دانشکده ریاضی و علوم کامپیوتر
دانشگاه صنعتی شریف

فیلتر کالمن (Kalman Filter)

گزارش مطالعه و پیاده‌سازی فیلتر کالمن

پروژه درس بهینه‌سازی در علوم داده

دکتر مجتبی تفاق، ترم دوم سال تحصیلی ۱۴۰۱

نگارندگان: محسن قدرتی، محمد صالح بهرامی

۱ مقدمه

در این گزارش قصد داریم الگوریتم فیلتر کالمن [Kal۶۰] را برای بهبود (فیلتر) برآوردهای یک بردار وضعیت، معرفی کنیم و با ارایه تعدادی مدل ساده از دینامیک تغییرات قیمت رمز ارز بیت‌کوین، قدرت فیلتر کالمن در پیش‌بینی آینده را محک بزنیم. در انتها نیز، تعمیم‌هایی از فیلتر کالمن به دینامیک‌های غیر خطی و مساله‌های با چندین مدل موازی را بیان خواهیم کرد. مشوق ما در مطالعه فیلتر کالمن، فهم بهتر تعدادی از مقالات بهینه‌سازی در این خصوص از جمله [YDB۲۲]، [BB۲۰] و [ea۲۰] بود.

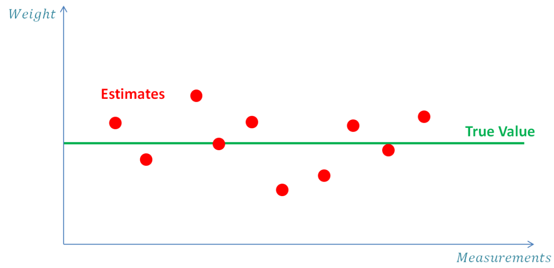
مطالعات ما در این گزارش، مربوط به نحوه تغییرات میانگین قیمت معاملاتی بیت‌کوین در بازه دو ماهه از دسامبر ۲۰۱۹ الی ژوییه ۲۰۲۰ است که از صرافی بایننس تهیه شده و مربوط به کندل‌های ۵ دقیقه‌ای است. به نظر می‌رسد نتایج نهایی کمابیش در سایر بازارها و یا سایر پارامترهای بازار و سایر بازه‌های زمانی نیز قابل پیاده‌سازی باشد.

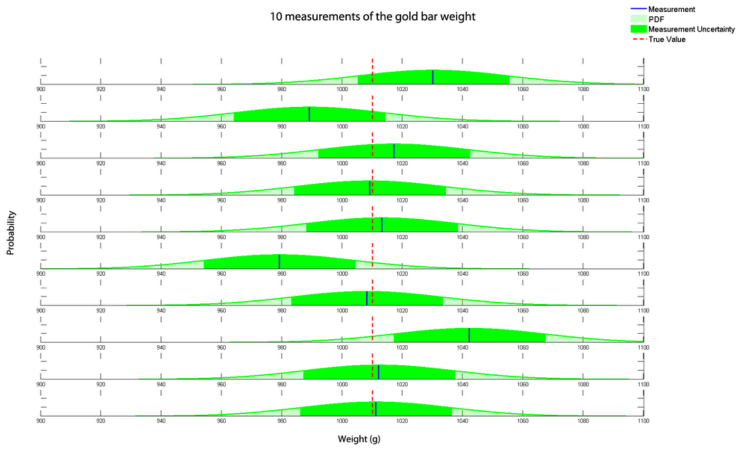
۲ تعاریف اولیه

۱.۲ مدل‌های اتورگرسیو

مدل‌های اتورگرسیو [Wik]، مدل‌هایی هستند که فرض می‌کنند مقدار یک سری زمانی به تعدادی ثابت از مقادیر قبلی به شکل خطی وابسته است و تفاضل این دو مقدار از یکدیگر یک نویز غیر قابل اندازه‌گیری خواهد بود. به عبارت دیگر، اگر x_t یک سری زمانی باشد، مدل زیر یک مدل اتورگرسیو است:

$$x_t = \sum_{i=1}^{m} x_{t-i} + \epsilon_t$$





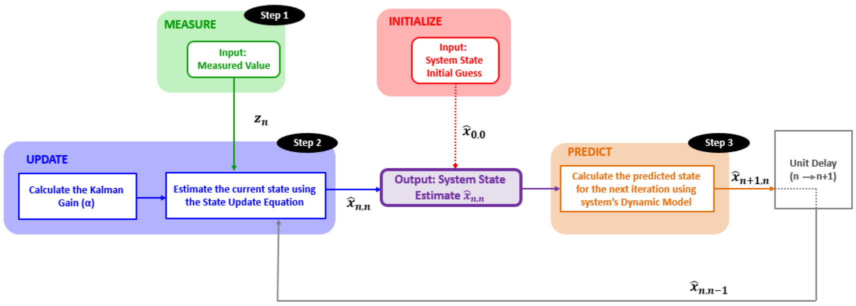
۲.۲ فیلتر

فرض کنید به دنبال اندازه‌گیری وزن یک شمش طلا هستیم و یک ترازو داریم که در اندازه‌گیری خود مقداری خطا دارد و هر بار یک وزن خاص بر می‌گرداند [Bec].

با فرض آنکه مانگین خطای ترازوی مورد نظر ما صفر است (نویز سفید داریم)، یک راه موثر برای تخمین وزن واقعی طلا، آن است که چندین بار شمش را وزن کنیم و میانگین وزن‌های یافته شده را برگردانیم. اما برای بروزرسانی میانگین مذکور، وقتی n بار آزمایش انجام شده است و مشاهداتی داشته‌ایم، نیازی به ذخیره‌سازی همگی نتایج نیست. کافی است به نکته زیر توجه کنیم:

$$\bar{x}_n = \frac{x_1 + \cdots + x_n}{n} = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} = \bar{x}_{n-1} + \frac{1}{n}(x_n - \bar{x}_{n-1})$$

یعنی کافیست در گام nام، میانگین را به اندازه ضریبی $(\frac{1}{n})$ در جهت تفاضل مشاهده جدیدمان از این میانگین موجود، تغییر دهیم. به این فرآیند، فیلتر کردن می‌گویند. در واقع با داشتن یک مشاهده و یک تقریب تا لحظه nام، فیلتر کردن یعنی آن که به نحوی این دو مقدار را با یکدیگر ترکیب کنیم که بهترین تقریب جدید ساخته شود. در شکل ؟؟ شمایی از روند تکراری فیلتر کردن یک سری زمانی را مشاده می‌کنیم:



	Open	High	Low	Close	Volume	Value	No. Trades	Taker Buy Volume	Taker Buy Value	Average Price	Average Price Change
2019-11-27 00:00:00	7154.75	7164.59	7140.54	7162.89	164.200264	1.174323e+06	1234	103.771014	742217.277131	7151.772051	NaN
2019-11-27 00:05:00	7164.4	7169.00	7152.92	7157.86	115.531698	8.275534e+05	904	73.404774	525826.737982	7162.998797	0.001570
2019-11-27 00:10:00	7157.32	7160.19	7144.93	7151.68	112.780186	8.065373e+05	914	57.210353	409185.537990	7151.409684	-0.001618
2019-11-27 00:15:00	7152.16	7152.20	7138.93	7142.66	157.903923	1.128197e+06	833	70.538540	503984.524916	7144.828939	-0.000920
2019-11-27 00:20:00	7142.63	7148.81	7138.94	7146.63	71.999481	5.143852e+05	638	42.162902	301241.148857	7144.290518	-0.000075
...	...	...	...	...	...	...	...	...	...	...	...
2020-02-03 23:35:00	9298.38	9300.35	9277.07	9294.66	148.543688	1.379950e+06	1348	82.834562	769549.767916	9289.856558	-0.000593
2020-02-03 23:40:00	9294.66	9307.69	9294.01	9304.29	90.220836	8.392079e+05	908	64.413847	599153.526012	9301.708836	0.001276
2020-02-03 23:45:00	9304	9319.99	9295.98	9316.82	139.875918	1.301847e+06	1874	68.115588	633991.892739	9307.156778	0.000586
2020-02-03 23:50:00	9316.15	9316.88	9303.73	9306.52	95.824739	8.921500e+05	1331	45.580965	424366.578549	9310.226182	0.000330
2020-02-03 23:55:00	9307.21	9307.73	9291.34	9292.24	127.278443	1.183576e+06	1141	69.671406	647856.534166	9299.110800	-0.001194

