Analysis of High-Dimensional Estimation Techniques

Mohamad Lakkis

December 23, 2024

1 Section: The Inadmissibility of the Sample Mean in High-dimension and the James-Stein Estimator

1.1 Part 1: A graphical illustration.

What is it that we are shrinking?

In short we are shrinking the sample mean$(\hat{\mu}_1, ..., \hat{\mu}_p)$, towards the origin.
I got really intersested in this topic, I found a very nice nice and intuitive explanation on the following link: `https://www.youtube.com/watch?v=cUqoHQDinCM`
Even though individual coordinates might be accurate($\hat{\mu}_1$ might be very close to μ_1), their collective magnitude tends to be large.
In high-dimensional space, the variance of the estimation grows quickly as the dimension increases. Shrinking the estimator towards the origin reduces this variance significantly. (by introducing some bias)
However, in high dimensions, most of the data points lie far from the origin due to the nature of high-dimensional geometry.
From the formula we see that the shrinkage is reverse proportional to the distance from the origin.(so further the point from the origin the less it gets shrunk)
This differential shrinkage reflects the statistical trade-off between bias and variance:
For near-field points, the sample mean is highly uncertain, so aggressive shrinkage reduces variance at the cost of some bias.
For far-field points, the sample mean is more reliable, and less shrinkage ensures that the estimator does not introduce unnecessary bias.
That is why the James-Stein estimator is preferred over the sample mean in high-dimensional space. (especially in cases where the true mean is close to the origin)
This is in essence one justification of why we use lasso, or ridge regression in high-dimensional cases, because these methods introduce some bias to reduce the variance of the estimator, by shrinking the coefficients towards zero.(STAT 239)

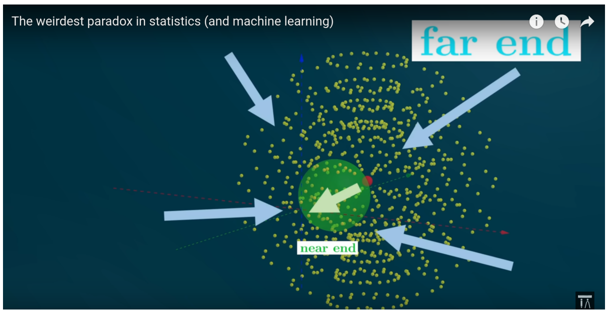


Figure 1: James-Stein Estimator, and its effect on near-field and far-field points. (Taken from the video link above)

Graphical Illustration

Note Regarding the spherical symmetry:
This means the James-Stein estimator shrinks $\boldsymbol{X}$ along its own direction by a factor of $g(\|\boldsymbol{X}\|)$. And so the MLE $\boldsymbol{X}$ lies at the observed point, and the James-Stein estimator lies on the line connecting the origin and the observed point. But it is more closer to zero, whcih will reduce the variance as discussed previsouly!
How we can visaualize the problem of ocmparing MLE and JS in 2D:
The random vector $Z = (X_1, \|U\|)$, where X_1 represents the observed point and $\|U\|$ summarizes the "noise" in the orthogonal direction.
And so in this $Z - plane$ the comparison will take place, having the MLE at $(X_1, \|U\|)$, and the $\delta_{JS}(\boldsymbol{X})$ shrinks the MLE towards the origin.
Note: the hyperplane perpendicular to μ corresponds to the subspace spanned by the remaining $p-1$ coordinates. (so the magnitude $\|u\|$ tell us how far $\boldsymbol{x}$ lies in the orthongal space), and so as we were told in the question (The symmertry in spherical ot depends on the total distance, not on specific direction of u).
Simply said the direction of u does not affect the shrinkage because the estimator treats all directions equally.
Setup:
$X_1 \sim N(\theta, 1)$ and $\|U\| \sim \chi^2_{p-1}$ indepedent rvs.
In order to get the "center" of the distribution in the z-space, which is the point $(E(X_1), E(\|U\|))$, we need to calculate both of these expectations
Calculating $E(X_1)$:
Straightforward, since $X_1 \sim N(\theta, 1)$, and so $E(X_1) = \theta$
Calculating $E(\|U\|)$:
We know that $\|U\|^2 \sim \chi^2_{p-1}$ and that $\|U\| \sim \chi_{p-1}$, and so we have from [1] $E(\|U\|) = \sqrt{2}\frac{\Gamma(\frac{p}{2})}{\Gamma(\frac{p-1}{2})}$

