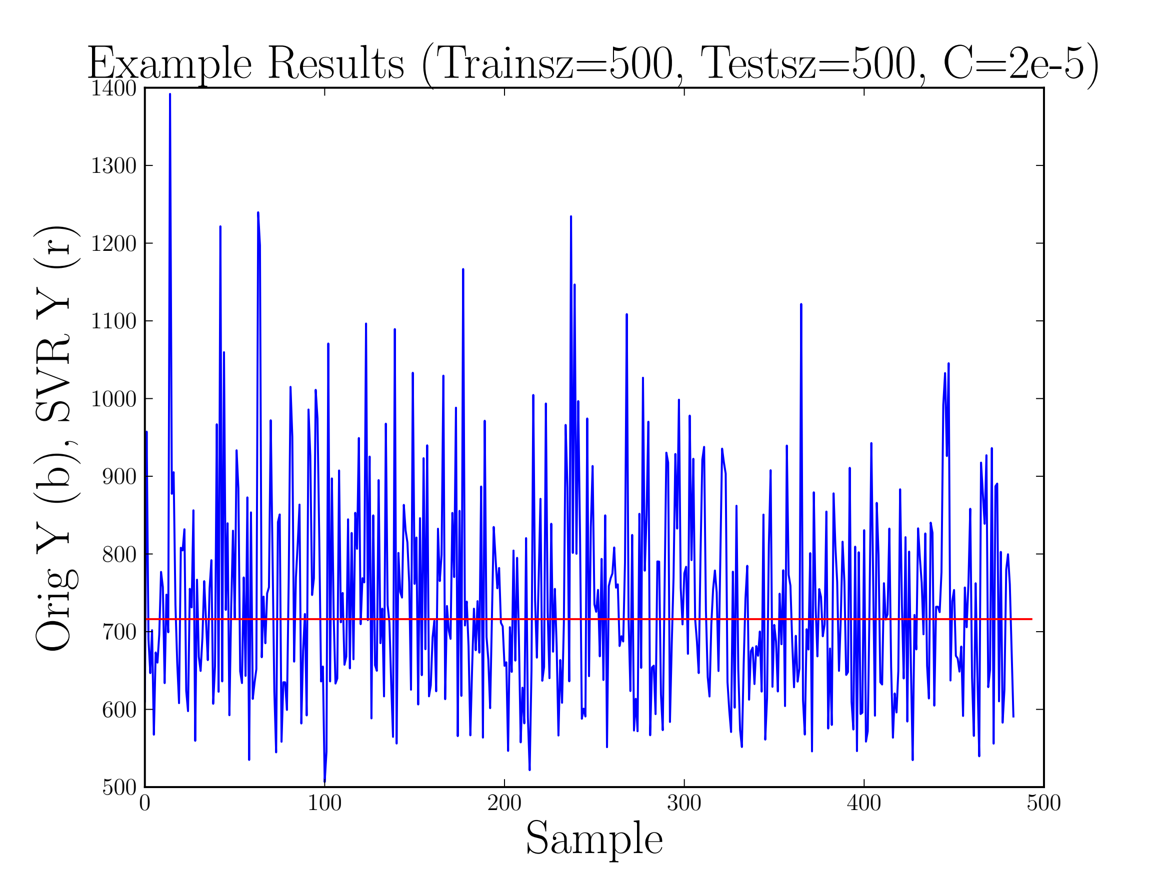
- A new look at the dependence of the target value (now **a**) on the input parameters (this time presented more reasonably with scatter plots), for 10,000 samples