19872 rows × 11 columns

۳ معرفی مساله داده‌کاوی

۱.۳ صورت مساله و داده‌ها

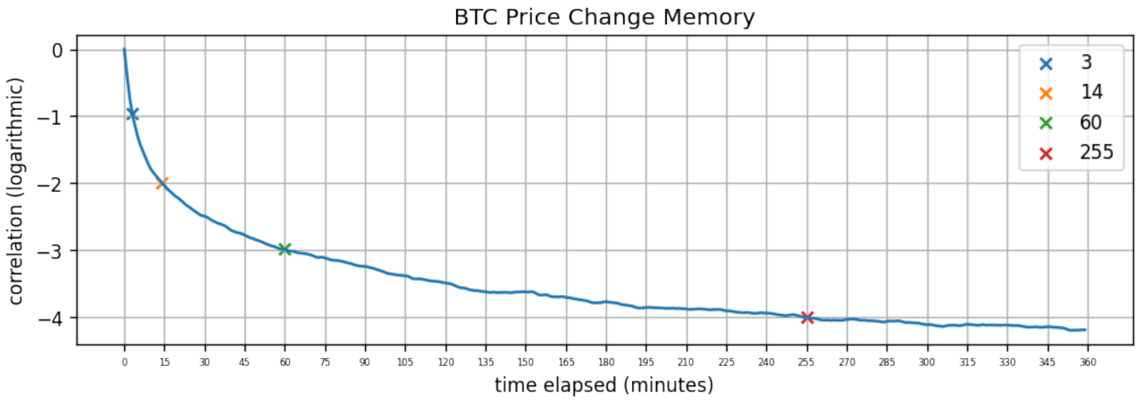
داده‌های این مساله، شامل اطلاعات کندلی ۵ دقیقه‌ای دو ماه از بازار بیت‌کوین در صرافی بایننس است [Bin]. این اطلاعات شامل میانگین قیمت معاملاتی بیت‌کوین در هر کندل نیز می‌شود. برای دسترسی به کلیه کدهای نوشته شده و تصاویر با وضوح اصلی، به [Gho] مراجعه کنید. قصد داریم با مدل‌های مختلف فیلتر کالمن، و اطلاعات قیمتی و حجمی تا به یک لحظه خاص، تغییرات قیمتی در ۵ دقیقه بعدی را پیش‌بینی کنیم.

سطرهای اولیه داده‌ها به صورت زیر است:

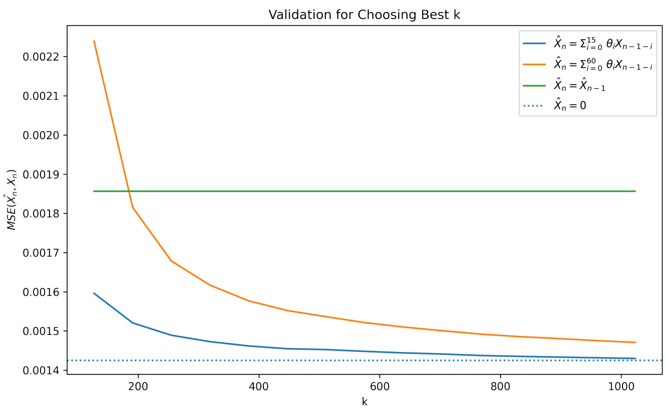
مشاهده ۱. همانطور که در شکل ؟؟ مشاهده می‌شود، قیمت بیت‌کوین در حوالی روزهای سال نو میلادی، در یک روند صعودی قرار داشته است [Tra].

۲.۳ تخمین حافظه موثر

حافظه یک مدل اتورگرسیو عبارت است از تعداد مشاهداتی از سری زمانی که برای مدل‌سازی دینامیک مشاهده جدید لازم دارد. در شکل ؟؟، ضریب همبستگی تغییرات قیمتی بیت‌کوین در یک دقیقه آینده با تغییرات قیمتی از t دقیقه قبل تا این لحظه رسم شده است. همانطور که از شکل ؟؟



correlation diagram (in binary logarithmic scale) of BTC price return with cummulative price return up to some future point



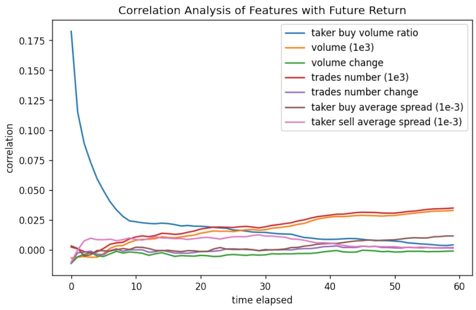
مشخص است، برای آنکه حداقل همبستگی $\frac{1}{4}$ داشته باشیم، لازم است حافظه را به $m = 14$ محدود کنیم. یعنی رابطه جدی بین زمان‌های دورتر از ۱۴ دقیقه قبل با لحظه فعلی وجود ندارد.

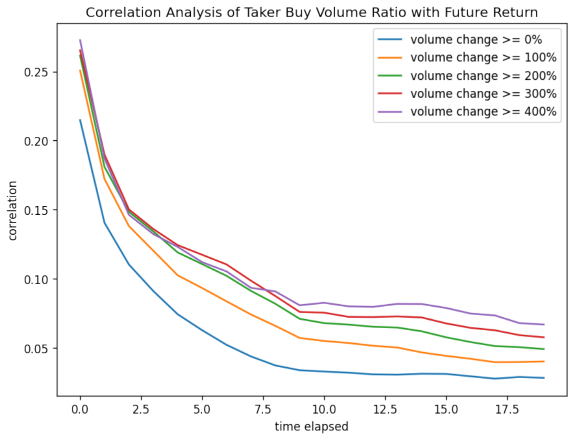
در ادامه، پیاده‌سازی‌های خود را به دو عدد $m = 15$ و $m = 60$ محدود می‌کنیم.

۳.۳ تخمین سایز مناسب برای یادگیری دینامیک مساله

دینامیک مساله در مدل اتورگرسیو، شامل تعدادی ضریب اسکالر و یک عرض از مبدا ثابت است. در واقع برای یافتن این ضرایب، نیاز به حل یک معادله رگسیون خطی داریم. اما اینکه تا چند نمونه برای یافتن ضرایبی با دقت بالا (pvalue ناچیز) لازم است، خود به validation اختیاج دارد. در شکل ؟؟، واریانس نویز مدل‌های خطی بدون کنترل، به ازای تعداد نمونه استفاده شده در یادگیری ضرایب دینامیک مدل رسم شده است. مدل آبی به ازای حافظه $m = 15$ و مدل نارنجی به ازای حافظه $m = 60$ تولید شده است. همانطور که از شکل پیداست، هنگامی که حافظه افزایش می‌یابد، واریانس نویز سیستم خطی نیز افزایش می‌یابد. یعنی حافظه خیلی بزرگ معادل است با عدم قطعیت بیشتر در پیش‌بینی تغییرات قیمت بیت‌کوین. هم‌چنین توجه می‌کنیم که هر دو مدل تقریبا به ازای همه انتخاب‌های k از مدلی که تغییرات قیمتی بیت‌کوین را برابر با تغییرات قیمتی در لحظه قبل می‌داند، بهتر هستند. اما از طرفی هیچ‌کدام از مدلی که تغییرات قیمتی را صفر پیش‌بینی می‌کند بهتر نیستند. دلایل اینکه از صفر پیش‌بینی کردن بهتر نیستیم، زیاد است و به تحلیل اقتصادی در کنار تحلیل داده‌ها نیاز دارد. اما تعدادی از عمده‌ترین دلایل چنین پیش‌بینی‌ناپذیریی عبارتند از اینکه اول از همه، دینامیک تغییرات قیمت در بازارهای کارایی مانند بیت‌کوین، به شدت غیرخطی و غیریکنواخت است. دوم اینکه در بسیاری از زمان‌ها، به دلیل عدم وجود اتفاق نظر در بین اهالی بازار، رفتار قیمت تصادفی یا بسیار سبه تصادفیست. پس مدلی که بخواهد در همه زمان‌ها تغییرات قیمت را پیش‌بینی کند، ناچار است بسیاری از نمونه‌هایش را از متغیرهایی بگیرد که کاملا نرمال و تصادفی متقارن هستند. در نتیجه بهتر از توزیعی که نمونه‌هایش داراست، نمی‌تواند پیش‌بینی انجام دهد. یعنی همان پیش‌بینی $\hat{X}_n = 0$.

در این گزارش، هدف پیش‌بینی موثر بازار بیت‌کوین نیست و صرفا به دنبال پیاده سازی فیلتر کالمن هستیم. لذا با وجود موارد بالا، به روند خود ادامه می‌دهیم.





۴.۳ تهیه تعدادی ویژگی (کنترل دینامیک)

همانطور که در ادامه خواهیم دید، افزودن ویژگی‌های مختلف بر حسب قیمت و حجم معاملات، کمک قابل توجهی به رگرسیون نمی‌کند. نقش این ویژگی‌ها بیشتر از آنکه در رگرسیون باشد، در جداسازی موقعیت‌های خاص بازار است. مثلا هنگامی که حجم معاملات به طور ناگهانی چند برابر می‌شود، یا تعداد معاملات افزایش چشمگیر می‌یابد. نقش اصلی ویژگی‌ها در تعیین خاص بودن یک موقعیت بازار است و نه در پیش‌بینی قیمت. اما این یک تجربه و نظر شخصی است و از ویژگی‌های زیر در کنار داده‌های سری زمانی استفاده خواهیم کرد.