Additionally we need to get the interception point (x,y)between the perpendicular line (the one shown in blue) from the line $((\theta, 0)\ to\ (x, y))$ to the line connecting the origin and the point $(E(X_1), E(\|U\|))$
Simple math(staritng from the first slope which is $\frac{E(\|U\|)}{\theta}$, and getting the second slope which is the negative reciprocal $slope_2 = -\frac{-\theta}{E(\|U\|)}$, ...), we get: $x = \frac{\theta^3}{E(\|U\|)^2+\theta^2}$ and $y = \frac{\theta^2 E(\|U\|)}{E(\|U\|)^2+\theta^2}$

Simulation Results, and analysis of the results:

- As ”p” grows the shrinkage becomes more and more important, which is shown how the $\delta_{JS}(X)$ are closer to the **TRUE** mean, and the MLE are further away.
- We can see how the James-Stein estimator, shrinks the MLE towards the origin, reducing the variance, by introducing some bias.
- We can see how the perpendicular line from $(\theta, 0)$ to the red line goes directly into the black clouds (the JS estimator points), which means that the black cloud is closer to the True mean, meaning JS estimators are better, especialy for large p.
- Additionally, I added the green lines, these green lines, just show that indeed the $\boldsymbol{X}, \delta_{JS}(\boldsymbol{X}), \delta_0(\boldsymbol{X})$, are on the same line, as discussed in the spherical symmetry.
- Another mind blowing thing, is that we can clearly see how this validates equation 1, in the question, we can see that in the triangle formed by the red line, the blue line and the black line. WE can clearly see how the the hypotenuse (the black line) is the longest distance, which represents the difference between the MLE predicitions and the true mean (shown in red) this phenonmenon is more clear in higher dimensions, such as p = 1000 which is very clear in 5. So in short the Risk between the MLE and the True mean is largeer than the risk between the JS estimator and the True mean, especially fir higher dimension, which is the whole point of the James-Stein Estimator.