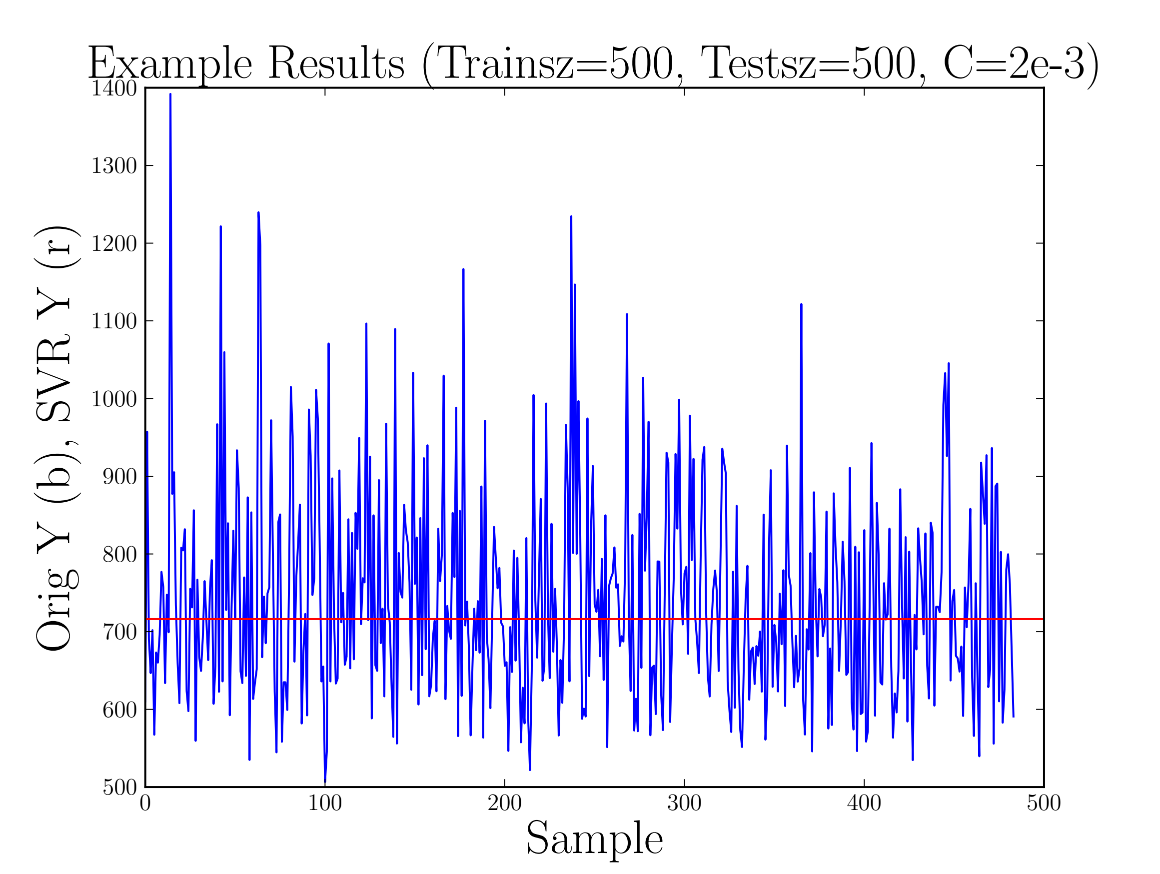




RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).