۱. taker buy volume ratio: نسبت حجمی اجرا کنندگان معاملات بازار که موقعیت خرید داشته‌اند به آن‌هایی که موقعیت فروش داشته‌اند.

۲. volume (1e3): حجم در مقیاس کیلو

۳. volume change: تغییرات حجم به صورت درصدی (صد درصد همان یک فرض می‌شود)

۴. trades number (1e3): تعداد معاملات انجام شده

۵. trades number change: تغییرات تعداد معاملات انجام شده

۶. taker buy average spread (1e-3): فاصله میانگین قیمت خرید اجراکنندگان سفارش که بیت‌کوین خریده‌اند با میانگین کل

۷. taker sell average spread (1e-3): فاصله میانگین قیمت فروش اجراکنندگان سفارش که بیت‌کوین فروخته‌اند با میانگین کل

مشاهده ۲. همانطور که در شکل ؟؟ مشاهده می‌کنید، از میان ویژگی‌هایی که انتخاب کرده‌ایم، نسبت حجم بیت‌کوینی که خریداران سفارش را برداشته‌اند به حجم بیت‌کوینی که فروشندگان سفارش را برداشته‌اند، همبستگی قابل توجهی تا چند دقیقه با بازدهی قیمت دارد. هم‌چنین مشاهده می‌کنیم که همبستگی حجم و تغییرات حجم با بازدهی بلند مدت به مرور مثبت می‌شود.

مشاهده ۳. مشاهده بعدی مربوط به محدود کردن وضعیت‌های مورد مطالعه بازار به آن‌هایی است که تغییرات حجم مثبت قابل توجه داریم. در این راستا، شکل ؟؟ همبستگی ویژگی مورد توجه در نمودار بالا را با بازده آتی به ازای محدودسازی‌های مختلف تغییرات حجم نشان می‌دهد. شکل خود گویاست که با افزایش محدودیت تغییرات حجم، همبستگی ویژگی مورد نظر با بازدهی قیمت افزایش می‌یابد.

Prediction:

Predicted state estimate	$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{B}\boldsymbol{u}_{k-1}$
Predicted error covariance	$\boldsymbol{P}_k^- = \boldsymbol{F}\boldsymbol{P}_{k-1}^+\boldsymbol{F}^T + \boldsymbol{Q}$

Update:

Measurement residual	$\widetilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-$
Kalman gain	$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}^T\left(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^T\right)^{-1}$
Updated state estimate	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\widetilde{\boldsymbol{y}}$
Updated error covariance	$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^-$

۴ فیلتر کالمن

۱.۴ تعریف

فیلتر کالمن الگوریتمی است که با ترکیب تقریب‌های خام اولیه و مشاهده و اندازه‌گیری‌هایی از سیستم (که فرض می‌شود هم مشاهده ما از سیستم و هم تقریب‌های خام هر دو دارای خطا هستند)، یک تقریب بهبود یافته به دست می‌دهد. فرض کنید x_t متغیری باشد که نحوه تغییرات آن در طول زمان به طور "حدودی" به وسیله سیستم دینامیکی خطی

$$x_t = Fx_{t-1} + Gu_{t-1} + wt - 1$$

قابل توصیف باشد.که در آن w_{t-1} یک متغیر تصادفی است که نویز یا خطای مدل نام دارد. همچنین فرض کنید Q_{t-1} ماتریس کواریانس w_{t-1} باشد به گونه‌ای که $w_{t-1} \sim \mathcal{N}(0, Q_{t-1})$. از طرفی فرض کنید برای هر t به جای دسترسی مستقیم به x_t به یک نسخه نویزی از آن مثل z_t دسترسی داریم که :

$$z_t = x_t + v_t$$

و در اینجا v_t یک متغیر تصادفی با توزیع $v_t \sim \mathcal{N}(0, R_t)$ است. تا به این جا ما دو تقریب از مقدار x_t بدست آورده ایم که عبارتند از:

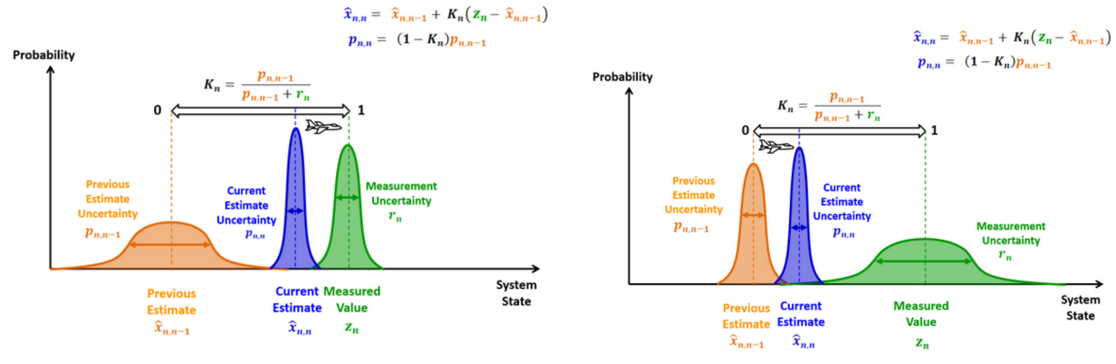
- تقریب با استفاده از مدل دینامیک سیستم $(Fx_{t-1} + Gu_{t-1})$
- تقریب با استفاده از مشاهده و اندازه گیری (z_t)

الگوریتم کالمن به طریق زیر یک تقریب بهبود یافته نسبت به هردوی تقریب‌های قبلی بدست می‌دهد.

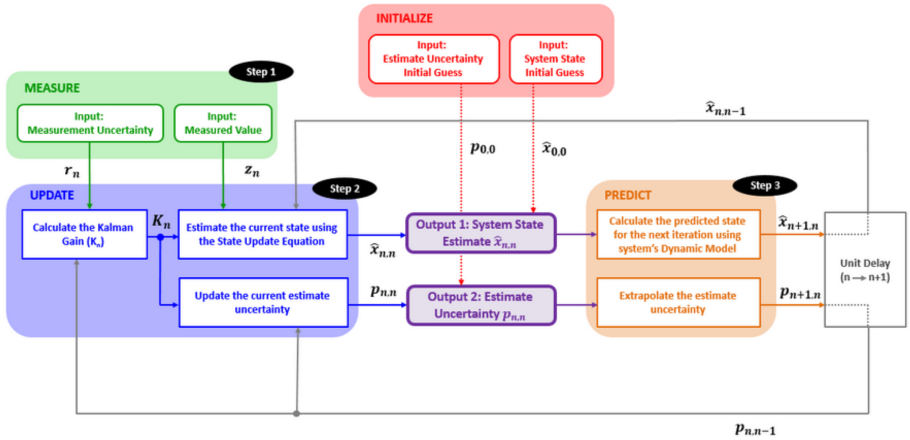
۲.۴ الگوریتم

در این تصویر [KB] علامت منفی روی تقریب‌ها به معنی این است که این تقریب اولیه است و علامت مثبت یعنی تقریب بعد از انجام الگوریتم کالمن بهبود یافته است. آنچه که در تصویر بالا در قسمت محاسبات مربوط به prediction مطرح شده است در واقع تخمین اولیه و میزان عدم قطعیتی است که ما از تخمین خود داریم. این تقریب از فرض ما بدست می آید که نحوه تغییرات x_t به وسیله سیستم دینامیک خطی اشاره شده توضیح داده می‌شود.

الگوریتم کالمن به طور شهودی، عدم قطعیت هر یک از دو برآورد (مشاهده و تخمین اولیه) خود را در نظر می‌گیرد و بر مبنای آن، یک ترکیب محدب از این دو برآورد ارایه می‌دهد.



(ب) تمایل به مقدار اندازه‌گیری شده (آ) تمایل به تخمین مدل



در این خصوص، توجه به شکل ؟؟ خالی از لطف نیست [Bec]. در این شکل دو موقعیت فرضی تصویر شده است که در یکی، عدم قطعیت مقدار اندازه‌گیری شده بالاست ولی عدم قطعیت تخمینی مدل کم است. لذا تخمین تصحیح شده توسط فیلتر کالمن به مقدار تخمینی مدل وزن بیشتری می‌دهد. در عکس دوم وضعیت برعکس است.