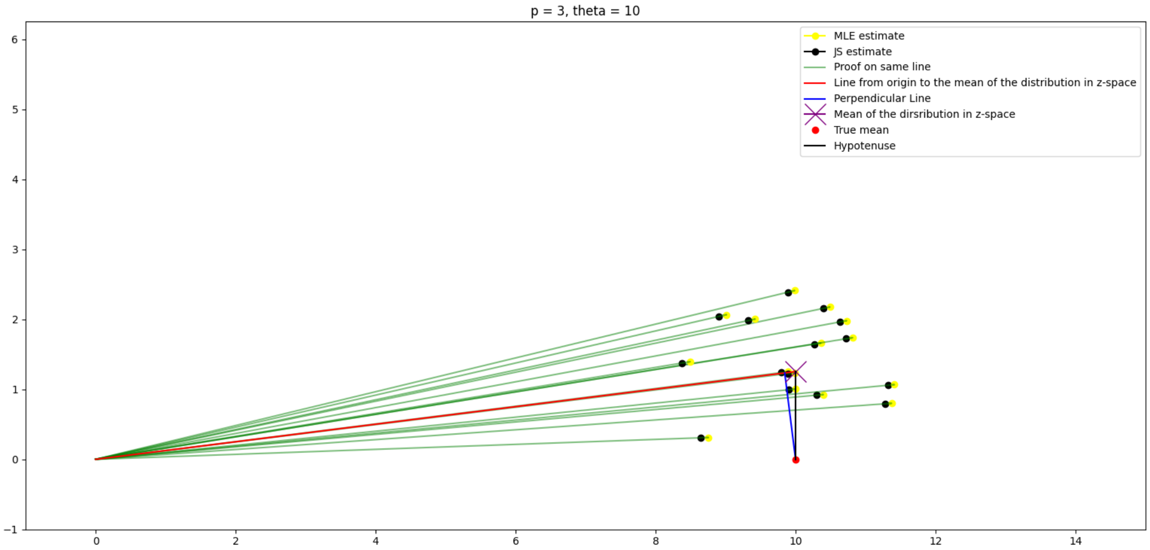


Figure 2: p = 3: Code Available in graphical_illustration.py

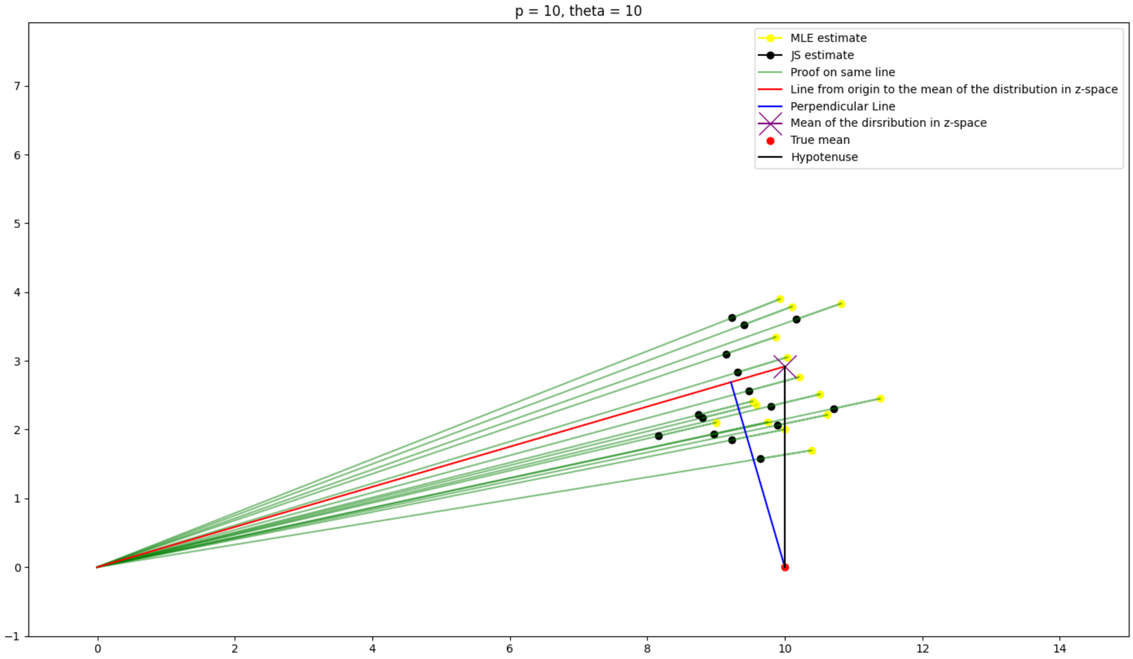


Figure 3: p = 10: Code Available in graphical_illustration.py

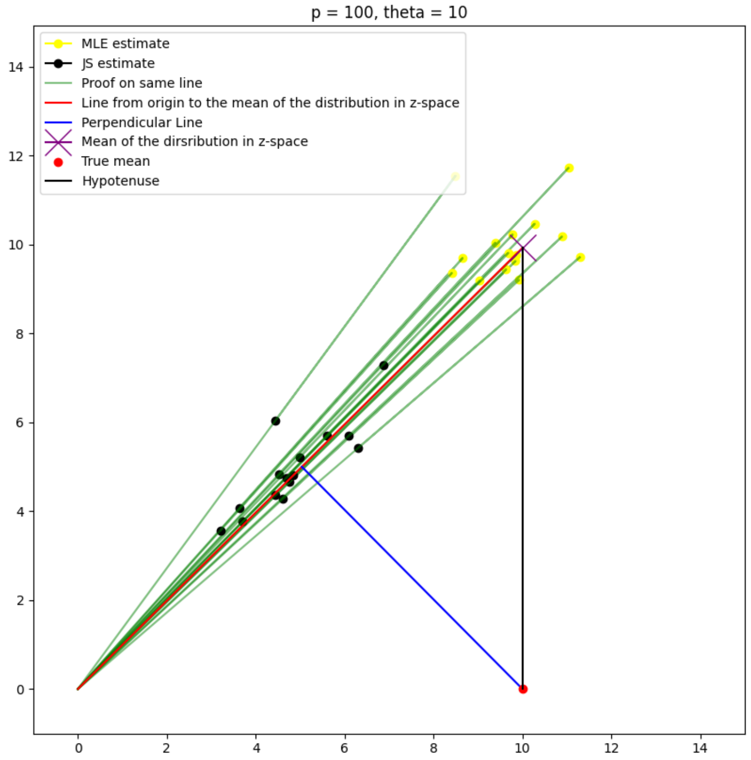


Figure 4: p = 100: Code Available in graphical_illustration.py

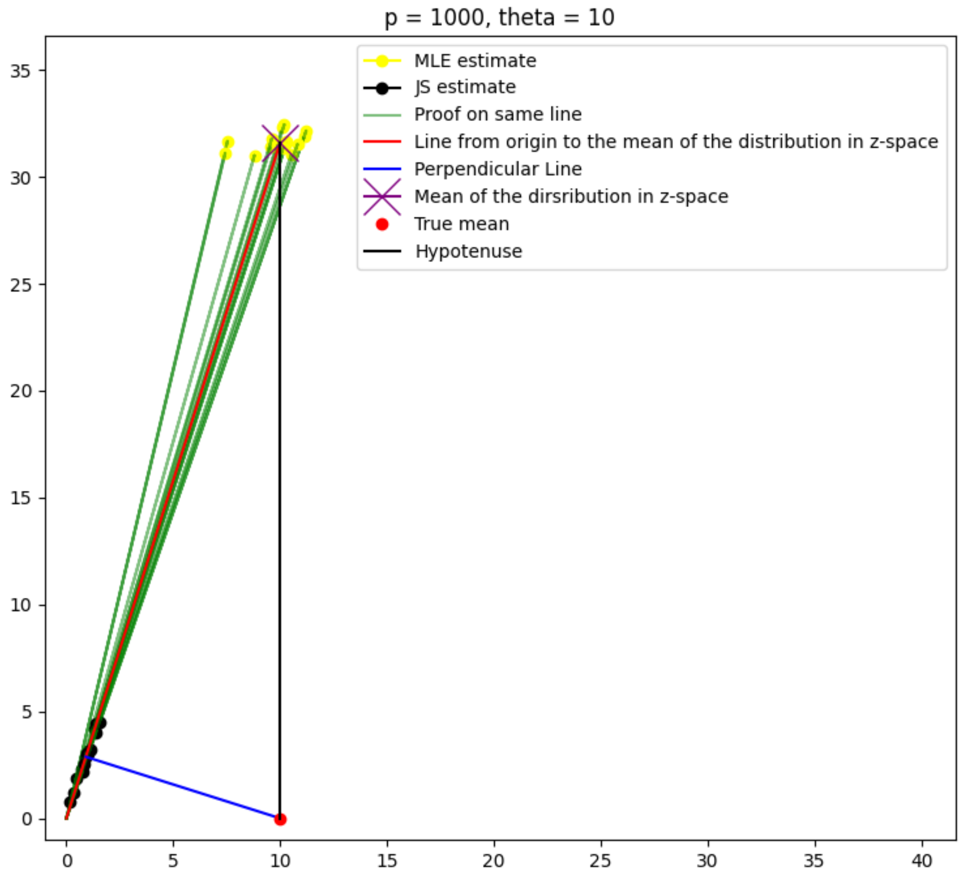


Figure 5: p = 1000: Code Available in graphical_illustration.py

1.2 Monte Carlo Risk Estimates

So what I did is I fixed my $\theta = 10$, and then took many samples of $\boldsymbol{X}$, to make the average of each term, and then I calculated the 3 quantities provided in Equation (1) in the question. But as mentioned here directly I am working in $\mathbb{R}^p$
First let us validate our reasoning from the previous prt to compare the risk in estimating the true mean using the MLE and the James-Stein Estimator.