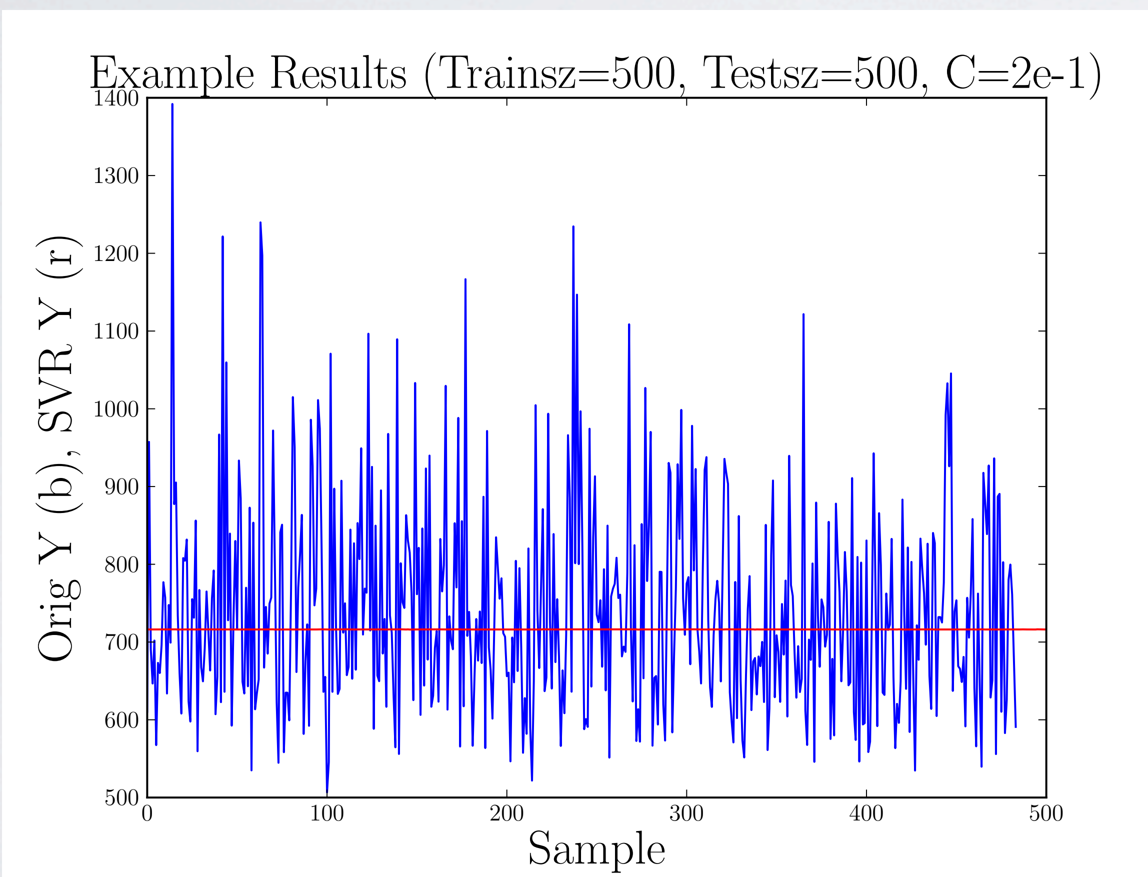


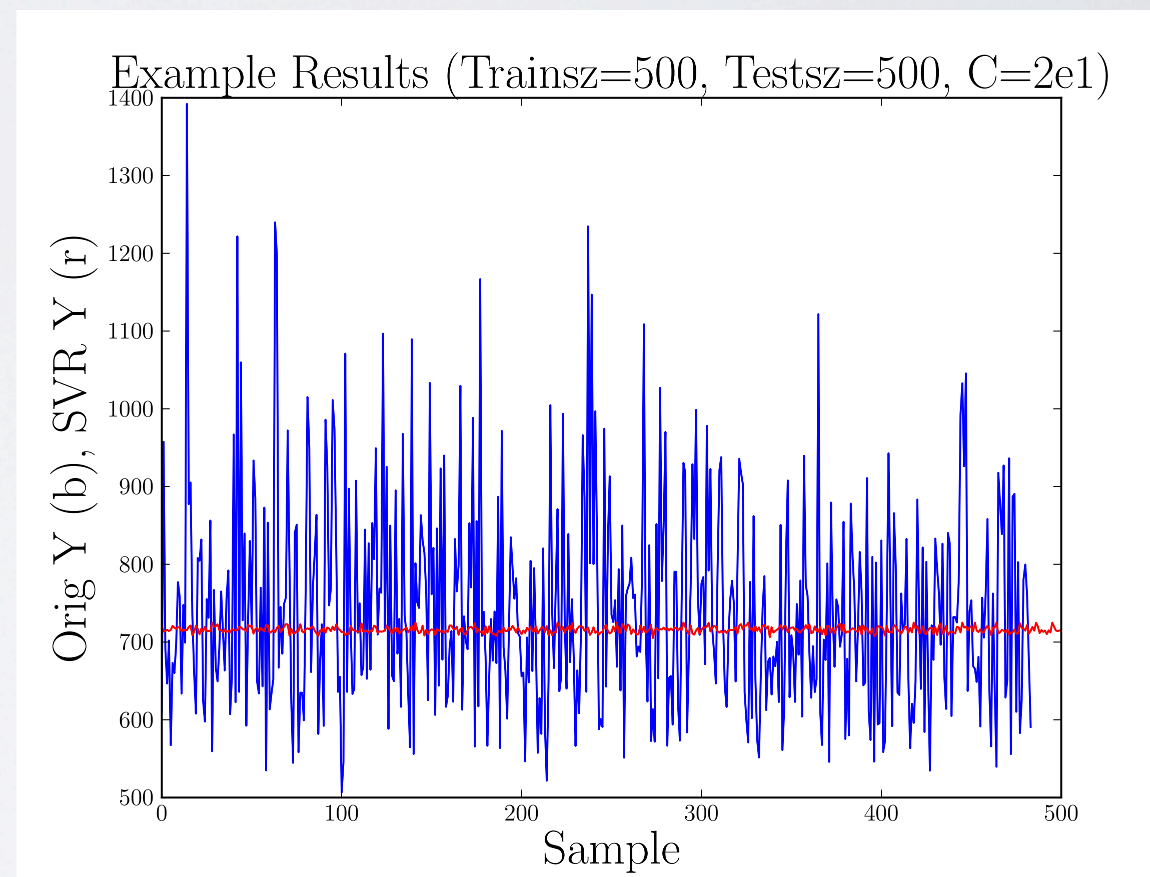


Small C -> SVR basically just predicts a constant

RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).