به طور خلاصه، فرآیندی که توسط الگوریتم کالمن به صورت تکراری و متناوب طی می‌شود، در شماتیک زیر آمده است:

۳.۴ عدم قطعیت

در این قسمت سعی داریم فرمول محاسبه عدم قطعیت را ثابت کنیم:

برای این کار، به اثبات گام به گام ؟؟ توجه می‌کنیم [Bec]:

$$P_{n,n} = (I - K_n H)\, P_{n,n-1} (I - K_n H)^T + K_n R_n K_n^T$$

where:

$P_{n,n}$	is the estimate uncertainty (covariance) matrix of the current state
$P_{n,n-1}$	is the prior estimate uncertainty (covariance) matrix of the current state (predicted at the previous state)
K_n	is the Kalman Gain
H	is the observation matrix
R_n	is the Measurement Uncertainty (measurement noise covariance matrix)
I	is an Identity Matrix (the $n \times n$ square matrix with ones on the main diagonal and zeros elsewhere)

	Notes
$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(z_n - H\hat{x}_{n,n-1})$	State Update Equation
$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(Hx_n + v_n - H\hat{x}_{n,n-1})$	Plug the Measurement Equation into the State Update Equation
$e_n = x_n - \hat{x}_{n,n}$	Estimate error
$e_n = x_n - \hat{x}_{n,n-1} - K_n\left(Hx_n + v_n - H\hat{x}_{n,n-1}\right)$	Plug in $\hat{x}_{n,n}$
$e_n = x_n - \hat{x}_{n,n-1} - K_nHx_n - K_nv_n + K_nH\hat{x}_{n,n-1}$	Expand
$e_n = x_n - \hat{x}_{n,n-1} - K_nH\left(x_n - \hat{x}_{n,n-1}\right) - K_nv_n$	Localize $(x_n - \hat{x}_{n,n-1})$
$e_n = (I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right) - K_nv_n$	
$P_{n,n} = E\left(e_ne_n^T\right) = E\left(\left(x_n - \hat{x}_{n,n}\right)\left(x_n - \hat{x}_{n,n}\right)^T\right)$	Estimate Uncertainty
$P_{n,n} = E\left(\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right) - K_nv_n\right) \times \times\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right) - K_nv_n\right)^T\right)$	Plug in e_n

(آ)

$P_{n,n} = E\Big((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T - -(I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)(K_nv_n)^T - -K_nv_n\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T + +K_nv_n(K_nv_n)^T\Big)$	Expand
$P_{n,n} = E\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T\right) - -E\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)(K_nv_n)^T\right) - -E\left(K_nv_n\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T\right) + +E\left(K_nv_n(K_nv_n)^T\right)$	Apply the rule $E(X \pm Y) = E(X) \pm E(Y)$
$E\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)(K_nv_n)^T\right) = 0$ $E\left(K_nv_n\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T\right) = 0$	$(x_n - \hat{x}_{n,n-1})$ is the error of the prior estimate in relation to the true value. It is uncorrelated with the current measurement noise v_n. The expectation value of the product of two independent variables is zero.
$P_{n,n} = E\left((I - K_nH)\left(x_n - \hat{x}_{n,n-1}\right)\left(x_n - \hat{x}_{n,n-1}\right)^T(I - K_nH)^T\right) + +E\left(K_nv_nv_n^TK_n^T\right)$	Apply the matrix transpose property: $(AB)^T = B^TA^T$
$P_{n,n} = (I - K_nH)\,E\left(\left(x_n - \hat{x}_{n,n-1}\right)\left(x_n - \hat{x}_{n,n-1}\right)^T\right)(I - K_nH)^T +$	Apply the rule $E(aX) = aE(X)$

(ب)

۴.۴ پیاده‌سازی

۱.۴.۴ مدل ۱: دینامیک خطی بدون کنترل

فرض می‌کنیم دینامیک تغییرات قیمت به شکل زیر باشد:

$$\tilde{r}_n = f_{0,n}\tilde{r}_{n-1} + \sum_{i=1}^{m-1} f_{i,n} r_{i,n-1} + f_{m,n} + w_n, \qquad w_n \sim \mathcal{N}(0,\ \sigma_n^2). \tag{1}$$

به عبارت دیگر، فرض می‌کنیم اگر تغییرات قیمت از $i = 2, \ldots, m-1$ کندل گذشته تا کندل گذشته را بدانیم و تغییرات قیمت از کندل گذشته تا کندل فعلی را نیز بدانیم، بازدهی پیش رو ترکیبی خطی از این مقادیر به علاوه یک نویز با توزیع نرمال است. در این صورت برای بردار $\mathbf{x^{(n)}}$ فرم معادل زیر را خواهیم داشت:

$$\mathbf{x^{(n)}} = F_n \mathbf{x^{(n-1)}} + \mathbf{w^{(n)}}, \tag{2}$$

که در آن

$$\mathbf{x^{(n)}} := \begin{bmatrix} \tilde{r}_n \\ r_{1,n} \\ \vdots \\ r_{m-1,n} \\ 1 \end{bmatrix}, \quad F_n := \begin{bmatrix} f_{0,n} & f_{1,n} & \cdots & f_{m,n} \\ 1+\tilde{r}_{n-1} & 0 & \ddots & 0 \\ 0 & 1+\tilde{r}_{n-1} & 0 & \vdots \\ 0 & \ddots & & 0 \\ \vdots & 0 & 1+\tilde{r}_{n-1} & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}, \quad \mathbf{w^{(n)}} := \begin{bmatrix} w_n \\ -\tilde{r}_{n-1}^2 \\ \tilde{r}_{n-1} \\ \vdots \\ \tilde{r}_{n-1} \\ 0 \end{bmatrix}.$$

با توجه به این که در هر لحظه n، بردار $\mathbf{x^{(n-1)}}$ معلوم است و کواریانس بردار $\mathbf{w^{(n)}}$ تنها در درایه ۱٫۱ ناصفر و برابر با σ_n^2 است، به دنبال مقادیری از $f_{0,n}, \ldots, f_{m-1,n}$ می‌گردیم که σ_n^2 را کمینه کند. (در واقع خطای مدل را در حد امکان کاهش دهد.) هم‌چنین توجه می‌کنیم که اضافه کردن عدد ۱ به انتهای بردار $\mathbf{x^{(n)}}$ باعث می‌شود بتوان عرض از مبدا مدل ۱ را به $f_{m,n}$ منتقل کرد و لذا فرض صفر بودن میانگین نویز w_n برقرار است. اکنون سعی می‌کنیم با بازنویسی معادله بالا، فرم مجموع مربعاتی برای یافتن بهترین $f_{i,n}$ها ارایه کنیم.

اگر قرار دهیم $\theta_n = (f_{0,n}, \ldots, f_{m,n})^T$، به دنبال حل مساله بهینه‌سازی زیر هستیم:

$$minimize_{(\theta_n)} \quad |\tilde{r}_n - \langle \theta_n,\ \mathbf{x^{(n-1)}} \rangle|^2$$

اما با توجه به اینکه مساله بالا احتمالا بی‌نهایت جواب دارد و این موضوع باعث تغییرات سریع دینامیک مدل در هر لحظه می‌شود که خلاف طبیعت مدل است، و مهم‌تر آنکه انتظار داریم دینامیک مدل در طول یک پنجره کوتاه مدت (kتایی) پایدار باشد (یعنی $\theta_n \simeq \theta_{n-1} \simeq \cdots \simeq \theta_{n-(k-1)}$) و بتوانیم از اطلاعات این پنجره برای تقریب یک دینامیک کم‌نوسان بهره ببریم، مساله بهینه‌سازی فوق را به مساله زیر تعمیم می‌دهیم:

$$minimize_{(\theta_n)} \quad |\tilde{r}_{n-(k-1)} - \langle \theta_n,\ \mathbf{x^{(n-1-(k-1))}} \rangle|^2 + \cdots + |\tilde{r}_n - \langle \theta_n,\ \mathbf{x^{(n-1)}} \rangle|^2$$

که با در نظر گرفتن ماتریس $X^{(n)}$ به صورت

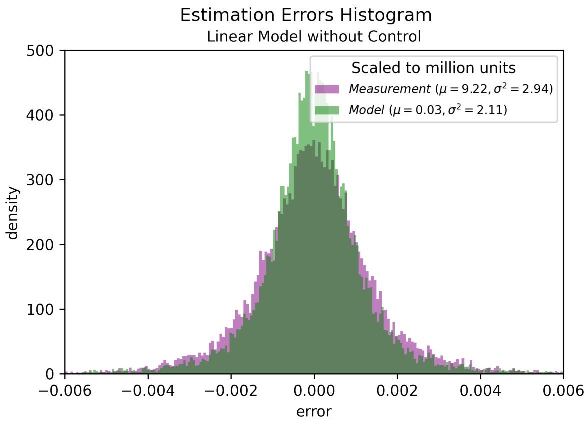
$$X^{(n)} = \begin{bmatrix} \mathbf{x^{(n-(k-1))}}^T \\ \vdots \\ \mathbf{x^{(n)}}^T \end{bmatrix} \quad \rightarrow \quad X^{(n)} := \begin{bmatrix} \mathbf{b_0^{(n)}} & \ldots & \mathbf{b_m^{(n)}} \end{bmatrix}$$

