گزارش پروژه ریاضیات مالی
Detecting and repairing arbitrage in traded option prices

محمدسعید حقی، فاطمه ابراهیمی

۳۰ تیر ۱۴۰۲

۱ مقدمه

در مدلسازی‌های اقتصادی نبود آربیتراژ یک اصل است. بنابراین مدل‌هایی که برای قیمت‌گذاری اوراق مشتقه به کار می‌روند نیز از فرض نبود آربیتراژ استفاده می‌کنند. بنابراین اگر در داده‌های قیمتی موجود، آربیتراژ وجود داشته باشد مدلسازی و پیش‌بینی دقیق انجام نمی‌شود. برای رفع این مشکل اکثرا از روش‌هایی مثل filtering و یا smoothing استفاده می‌شود. در روش smoothing هدف این است که یک تابع هموار از قیمت یا نوسان ضمنی (implied volatility) دارایی بسازیم که تا حد ممکن با داده‌های در دسترس همخوانی داشته باشد. در روش filtering نیز هدف این است که با معیارهای مختلف، تعدادی از داده‌های نامناسب را حذف کرده تا آربیتراژ را از بین ببریم.

در مقاله‌ای که بررسی کردیم، هدف این است که روشی ارائه شود تا با کمترین تغییر در داده‌های قیمتی قراردادهای اختیار، آربیتراژ موجود در داده‌ها حذف شود. در این روش، با استفاده از شروط خطی لازم و کافی‌ای که برای نبود آربیتراژ بدست می‌آوریم، یک مسئله بهینه‌سازی خطی با هدف حداقل بودن تغییرات اعمال شده روی داده‌ها را تشکیل می‌دهیم و در نهایت با انتخاب تابع هدف مناسب، به یک الگوریتم کارا برای از بین بردن آربیتراژ از داده‌ها می‌رسیم. لازم به ذکر است که در این روش، نرخ بهره و سود سهام را ناصفر ولی ثابت در نظر گرفته و آربیتراژ را نیز صرفا آربیتراژ ثابت (static arbitrage) در نظر می‌گیریم. در نهایت روش ارائه شده برای اختیار خرید و فروش به کار می‌رود اما برای سادگی، مدلسازی را روی قراردادهای اختیار خرید انجام دادیم.

۲ شروط لازم برای نبود آربیتراژ

تعداد متناهی اختیار خرید روی یک دارایی مشخص با زمان‌های سررسید $T_1 < T_2 < ... < T_m$ در نظر می‌گیریم که برای هر زمان T_i که $i > 0$ قراردادهای اختیار خرید با قیمت‌های اعمال $K_1^i < K_2^i < ... < K_{n_i}^i$ وجود دارند. قیمت قرارداد (i,j) ام را با C_j^i نمایش می‌دهیم.

با استفاده از استراتژی‌هایی که برای اختیار خرید داریم، شروط لازمی روی قیمت این اختیارها برای نبود آربیتراژ بدست می‌آوریم.

vertical spread: تعریف می‌کنیم

$$j_1 > j_2 \ : \ VS^i_{j_1,j_2} = \frac{C^i_{j_2} - C^i_{j_1}}{K^i_{j_1} - K^i_{j_2}}$$

کمیت تعریف شده همان هزینه اولیه استراتژی vertical spread است که در آن $\dfrac{1}{K^i_{j_1} - K^i_{j_1}}$ واحد اختیار خرید با قیمت اعمال $K^i_{j_1}$ خریداری شده و $\dfrac{1}{K^i_{j_1} - K^i_{j_1}}$ واحد اختیار خرید با قیمت اعمال $K^i_{j_2}$ فروخته شده است. بنابراین در صورت نبود آربیتراژ باید $0 \leq VS^i_{j_1,j_2}$ برقرار باشد.
از طرفی باتوجه به نمودار payoff بر حسب قیمت سهام در این استراتژی، می‌بینیم که payoff کران پایین صفر و کران بالای یک دارد بنابراین خواهیم داشت:

$$0 \leq VS^i_{j_1,j_2} \leq 1$$

به طریق مشابه شروط دیگر را با توجه به استراتژی‌های دیگر بدست می‌آیند.