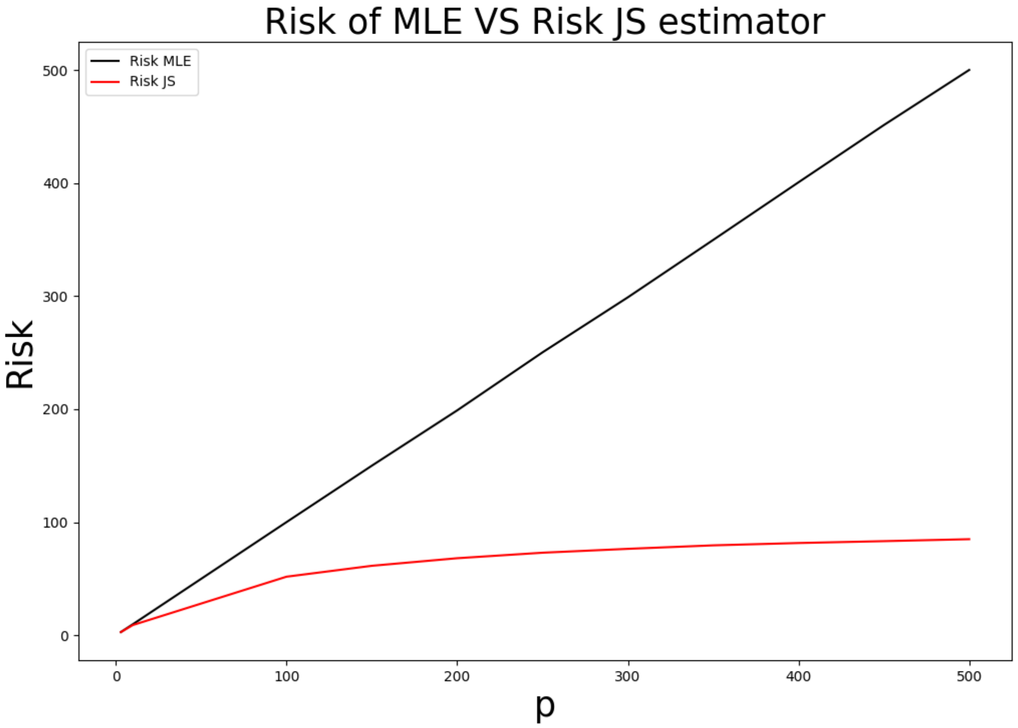


Figure 6: Monte Carlo Risk Estimates: Code Available in comparison_MC.py

We can see how the red curve in 6 which represents the risk of the JS, is always lower than the risk of the MLE(consistent with what we found earlier in the first part of this question), notic how as p grows the risk of the MLE grows significantly, while the risk of the JS remains relatively constant. Which is the whole point of JS increasing a little the bias (by shrining them towards 0) while reducing significantly the variance

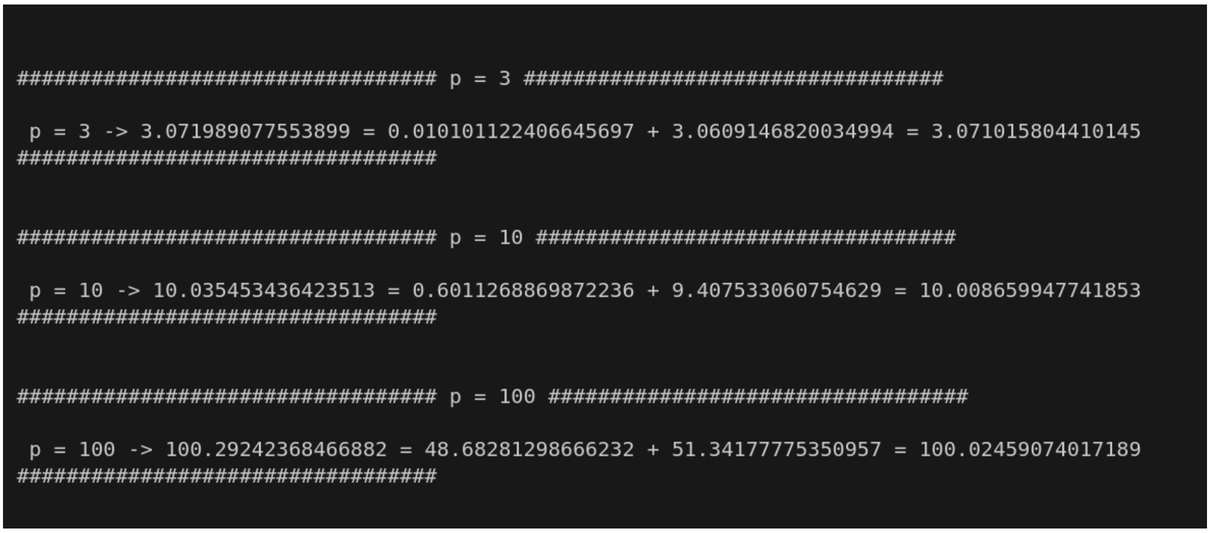
2. Additionally, we can see how the Risk of MLE: is exactly equal to the Variance of $\boldsymbol{X}$, since