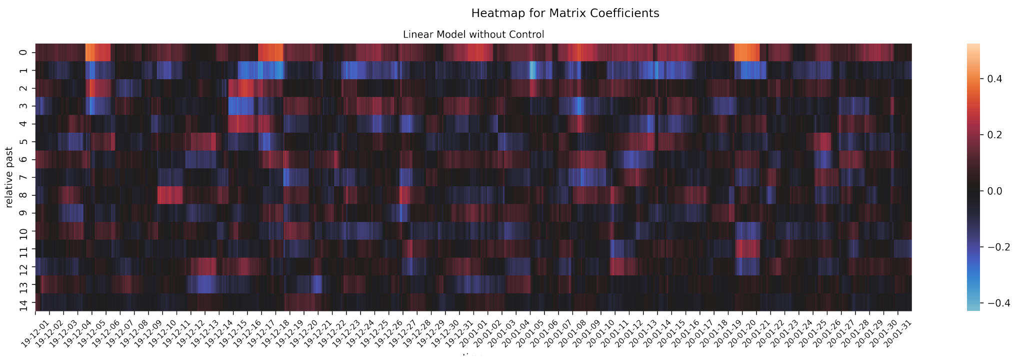
به فرم زیر قابل بیان است:

$$for\ given\ n: \quad minimize_{(\theta_n)}\ ||X^{(n-1)}\theta_n - \mathbf{b_0^{(n)}}||_2^2 \tag{3}$$

توجه به این نکته ضروری است که از بردار $X_0^{(n)}$، درایه اول مربوط به بازدهی آینده است و در درسترس نیست. لذا به جای حل مساله فوق برای n آن را برای $n-1$ حل می‌کنیم.

در ادامه به بررسی نتایج حاصل از حل مساله بهینه‌سازی ۳ به کمک روش کمترین مجموع مربعات و پیاده‌سازی فیلتر کالمن بر روی مدل ۱ می‌پردازیم.

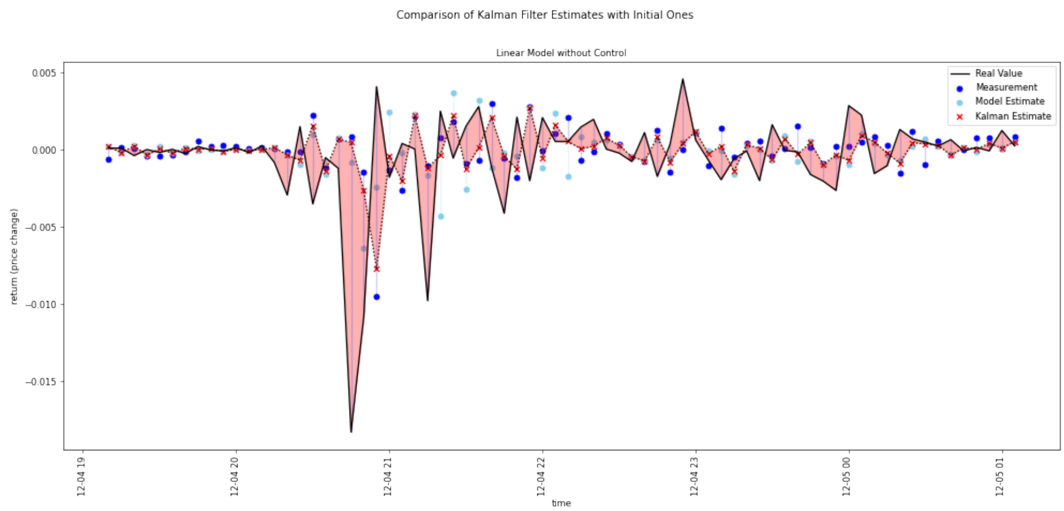




مشاهده ۴. مطابق با نمودار زیر می‌توان فرض نرمال بودن نویز در اندازه‌گیری و مدل را تا حدودی تایید کرد. همچنین توجه می‌کنیم که میانگین توزیع نویز، با دقت ۷ رقم اعشار صفر خواهد بود.

مشاهده ۵. همان‌طور که اشاره کردیم، فرض می‌کنیم دینامیک مدل دچار تغییرات سریع نمی‌شود. این نکته در نمودار زیر به صورت طیف‌های پیوسته قابل مشاهده است. همچنین در نمودار زیر این نکته قابل توجه وجود دارد که ضرایب بزرگ‌تر و تاثیرگذارتر در مدل خطی پیشنهاد شده، به لحظات نزدیک‌تر به لحظه حال مرتبط می‌شوند. این نیز مشاهده‌ای است که انتظارش را داشتیم.

نتیجه ۶. در شکل می‌توان خروجی فیلتر کالمن را بر روی مدل خطی در هر لحظه به ازای یک پنجره خاص زمانی مشاهده کرد.



۲.۴.۴ مدل ۲: دینامیک خطی همراه با کنترل

در مدل زیر، فرض می‌کنیم تعدادی ویژگی (که در بخش قبل تهیه شده‌اند و در همان بخش، همبستگیشان با تابع هدف ما بررسی و تایید شده است) نیز علاوه بر بردار $\mathbf{x}^{(\mathbf{n-1})}$، در دینامیک تغییرات قیمت به شکل خطی موثر باشند. به عبارت دیگر دینامیک مساله را با مدل زیر تقریب می‌زنیم:

$$\tilde{r}_n = f_{0,n}\tilde{r}_{n-1} + \sum_{i=1}^{m-1} f_{i,n}r_{i,n-1} + f_{m,n} + \sum_{i=0}^{s-1} g_{i,n}u_{i,n} + w_n, \qquad w_n \sim \mathcal{N}(0,\ \sigma_n^2). \tag{4}$$

که با فرم زیر معادل است:

$$\mathbf{x}^{(\mathbf{n})} = F_n\mathbf{x}^{(\mathbf{n-1})} + G_n\mathbf{u}^{(\mathbf{n})} + \mathbf{w}^{(\mathbf{n})}, \tag{5}$$

و در آن $\mathbf{u}^{(\mathbf{n})}$ بردار ویژگی‌ها در لحظه n است. در واقع:

$$G_n := \begin{bmatrix} g_{0,n} & \cdots & g_{s-1,n} \\ 0 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{u}^{(\mathbf{n})} := \begin{bmatrix} u_{0,n} \\ \vdots \\ u_{s-1,n} \end{bmatrix}.$$

با در نظر گرفتن ماتریس $X^{(n)}$ و بردار $\mathbf{b}_0^{(\mathbf{n})}$ مدل قبلی، و تعریف ماتریس U_n و بردار θ'_n به شکل زیر،

$$U_n := \begin{bmatrix} \mathbf{u}^{(\mathbf{n-(k-1)})^T} \\ \vdots \\ \mathbf{u}^{(\mathbf{n})^T} \end{bmatrix}, \qquad \theta'_n := \begin{bmatrix} g_{0,n} \\ \vdots \\ g_{s-1,n} \end{bmatrix}$$

می‌توان به سادگی و مشابه با روند مدل ۱، به مساله کمترین مجموع مربعات زیر دست یافت:

$$for\ given\ n: \quad minimize_{(\theta_n,\theta'_n)}\ ||X^{(n-1)}\theta_n + U_n\theta'_n - \mathbf{b}_0^{(\mathbf{n})}||_2^2 \tag{6}$$

اکنون با ادغام دو ماتریس $X^{(n-1)}$ و U_n و ادغام دو بردار θ_n و θ'_n مساله بهینه‌سازی معادل زیر را داریم:

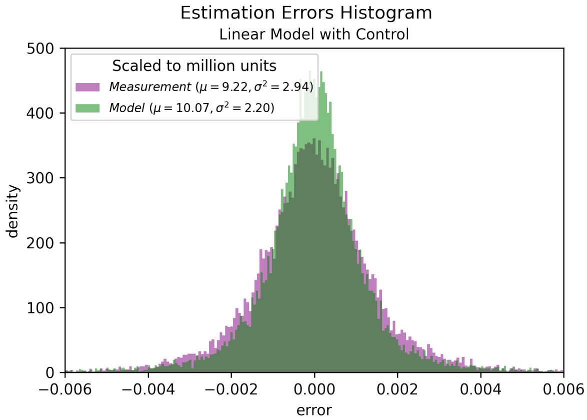
$$A_n := \begin{bmatrix} X^{(n-1)} & U_n \end{bmatrix}, \quad \theta_n^* := \begin{bmatrix} \theta_n \\ \theta'_n \end{bmatrix} \qquad \longrightarrow \qquad for\ given\ n: \quad minimize_{(\theta_n^*)}\ ||A_n\theta_n^* - \mathbf{b}_0^{(\mathbf{n})}||_2^2$$