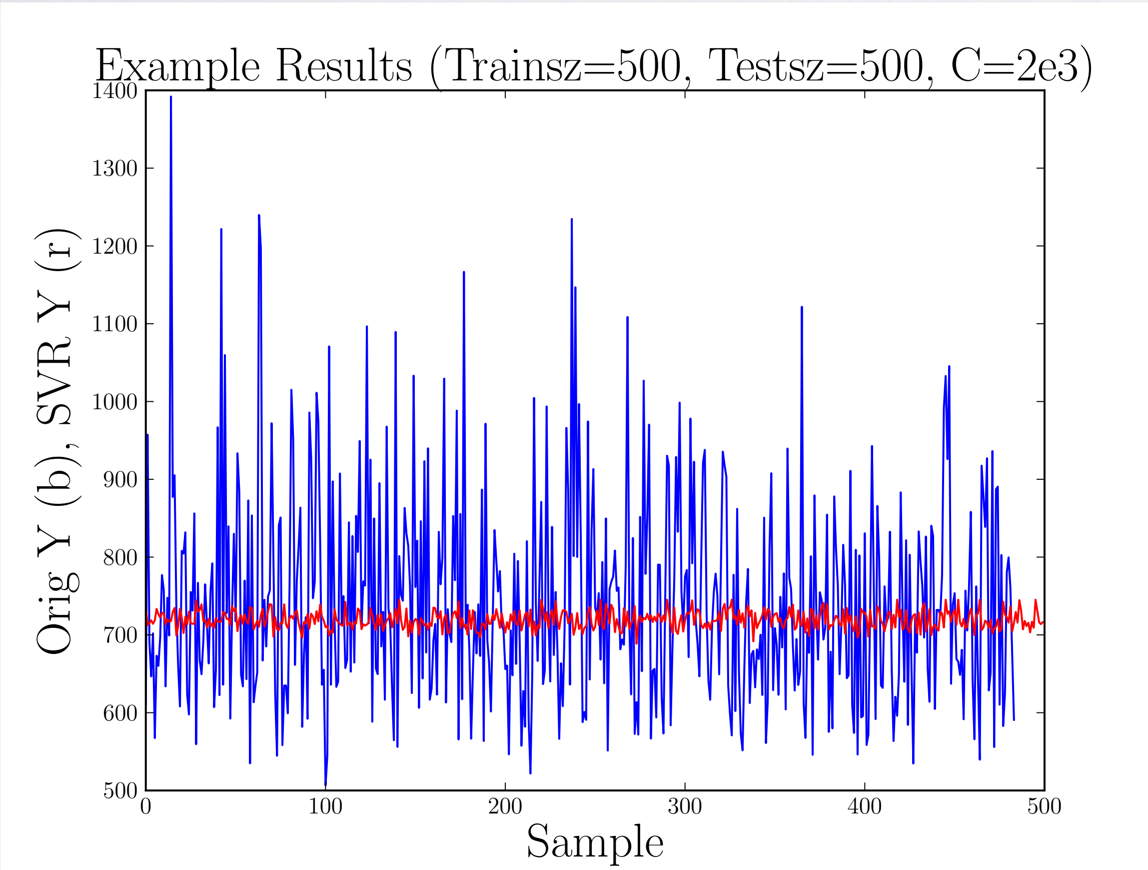


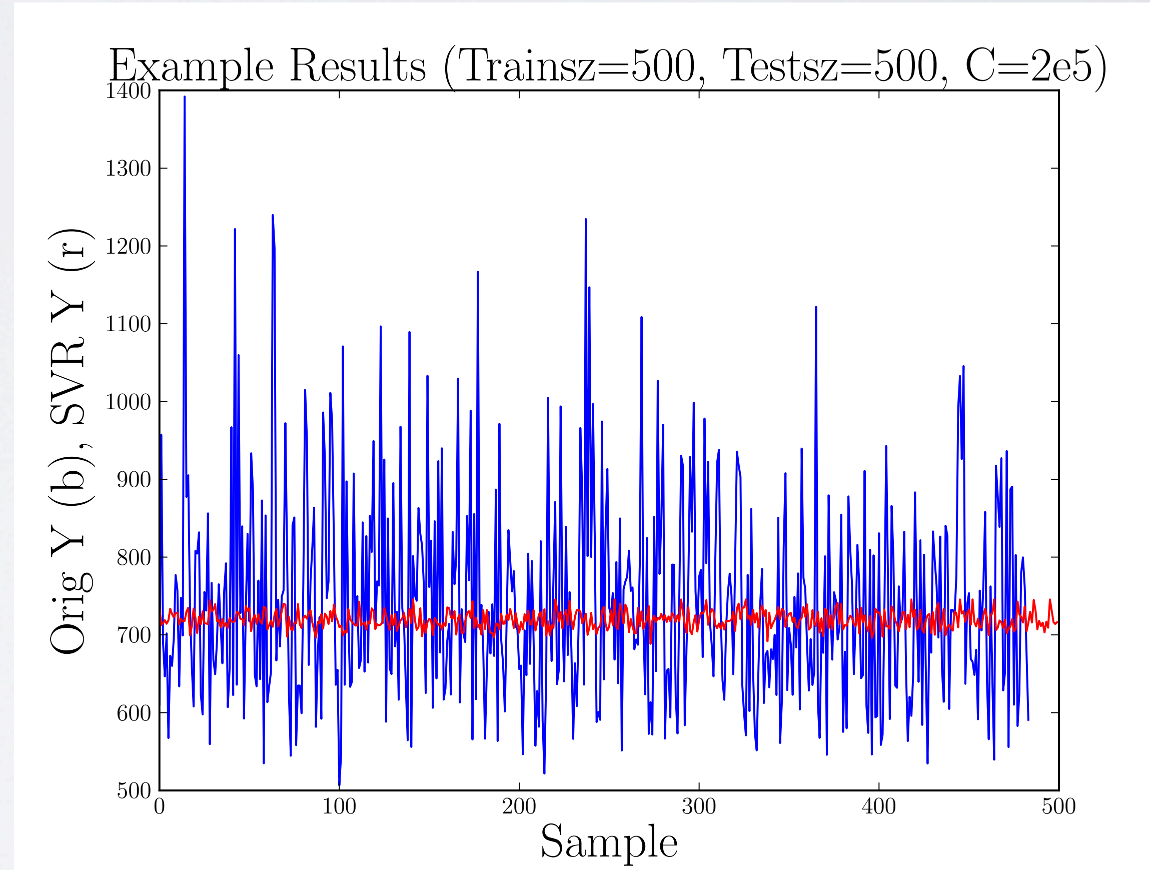


Small C -> SVR basically just predicts a constant

RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).