$$E[||\boldsymbol{X} - \boldsymbol{\mu}||^2] = E[(X_1 - \mu_1)^2] + \cdots + E[(X_p - \mu_p)^2] = 1 + \cdots + 1 = p$$

So we can clearly see how the black line is the line $y = x$.

And this indeed show that the MLE is inadmissible in high dimensions, since the risk of the MLE is always larger than the risk of the JS estimator.

Verification of Pythagorean Theorem:



```
################################## p = 3 ##################################

 p = 3 -> 3.071989077553899 = 0.010101122406645697 + 3.0609146820034994 = 3.071015804410145
##################################


################################## p = 10 ##################################

 p = 10 -> 10.035453436423513 = 0.6011268869872236 + 9.407533060754629 = 10.008659947741853
##################################


################################## p = 100 ##################################

 p = 100 -> 100.29242368466882 = 48.68281298666232 + 51.34177775350957 = 100.02459074017189
##################################
```

Figure 7: Pythagorean Theorem: Code Available in comparison_MC.py

From this figure we can clearly see that indeed the hypotenuse (the black line) is the longest, and the sum of the squares of the other two sides (the red and the blue) is equal to the square of the hypotenuse.
Which vefiries this equation(along some MC errors):

$$E[\|\boldsymbol{X}-\boldsymbol{\mu}\|^2] = E[\|\boldsymbol{X}-\delta_{JS}(\boldsymbol{X})\|^2] + E[\|\delta_{JS}(\boldsymbol{X})-\boldsymbol{\mu}\|^2]$$

Additionally, from this term $E[\|\boldsymbol{X}-\delta_{JS}(\boldsymbol{X})\|^2]$ from the values in 7 we can see how it is becoming larger as p grows, showing how the JS estimator is drifting away from the MLE, as shown in the previous figures (such as 5).

Clarification of the meaning of $\delta_{JSO}(\boldsymbol{X})$

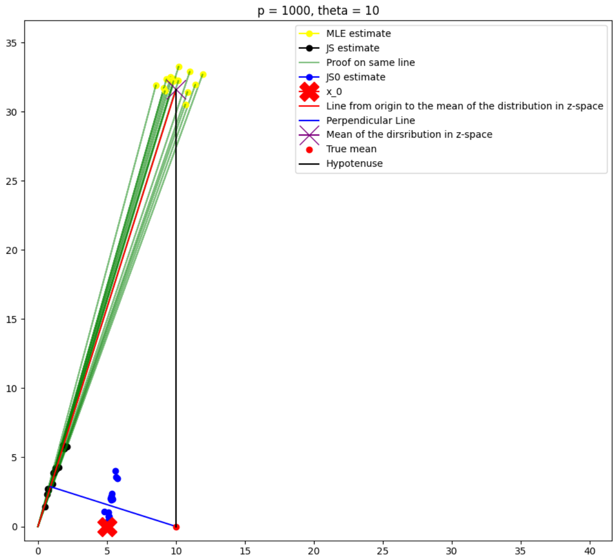


Figure 8: JSO: Code Available in JS0.py

So from this figure what is added compared to 5 is that we added the $\delta_{JSO}(\boldsymbol{X})$ estimators in "blue", we can see how they are drifting towards NOT the origin rather the point x_0 which we designated as $(\theta - 5, 0, ..., 0)$, just to illustrate the point but this works if I use other x_0 as well.