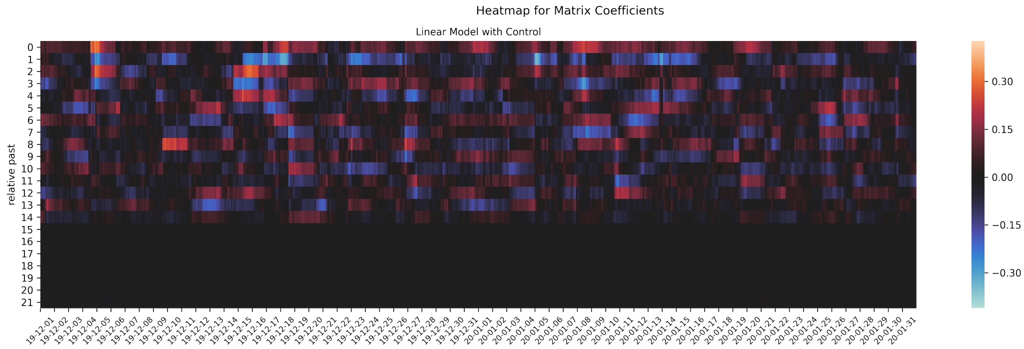
در ادامه به بررسی نتایج حاصل از حل مساله بهینه‌سازی ۶ به کمک روش کمترین مجموع مربعات و پیاده‌سازی فیلتر کالمن بر روی مدل ۴ می‌پردازیم.

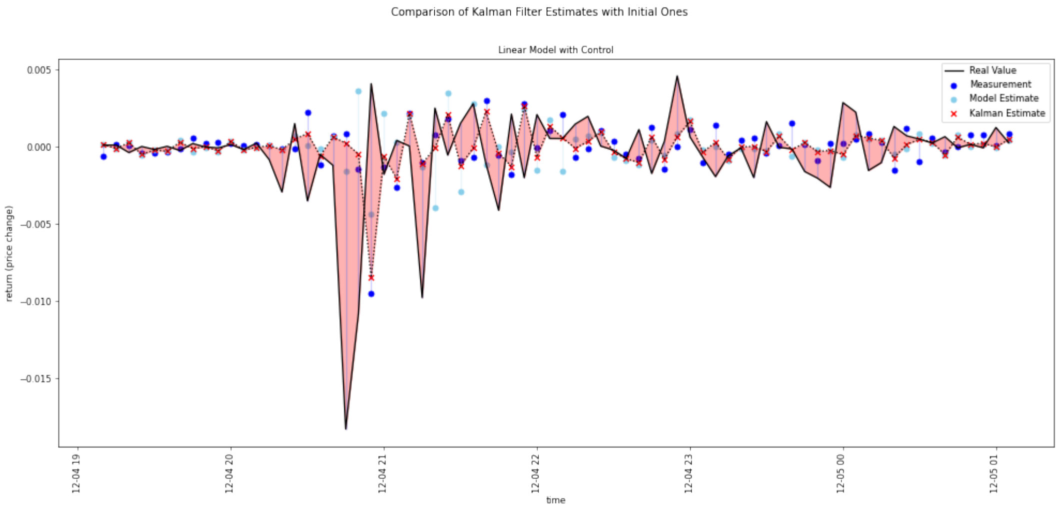
مشاهده ۷. مطابق با نمودار زیر می‌توان فرض نرمال بودن نویز در اندازه‌گیری و مدل را مجددا تا حدودی تایید کرد. هرچند همانطور که از نمودار پیداست، واریانس مدل کمی افزایش (جزیی) داشته و کم نشده است! در بخش‌های بعدی، زمانی که همه مدل‌های خطی را معرفی کردیم و به مقایسه عملکرد آن‌ها پرداختیم، دلیل عملیاتی این موضوع را توضیح خواهیم داد.

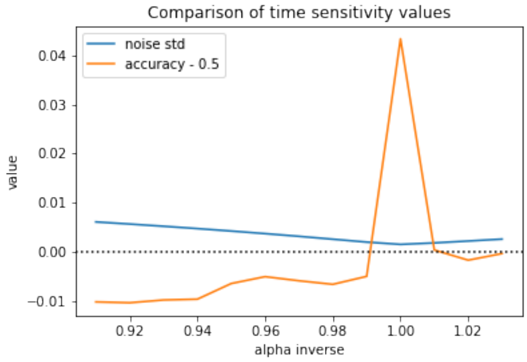
مشاهده ۸. نمودار زیر مشابه با نمودار مربوط به مدل ۱ قابل قبول است. نکته حایز اهمیت در این نمودار، تباهیده شدن ضرایب مربوط به ویژگی‌هاست که در واقع به دلیل انحراف معیار بالاتر ستون‌های مربوط به ویژگی‌ها نسبت به ستون‌های مربوط به بازدهی است و اطلاعاتی از دست نرفته است.

نتیجه ۹. در شکل ؟؟ می‌توان خروجی فیلتر کالمن را بر روی مدل خطی در هر لحظه به ازای یک پنجره خاص زمانی مشاهده کرد.


Estimation Errors Histogram
Linear Model with Control
Scaled to million units
Measurement ($\mu = 9.22, \sigma^2 = 2.94$)
Model ($\mu = 10.07, \sigma^2 = 2.20$)
density
error


Heatmap for Matrix Coefficients
Linear Model with Control
relative past


Comparison of Kalman Filter Estimates with Initial Ones
Linear Model with Control
Real Value
Measurement
Model Estimate
Kalman Estimate
return (price change)
time



۳.۴.۴ مدل ۳: دینامیک خطی حساس به زمان

منظور از مدل حساس به زمان، مدلی است که به نوعی به اطلاعات اخیرتر بیشتر توجه دارد (مثلا توسط یک ماتریس وزن‌دهی). همانطور که می‌توان حدس زد، چنین مدلی اساسا واریانس نویز بیشتری دارد. اما در عوض ممکن است به کمک فیلتر کالمن بتوان نویز آن را بهتر کنترل کرد، چرا که چنین مدلی در صورت خطی بودن واقعیت دینامیک مساله، سریع‌تر بروزرسانی می‌شود.

به دو روش می‌توان حساسیت به زمان را در مدل منعکس کرد. یکی حساسیت در یادگیری پارامترها، و دیگری حساسیت در توجه به اطلاعات سری زمانی.

روش اول معادل است با مساله بهینه‌سازی زیر:

$$for\ given\ n: \quad minimize_{(\theta_n^*)}\ ||D_1 A_n \theta_n^* - D_1 \mathbf{b}_0^{(\mathbf{n})}||_2^2, \quad D_1 = \frac{1}{\lambda} diag(\alpha^0,\ \dots,\ \alpha^{k-1}),\ \lambda = \sum_{i=0}^{k-1} \alpha^i \qquad (۷)$$

که در آن α پارامتر تاثیرگذاری است و مقداری حداقل برابر با ۱ دارد. پارامتر α در واقع بیان می‌کند کوچک کردن هر مقدار باقیمانده w_j در مساله مجموع کمترین مربعات، چه اندازه از کوچک کردن مقدار باقیمانده w_{j-1} مهم‌تر است.

روش دوم معادل است با وزن‌دهی به درایه‌های بردار θ_n به گونه‌ای که هرچه اندیس i در $f_{i,n}$ افزایش می‌یابد، $|f_{i,n}|$ کاهش یابد. پس باید وزن‌های بزرگ‌تری به $f_{i,n}$های با اندیس بزرگ‌تر در مساله بهینه‌سازی ۶ بدهیم. یعنی مساله بهینه‌سازی زیر:

$$for\ given\ n: \quad minimize_{(\theta_n^*)}\ |\left[X^{(n-1)} D_2 \quad U_n\right] \theta_n^* - \mathbf{b}_0^{(\mathbf{n})}||_2^2, \quad D_2 = \frac{1}{\lambda} diag(\alpha^0,\ \dots,\ \alpha^m),\ \lambda = \sum_{i=0}^{m} \alpha^i \qquad (۸)$$

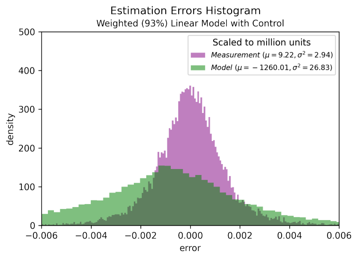
در ادامه تنها نتیجه حل مساله بهینه‌سازی ۷ آمده است:

مشاهده ۱۰. ابتدا به ازای مقادیر مختلف α با بررسی دقت پیش‌بینی (صرفا درستی علامت $\tilde{r}_n$) و انحراف معیار نویز مدل، بهترین α را انتخاب می‌کنیم:

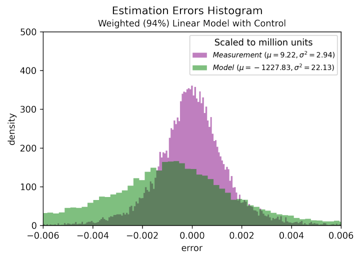
بر اساس شکل در می‌یابیم نقاطی که در پیش‌بینی جهت $\tilde{r}_n$ بیشترین دقت را دارند (از تشخیص تصادفی فاصله دارند و در عین حال انحراف معیار نویزشان کم است) عبارت‌اند از ۰٫۹۴ و ۱. از طرفی کمترین انحراف معیار نویز، همانطور که پیش بینی شده بود، در حوالی ۱ رخ می‌دهد. پس اگر به دنبال مقداری غیر بدیهی (مخالف ۱) هستیم، $\alpha = ۰٫۹۴$ گزینه مناسبی است. (توجه می‌کنیم که نیمه عمر چنین انتخابی حدود ۱۲ است، یعنی حساسیت مدل به داده هر لحظه حدود دو برابر حساسیت آن به داده ۱۲ کندل قبل است!)