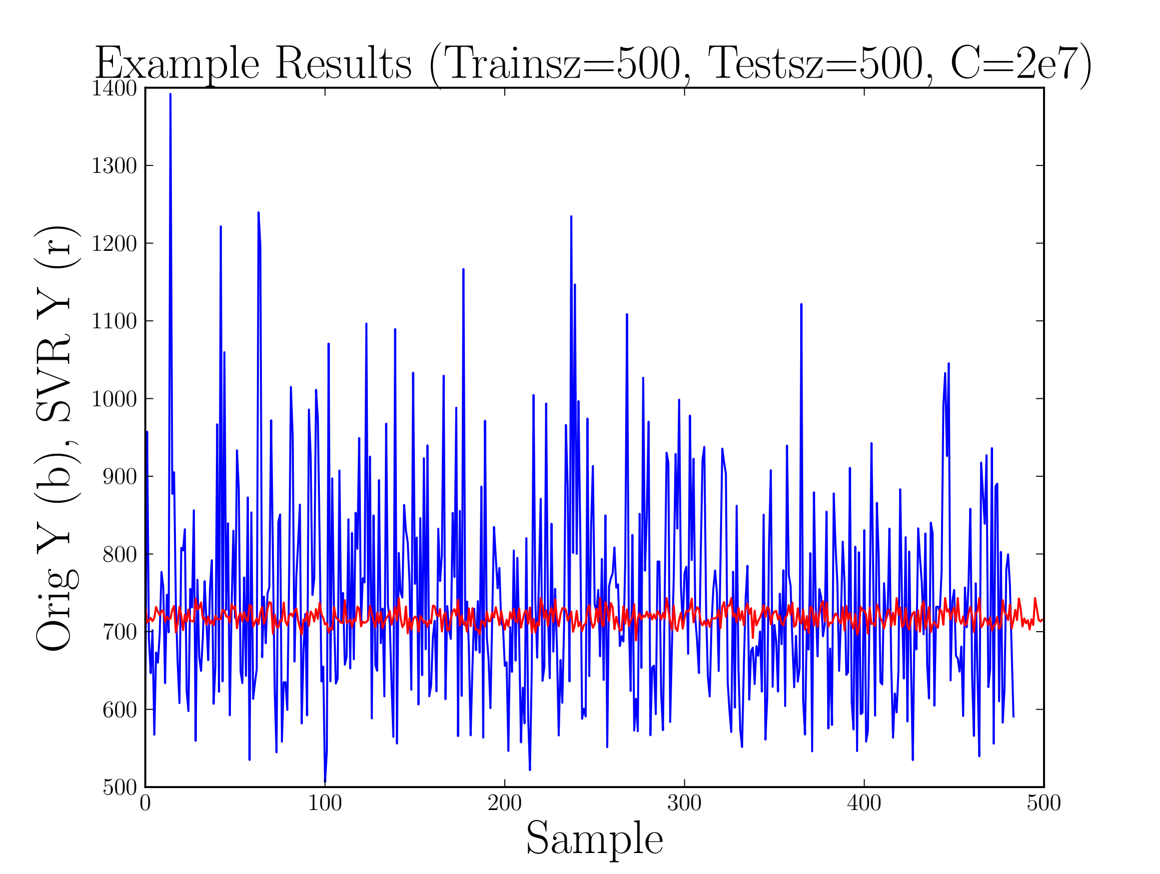


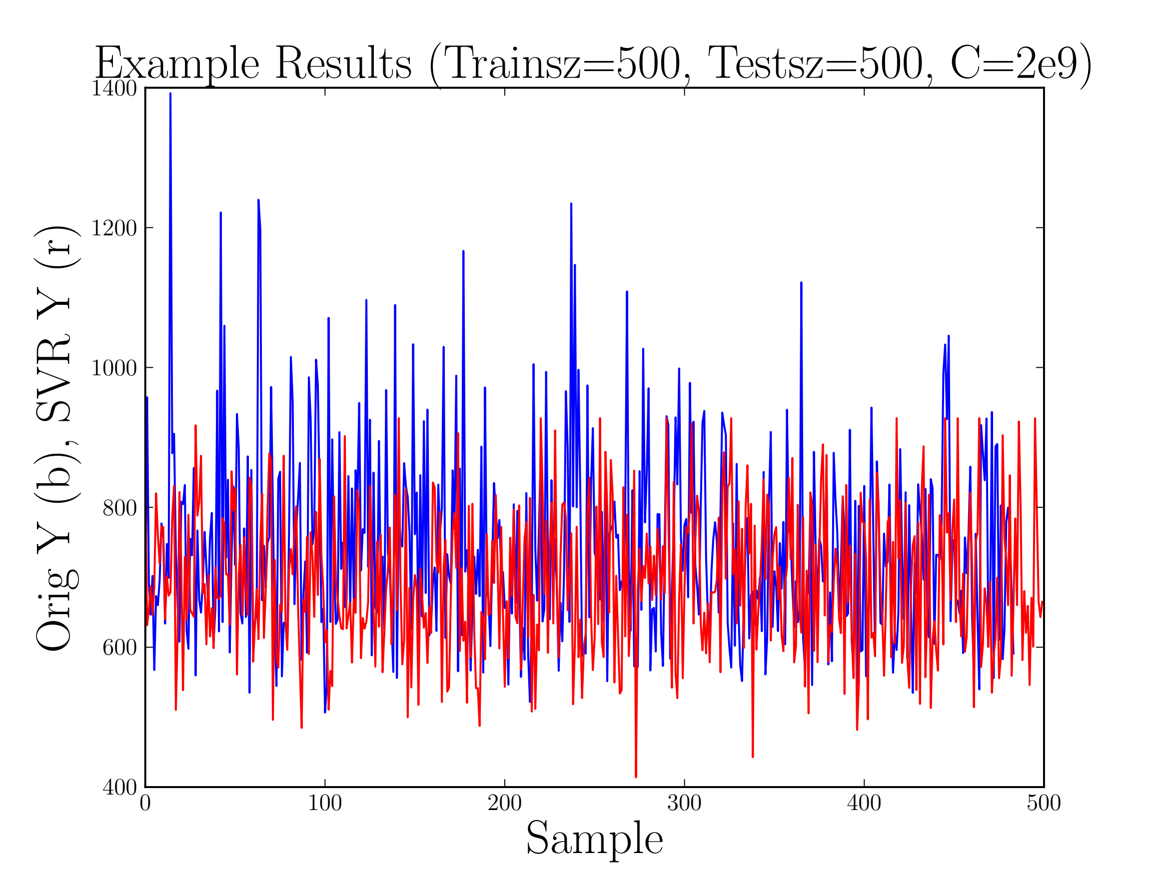


Larger C -> some structure apparent...

RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).