Where can we use $\delta_{JSO}(\boldsymbol{X})$?

We can use $\delta_{JSO}(\boldsymbol{X})$ in cases where we have some prior information about the true mean, and we want to shrink the MLE towards this point, rather than the origin.

In such cases we can expect that the risk of $\delta_{JSO}(\boldsymbol{X})$ will be lower than the risk of the MLE, and the JS estimator.(Given that x_0 is somehow closer to the true mean than the origin (or in other cases like we discussed in your office, so I wasn't able to check if this is the case for any x_0))

The case where $\delta_{JS0}(\boldsymbol{X})$ does better is illustrated in the below figure:

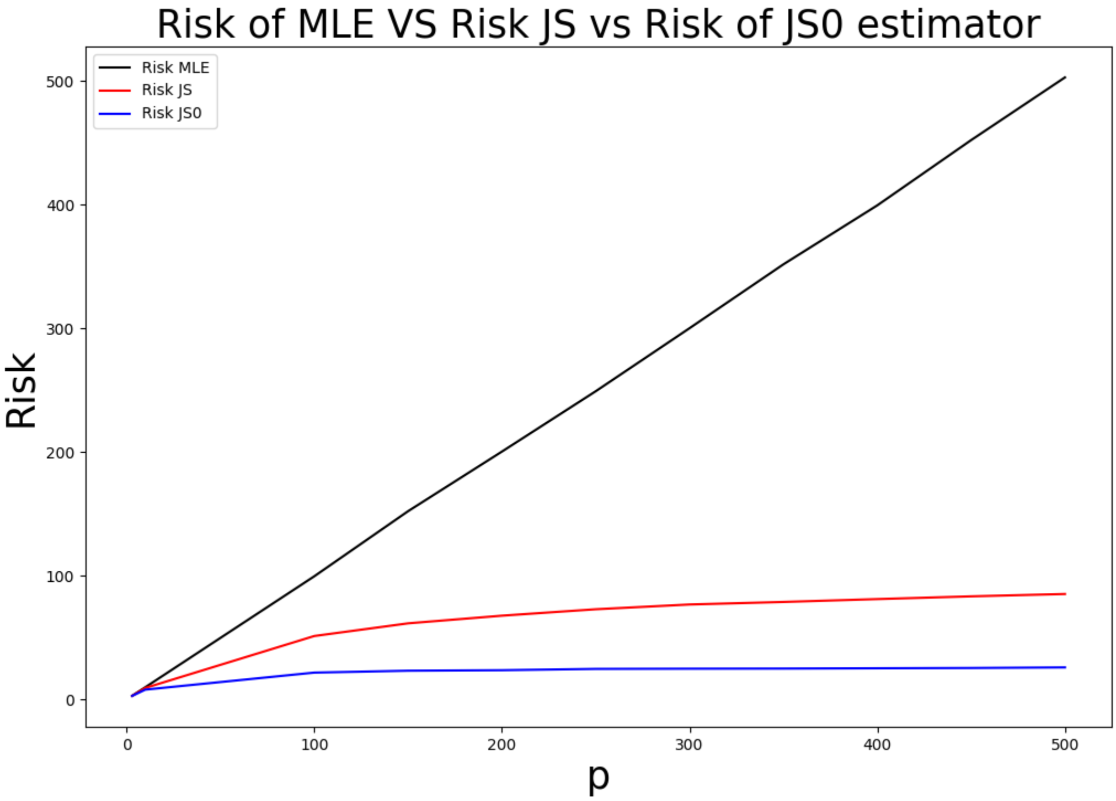


Figure 9: RISK of JSO: Code Available in RISK_JS0_and_MAX.py

From this figure we can indeed see how the risk of $\delta_{JSO}(\boldsymbol{X})$ is lower than the risk of the MLE, and the JS estimator, which is consistent with our reasoning.(since we chose a point x_0 closer to the true mean (I know this is "cheating" but just to illustrate the point)), so maybe we could start with the sample mean(as also discussed in your office!)