هم‌چنین در ادامه تصویری از هیستوگرام نویز تعدادی از انتخاب‌های α قابل مشاهده است.

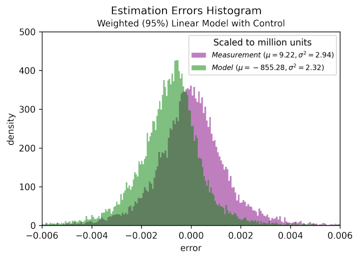
نتیجه ۱۱. در شکل ؟؟ می‌توان خروجی فیلتر کالمن را بر روی مدل خطی حساس به زمان (با $\alpha^{-1} = ۰٫۹۴$) در هر لحظه به ازای یک پنجره خاص زمانی مشاهده کرد.



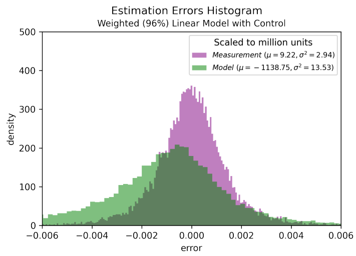
$\alpha^{-1} =$ ۰/۹۳ (آ)



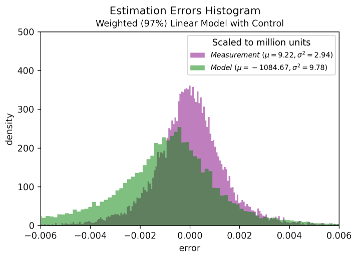
$\alpha^{-1} =$ ۰/۹۴ (ب)



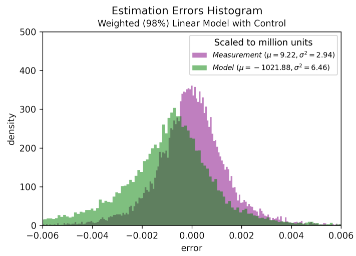
$\alpha^{-1} =$ ۰/۹۵ (ج)



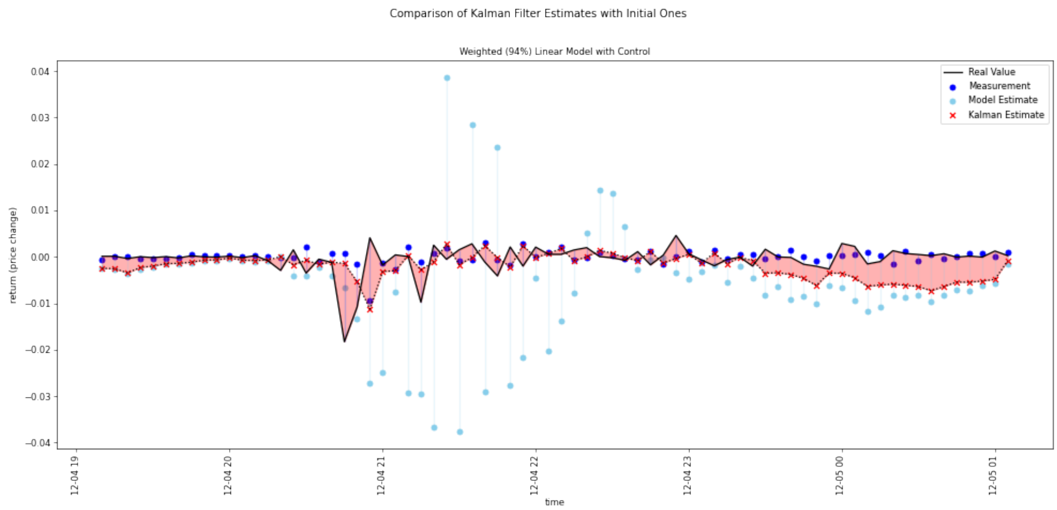
$\alpha^{-1} =$ ۰/۹۶ (د)



$\alpha^{-1} =$ ۰/۹۷ (ه)



$\alpha^{-1} =$ ۰/۹۸ (و)



۴.۴.۴ مدل ۴: دینامیک خطی با ماتریس چگال

در مدل‌های قبل، برای سهولت در محاسبه و سادگی مدل، فرض کردیم ماتریس‌های F_n و G_n تنک هستند. در این مدل، بدون تغییر فرض خطی بودن دینامیک مساله، فرض تنک بودن F_n و G_n را حذف می‌کنیم. در این صورت هرچند واریانس نویز تقریب آخرین بازدهی افزایش می‌یابد، اما با توجه به اینکه حالت تنک معادله خطی زیر، یک حالت خاص از معادله کلی است، انتظار داریم مساله بهینه‌سازی بتواند جواب‌های پایدارتری تولید کند. فایده دستگاه‌های خطی با واریانس نویز w_n بالاتر ولی پایداری بیشتر ان است که در بازارهای مالی به طور طبیعی مقدار بسیار زیادی نویز غیر قابل کنترل وجود دارد، و با پایدارتر کردن مدل، به ذات دینامیک مساله بهتر دسترسی داریم. (با فرض شبیه به خطی بودن این دینامیک

$$\mathbf{x}^{(\mathbf{n})} = F_n\mathbf{x}^{(\mathbf{n-1})} + G_n\mathbf{u}^{(\mathbf{n})} + \mathbf{w}^{(\mathbf{n})}$$

همچنین در مدل بالا، فرض کنیم یک بردار آفست $\mathbf{h}^{(\mathbf{n})}$ نیز در هر لحظه به سمت راست معادله اضافه می‌شود و در عوض $\mathbf{w}^{(\mathbf{n})}$ توزیع نرمال با میانگین ۰ دارد. در این صورت مساله بهینه‌سازی یافتن بهترین پارامترها برای تقریب خطی دینامیک مساله، در فرم کلی خود به صورت زیر است:

$$for\ given\ n: \qquad minimize_{(F_n,\, G_n,\, \mathbf{h}^{(\mathbf{n})})} \quad \left|\left| \begin{bmatrix} F_n & G_n & diag(\mathbf{h}^{(\mathbf{n})}) \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(\mathbf{n-1})} \\ \mathbf{u}^{(\mathbf{n})} \\ \mathbf{1}_{\mathbf{m\times 1}} \end{bmatrix} - \mathbf{x}^{(\mathbf{n})} \right|\right|_2^2 \qquad (9)$$

Prediction:

Predicted state estimate	$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}\right)$
Predicted error covariance	$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}$

Update:

Measurement residual	$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-\right)$
Kalman gain	$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T\left(\boldsymbol{R} + \boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T\right)^{-1}$
Updated state estimate	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\tilde{\boldsymbol{y}}$
Updated error covariance	$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-$

که با ترانهاده گرفتن از دو عبارت ماتریسی، و تکرار آنچه در مورد مدل ۱ بیان شد، به فرم زیر رسید:

$$for\ given\ n: \qquad minimize_{(F_n,\, G_n,\, \mathbf{h^{(n)}})} \quad \left|\left| \begin{bmatrix} X^{(n-1)} & U_n & \mathbf{1_{k\times m}} \end{bmatrix} \begin{bmatrix} F_n^T \\ G_n^T \\ diag(\mathbf{h^{(n)}}) \end{bmatrix} - X^{(n)} \right|\right|_2^2 \tag{۱۰}$$

مساله بهینه‌سازی فوق که تعمیم کمترین مجموع مربعات عادی به ماتریس مجهول است، در تمرین ۱۳.۱۵ کتاب [BV] بررسی شده است. در واقع هر ستون ماتریس مجهول به شکل جداگانه با روش کمترین مجموع مربعات یافته می‌شود و سپس کل ماتریس را از کنار هم قرار دادن این ستون‌ها معرفی می‌کنیم [Num]. توجه می‌کنیم که پس از یافتن ماتریس مجهول، در واقع ستون اول به تنهایی برای تشکیل دینامیک خطی همراه با کنترل لازم است. اما کل ماتریس را برای محاسبه ماتریس‌های کواریانس در هر مرحله از الگوریتم کالمن ذخیره می‌کنیم.