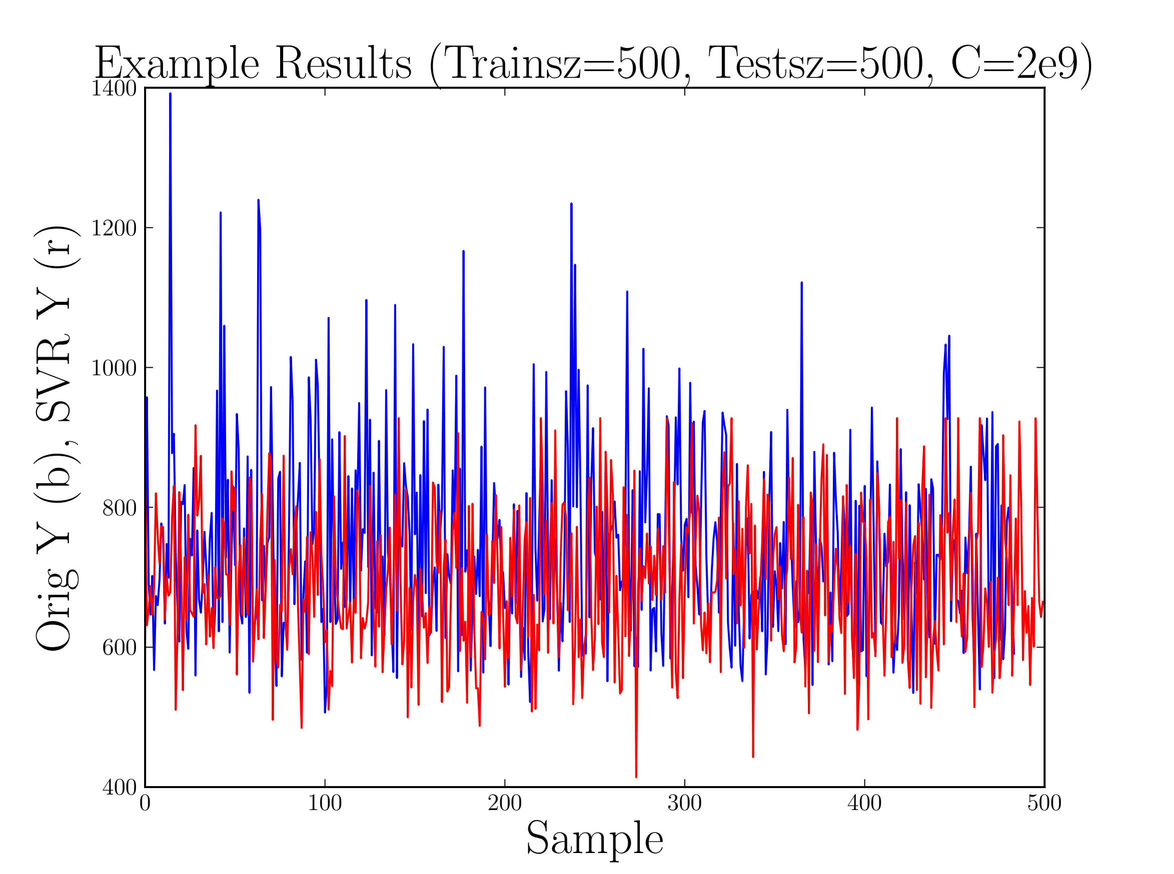


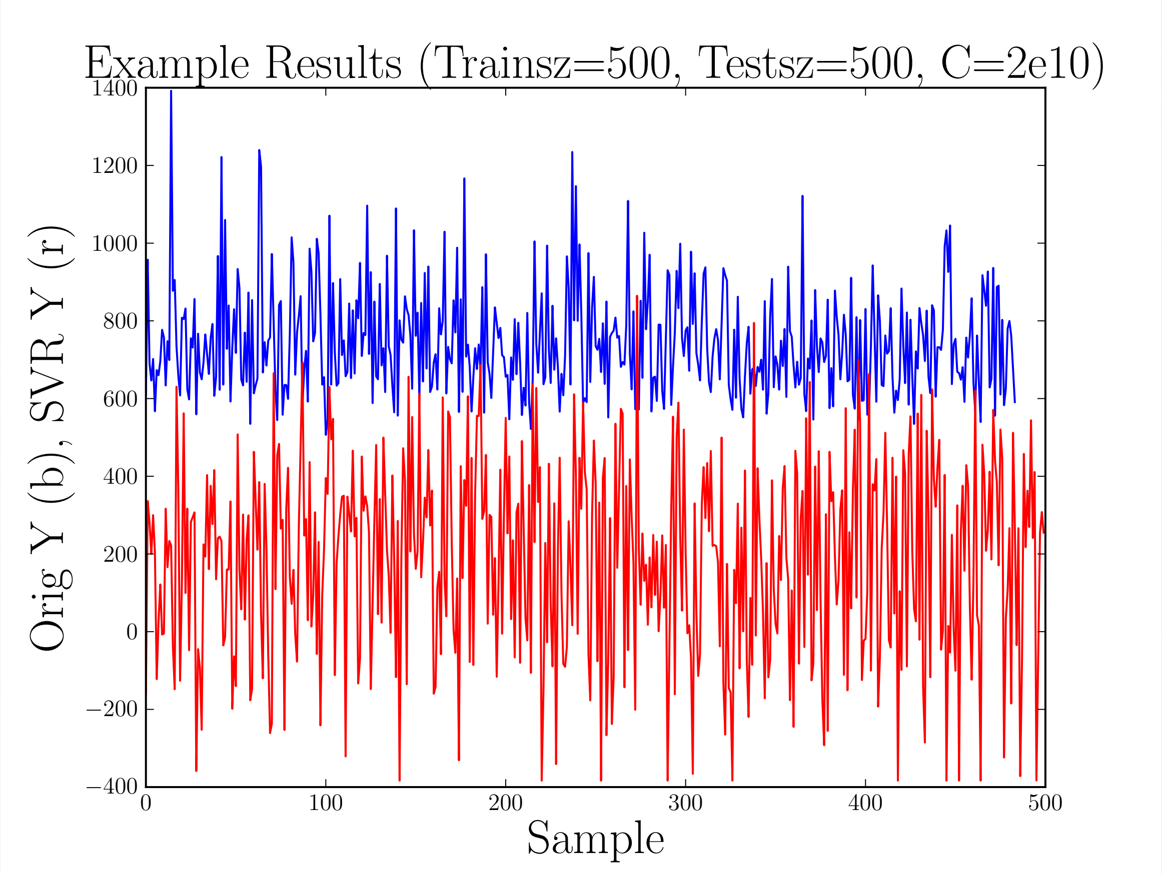


Reasonable results somewhere around C = 2e9

RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).