calendar spread:

$$i_1 < i_2 \ : \ CS^{i_1,i_2}_j = C^{i_2}_j - C^{i_1}_j \geq 0$$

calendar vertical spread:

$$i_1 < i_2 \ , \ K^{i_1}_{j_1} > K^{i_2}_{j_2} \ : \ CSV^{i_1,i_2}_{j_1,j_2} = \frac{C^{i_2}_{j_2} - C^{i_1}_{j_1}}{K^{i_1}_{j_1} - K^{i_2}_{j_2}} \geq 0$$

vertical butterfly:

$$K^i_{j_2} > K^i_j > K^i_{j_1} \ : \ VB^i_{j,j_1,j_2} = \frac{C^i_{j_1} - C^i_j}{K^i_j - K^i_{j_1}} - \frac{C^i_j - C^i_{j_2}}{K^i_{j_2} - K^i_j} \geq 0$$

calendar butterfly:

$$i \leq i_1, i_2 \ : \ CB^{i,i_1,i_2}_{j,j_1,j_2} = \frac{C^{i_1}_{j_1} - C^i_j}{K^i_j - K^{i_1}_{j_1}} - \frac{C^i_j - C^{i_2}_{j_2}}{K^{i_2}_{j_2} - K^i_j} \geq 0$$

در نهایت با استفاده از قضایا اثبات می‌شود که شروط بدست آمده هم‌ارز وجود m احتمال ریسک خنثی مرتبط به تمام $T_1, ..., T_m$ است و طبق قضیه اساسی اول قیمت‌گذاری، در این حالت آربیتراژ وجود ندارد.

در نهایت برای افزایش کارایی الگوریتم، تعدادی از شروط اضافه که توسط بقیه شروط پوشش داده می‌شوند را حذف می‌کنیم و به جدول زیر از شرط‌های معادل با نبود آربیتراژ می‌رسیم:

Table 1. The reduced set of static arbitrage constraints.

Category	Constraints	Number
C1 Outright	$i \in [1,m], c^i_{n_i} \geq 0$	m
C2 Vertical spread	$i \in [1,m], j \in [1,n_i]$, $\mathrm{VS}^i_{j,j-1} \geq 0$ and $\mathrm{VS}^i_{1,0} \leq 1$	$N+m$
C3 Vertical butterfly	$i \in [1,m], j \in [1,n_i-1], \mathrm{VB}^i_{j,j-1,j+1} \geq 0$	$N-m$
C4 Calendar spread	$1 \leq i_1 < i_2 \leq m, j_1 \in [0,n_{i_1}], j_2 \in [0,n_{i_2}]$, $\mathrm{CS}^{i_1,i_2}_{j_1,j_2} \geq 0$	$\mathcal{O}(mN)$
C5 Calendar vertical spread	$i^* \in [1,m], j^* \in [1,n_{i^*}]$, define $\mathcal{I} := \{i,j : T_i > T_{i^*}, k^{i^*}_{j^*-1} < k^i_j < k^{i^*}_{j^*}\}$, then $i,j \in \mathcal{I}, \mathrm{CVS}^{i^*,i}_{j^*,j} \geq 0$	$\mathcal{O}(mN)$
C6.1 Calendar butterfly I (Absolute location convexity)	$i^* \in [1,m], j^* \in [1,n_{i^*}-1]$, define $\mathcal{I} := \{i,j : T_i > T_{i^*}, k^{i^*}_{j^*-1} < k^i_j < k^{i^*}_{j^*}\}$, then $i,j \in \mathcal{I}, \mathrm{CB}^{i^*,i,i^*}_{j^*,j,j^*+1} \geq 0$;	$\mathcal{O}(m^2N)$
	$i^* \in [1,m], j^* \in [2,n_{i^*}]$, define $\mathcal{I} := \{i,j : T_i > T_{i^*}, k^{i^*}_{j^*-1} < k^i_j < k^{i^*}_{j^*}\}$, then $i,j \in \mathcal{I}, \mathrm{CB}^{i^*,i^*,i}_{j^*-1,j^*-2,j} \geq 0$;	
	$i^* \in [1,m]$, define $\mathcal{I} := \{i,j : T_i > T_{i^*}, k^i_j > k^{i^*}_{n_{i^*}}\}$, then $i,j \in \mathcal{I}, \mathrm{CB}^{i^*,i^*,i}_{n_{i^*},n_{i^*}-1,j} \geq 0$	
C6.2 Calendar butterfly II (Relative location convexity)	$i^* \in [1,m], j^* \in [1,n_{i^*}-1]$, define $\mathcal{I}_1 := \{i,j : T_i > T_{i^*}, k^{i^*}_{j^*-1} < k^i_j < k^{i^*}_{j^*}\}$, $\mathcal{I}_2 := \{i,j : T_i > T_{i^*}, k^{i^*}_{j^*} < k^i_j < k^{i^*}_{j^*+1}\}$, $i_1,j_1 \in \mathcal{I}, \quad i_2,j_2 \in \mathcal{I}_2, \mathrm{CB}^{i^*,i_1,i_2}_{j^*,j_1,j_2} \geq 0$;	$\mathcal{O}(m^2N)$
	$i^* \in [1,m]$, define $\mathcal{I}_1 := \{i,j : T_i > T_{i^*}, k^{i^*}_{n_{i^*}-1} < k^i_j < k^{i^*}_{n_{i^*}}\}$, $\mathcal{I}_2 := \{i,j : T_i > T_{i^*}, k^i_j > k^{i^*}_{n_{i^*}}\}$, $i_1,j_1 \in \mathcal{I}, \quad i_2,j_2 \in \mathcal{I}_2, \mathrm{CB}^{i^*,i_1,i_2}_{n_{i^*},j_1,j_2} \geq 0$	