۵ فیلتر کالمن تعمیم‌یافته

در صورتی که دینامیک مساله غیرخطی باشد، فیلتر کالمن دیگر تضمینی برای بهترین جواب بودن ندارد. اما هم‌چنان می‌توان با خطی‌سازی از تقریب کالمن استفاده کرد [KB]. فرض کنیم دینامیک مساله به شکل زیر تقریب زده شود:

$$\begin{cases} \mathbf{x_k} = f(\mathbf{x_{k-1}}, \mathbf{u_k}) + \mathbf{w_k} \\ \mathbf{z_k} = h(\mathbf{x_{k-1}}) + \mathbf{v_k} \end{cases} \tag{۱۱}$$

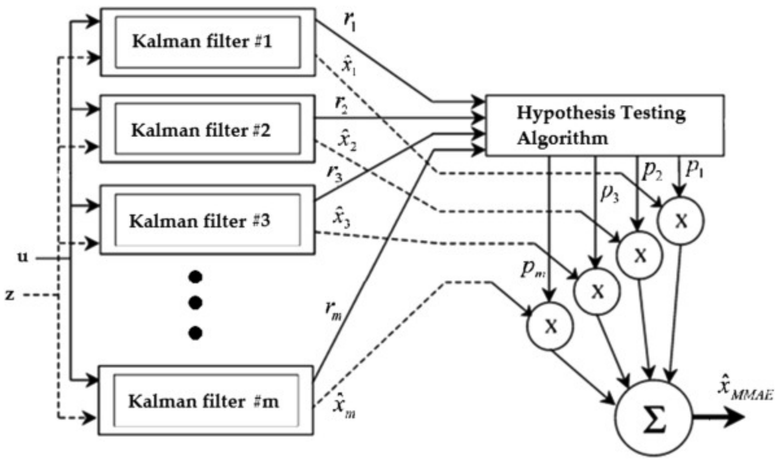
در این صورت می‌توان در فرمول‌های فیلتر کالمن، به جای F و H، تقریب خطی توابع f و h را گذاشت. در نتیجه الگوریتم شکل ؟؟ را خواهیم داشت [KB]:
که در آن

$$\begin{cases} F_{k-1} = \dfrac{\partial f}{\partial x}|_{\hat{\mathbf{x}}_{\mathbf{k-1}}^+, \mathbf{u_{k-1}}} \\ H_k = \dfrac{\partial h}{\partial x}|_{\hat{\mathbf{x}}_{\mathbf{k}}^-} \end{cases} \tag{۱۲}$$

۶ فیلتر کالمن با چند مدل

یک چالش پیش رو این است که احتمالا هیچ یک از برآوردگرهای قبلی یک غلبه کامل در همه حالات نسبت به سایرین ندارد و هر کدام تحت شرایط خاصی بهتر از بقیه عمل می‌کند. در صورتی که در الگوریتم کالمن فقط یک برآوردگر داریم. حال ما می‌خواهیم الگوریتم کالمن را به نحوی توسعه

دهیم که در حالت وجود چندین برآوردگر اولیه نیز کارآمد باشد. اساس این روش به این صورت خواهد بود که میانگین وزن‌داری از برآوردگرها و دستگاه‌های اندازه‌گیری (برآوردگر یا ابزار اندازه‌گیری iام را با $\hat{x}_i$ نمایش می‌دهیم) را به عنوان برآورد نهایی خود قرار دهیم و نحوه وزن‌دهی به برآوردگرها نسبت معکوس با عدم قطعیت موجود (واریانس) در آن برآوردگر داشته باشد. شکل زیر برگرفته از مدل ارایه شده در [۰۰Han] است.



به طرق مختلف میتوان این کار را انجام داد [۱۹ea]. به عنوان مثال ابتدا هر مدل را با مشاهدات خود فیلتر کنیم و بعد میانگین وزن داری از آنها را بگیریم. یا ابتدا میانگین وزن‌داری از برآوردگرها را محاسبه کنیم و مقدار آن را به عنوان تقریب خام اولیه به الگوریتم کالمن بدهیم. یک راه دیگر اینست که برآوردگر iام را با برآوردگر $i+1$ام به طور سری تلفیق کنیم (در این جا برآوردگر $i+1$ام نقش دستگاه اندازه گیری را بازی می‌کند) و نتیجه را با برآوردگر $i+2$ام تلفیق کنیم و همین‌طور الی آخر.

۷ خلاصه

در این گزارش، الگوریتم کالمن را به همراه پیاده‌سازی‌اش روی داده‌های قیمتی بیت‌کوین ارایه کردیم. شکل زیر به طور خلاصه متغیرها و الگوریتم را توضیح می‌دهد:

	Equation	Equation Name	Alternative names
Predict	$\boldsymbol{\hat{x}_{n+1,n} = F\hat{x}_{n,n} + Gu_n}$	State Extrapolation	Predictor Equation Transition Equation Prediction Equation Dynamic Model State Space Model
	$\boldsymbol{P_{n+1,n} = FP_{n,n}F^T + Q}$	Covariance Extrapolation	Predictor Covariance Equation
Update (correction)	$\boldsymbol{\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(z_n - H\hat{x}_{n,n-1})}$	State Update	Filtering Equation
	$\boldsymbol{P_{n,n} = (I - K_nH)\,P_{n,n-1}(I - K_nH)^T + K_nR_nK_n^T}$	Covariance Update	Corrector Equation
	$\boldsymbol{K_n = P_{n,n-1}H^T\left(HP_{n,n-1}H^T + R_n\right)^{-1}}$	Kalman Gain	Weight Equation
Auxiliary	$\boldsymbol{z_n = Hx_n}$	Measurement Equation	
	$\boldsymbol{R_n} = E\left(\boldsymbol{v_nv_n^T}\right)$	Measurement Uncertainty	Measurement Error
	$\boldsymbol{Q_n} = E\left(\boldsymbol{w_nw_n^T}\right)$	Process Noise Uncertainty	Process Noise Error

۸ ادامه

به طرق مختلفی می‌توان این مساله را تعمیم داد. در زیر به راهکارهایی برای تعمیم به حالت کلی‌تر سوال می‌پردازیم.

- چهار ویژگی ذکر شده بنا به تجربه قابل مطالعه‌تر از شاخص‌های مرسوم بازار بودند. می‌توان تعداد زیادی شاخص تصادفی تولید کرد و با استخراج ویژگی‌های جدید مناسب از آن‌ها برای استفاده در سری زمانی بهره گرفت.

- مقدار حافظه سری زمانی و مقدار داده‌ای که برای محاسبه پارامترهای دینامیک سری‌های زمانی استفاده کردیم به طور دستی در این گزارش تنظیم شده بود. می‌توان روش‌هایی برای تنظیم خودکار آن‌ها بوجود آورد.
- الگوریتم‌های بکار رفته در کدها دارای انواع پتانسیل‌ها برای بهبود بخشیدن چه از نظر سرعت اجرا چه از نظر دقت و غیره می‌باشند.

مراجع

[BB20] S. Barratt and S. Boyd. Fitting a kalman smoother to data. *Proceedings of the American Control Conference*, 2020.

[Bec] Alex Becker. Kalman filter. URL: `https://www.kalmanfilter.net/default.aspx`.

[Bin] Binance. Binance api. URL: `https://api.binance.com/api/v3/klines?symbol=BTCUSDT&interval=1m&limit=1000`.

[BV] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra.*

[ea19] Alper Akca et. al. Multiple model kalman and particle filters and applications: A survey. *IFAC Papers Online*, 52, 2019.

[ea20] M. Palan et. al. Fitting a linear control policy to demonstrations with a kalman constraint. *Proceedings of Machine Learning Research*, 120:1–10, 2020.

[Gho] Mohsen Ghodrati. Codes. URL: `https://github.com/MohsenGhodrati/cvx_project_Kalman_Filter`.

[Han00] P. D. Hanlon. Multiple-model adaptive estimation using a residual correlation kalman filter bank. *IEEE Transactions on Aerospace and Electronic Systems*, 2000.

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 1960.

[KB] Youngjoo Kim and Hyochoong Bang. Introduction to kalman filter and its applications. URL: `https://www.intechopen.com/chapters/63164`.

[Num] Numpy. Numpy least square solver. URL: `https://numpy.org/doc/stable/reference/generated/numpy.linalg.lstsq.html`.

[Tra] TradingView. Btcusdt chart. URL: `https://www.tradingview.com/`.

[Wik] Wikipedia. Autoregressive model. URL: `https://en.wikipedia.org/wiki/Autoregressive_model`.

[YDB22] S. J. Qin Y. Dong and S. Boyd. Extracting a low-dimensional predictable time series. *Optimization and Engineering*, 23, 2022.