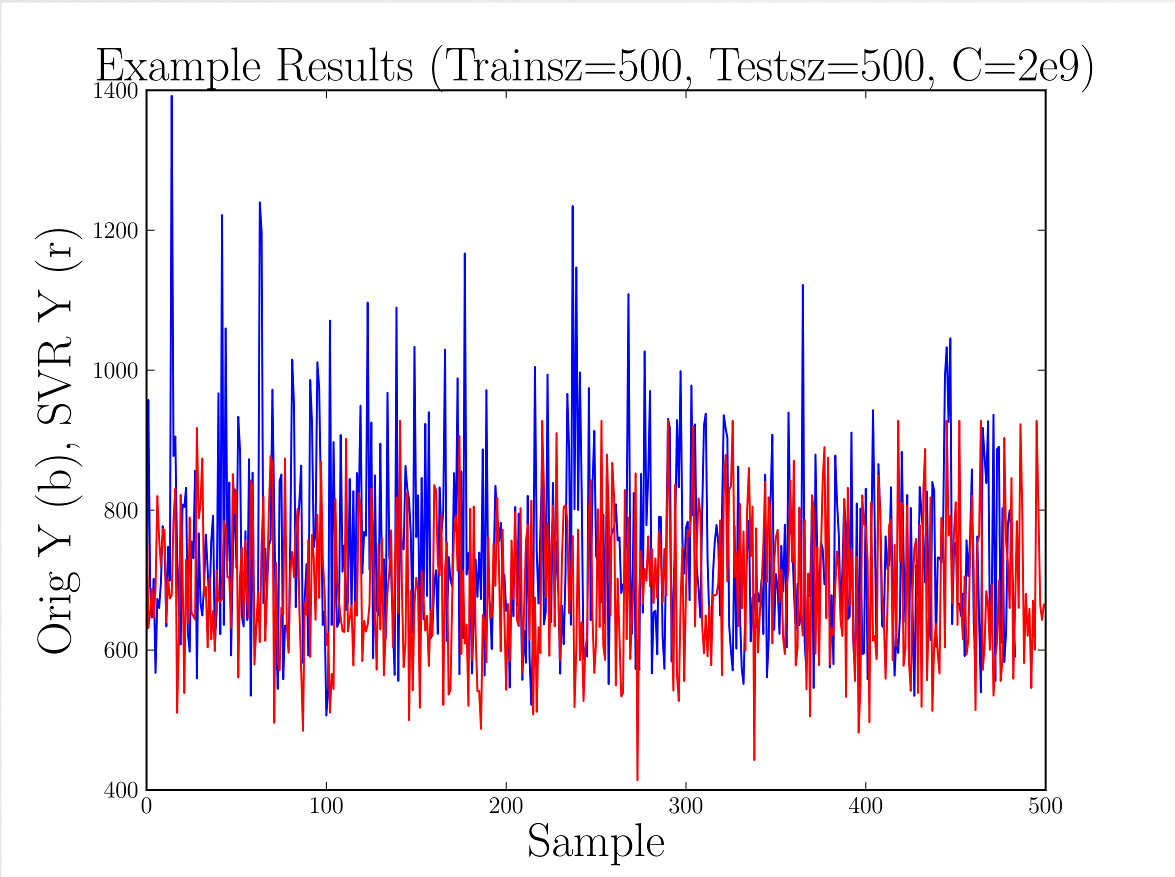


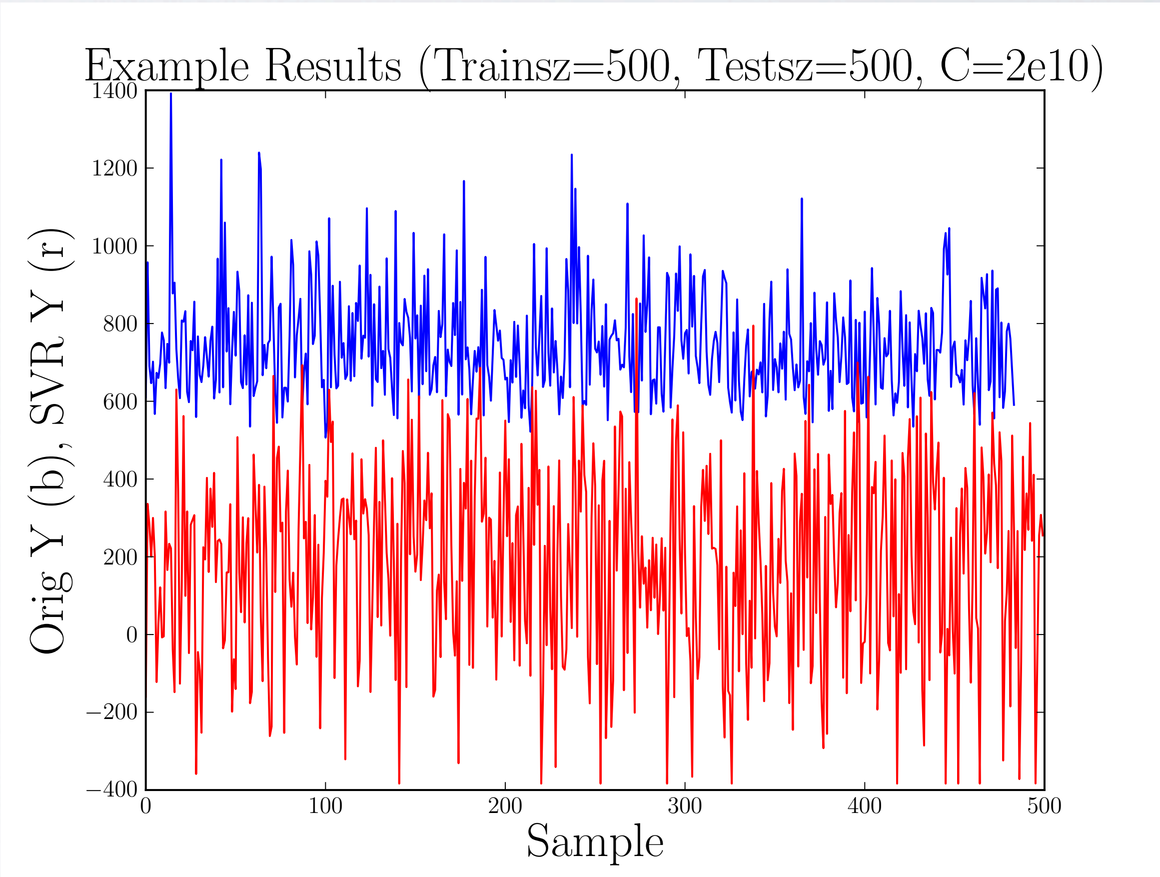


Much larger than C=2e9, though, and things get bad again

RESULTS

- Example results for the linear kernel. Note that the input data was shuffled during creation (i.e. samples are not in mass-increasing order).





Much larger than C=2e9, though, and things get bad again

RESULTS

- How does the RMSE vary with C? ...it doesn't (at least not significantly).

FUTURE WORK

- At the moment, the RMSE error is not terribly informative as to the performance of the SVR. This is presumably due at least in part to the large scatter in **a**. One solution may be to filter the input data more aggressively, i.e. by measuring its conformance with the best-fit Maxwell distribution and rejecting data that fits poorly. This should remove halos with recent mergers (doubly-peaked velocity histograms), for example, which should reduce the scatter and make RMSE more informative.

- Ultimately the subhalo velocities (or dispersions) are probably the more interesting quantity to learn, so it may not be necessary to expend too much energy optimizing the code to learn particle velocities. In my mind, the eventual goal of this work is to re-introduce radial position as an input quantity, learn subhalo velocities without a parametrization of the halo velocity distributions, and use the resulting classifier to "paint" halos whose subhalos have been generated via e.g. the algorithm in my paper (i.e. without recourse to N-body simulation). It may also be possible to learn the internal subhalo velocity distribution directly, rather than trying to learn a classifier for individual subhalos.