Risk of $\delta^+_{JS}(\boldsymbol{X})$

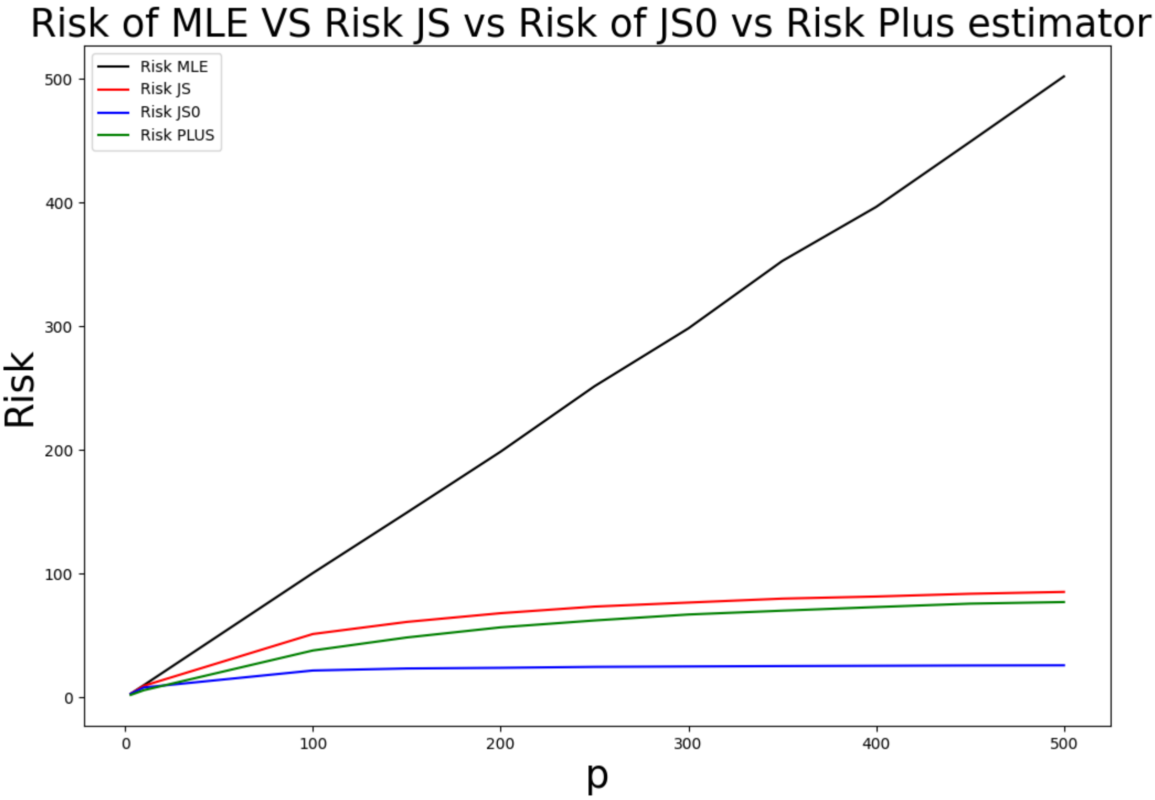


Figure 10: RISK of JS+: Code Available in RISK_JS0_and_MAX.py

We can clearly see aswell that the green curve (risk of $\delta^{+}_{JS}(\boldsymbol{X})$) is lower than the risk of the MLE, and the JS estimator.
These two "add-on" estimators clearly shows that even the JS is NOT admissible, which is just a small example of how strong this property is.

References

[1] Wikipedia contributors. (2023, October 10). Chi distribution. In Wikipedia, The Free Encyclopedia. Retrieved from `https://en.wikipedia.org/wiki/Chi_distribution`

[2] Wasserman, L. (2004). All of Statistics