۳ حل مساله حذف آربیتراژ

در اینجا ما مساله ترمیم داده به فرم برنامه‌ریزی خطی مینویسیم تا در قسمت بعدی به کمک ابزارهای حل دستگاه‌های برنامه‌ریزی خطی، مساله حذف آربیتراژ را حل کنیم.
با توجه به اینکه در قسمت قبل قیدهایی برای تضمین نبود آربیتراژ پیدا کردیم و تمام این قیود به شکل نامساوی‌های خطی بودند، پس میتوانیم کل این دستگاه نامعادله را به شکل یک نابرابری ماتریسی به شکل $Ac \geq b$ بازنویسی کنیم.
ماتریس A و بردار b هر دو ماتریس‌های ثابتی هستند که با توجه به قیود و اطلاعاتی از داده که در قسمت‌های قبلی بررسی شد بدست آمده‌اند. بردار c هر بردار متناظر با قیمت‌ها در زمان صفر میباشد. میدانیم لزوما بردار اولیه ما در قیود ما صدق نمیکند، بنابراین تصمیم داریم با تغییر این بردار و تبدیل آن به بردار $c + \epsilon$ کاری کنیم که بردار جدید در قیود ما صدق کند. یعنی به بیان برنامه‌ریزی خطی قصد ما این است که $A(c + \epsilon) \geq b$ که این امر نتیجه می‌دهد:

$$A(c + \epsilon) \geq b \Longrightarrow A\epsilon \geq b - Ac$$

از طرفی هدف ما این است که کمترین میزان تغییرات ممکن را بر روی داده‌های اولیه اعمال کنیم یا به عبارتی میخواهیم ϵ به زبانی کوچک باشد. از این روی تابع $f : \mathbb{R}^N \to \mathbb{R}$ را تعریف میکنیم و مساله بهینه‌سازی خطی ما به این شکل در می‌آید:

$$\begin{aligned} \text{minimize} \quad & f(\epsilon) \\ \text{subject to} \quad & A\epsilon \geq b - Ac \end{aligned} \tag{۱}$$

جواب این مساله به نحوه‌ای انتخاب می‌شود که با حداقل تغییرات (ϵ) بتوانیم قیود را ارضا کنیم. این مساله که تابع f را به چه شکل انتخاب کنیم کاملا جای بحث دارد. چهار پیشنهاد مختلف پیش روی ما میباشد که آن‌ها را معرفی خواهیم کرد.

۱. $l_2 - norm$

۲. $l_0 - norm$

۳. $l_1 - norm$

۴. $l_1BA - norm$

آیتم یک باعث میشود با یک مساله برنامه‌ریزی خطی محدب رو به رو باشیم که به راحتی برایمان قابل حل است. ولیکن خاصیتی که در کمینه سازی این نرم وجود دارد باعث میشود که جواب بدست آمده در نهایت درایه‌های ناصفر زیاد و در عوض کوچکی داشته باشد. این جواب برای ما مطلوب نیست و ما با جواب‌های اسپارس با درایه‌های صفر زیاد که نشان دهنده کمترین تعداد تغییر در قیمت‌ها می‌باشد را بیشتر مطلوب میدانیم. از این رو نرم دوم برای ما جالب‌تر است زیرا دقیقا تعداد درایه‌های ناصفر در ϵ را میشمارد.
اما مشکلی که این نرم دارد، نامحدب بودن آن است، درواقع حل این مساله معادل با حل یک مساله ترکیبیاتی بسیار زمانبر می‌باشد که برای مطلوب نیست. نرم l_1 به عنوان تقریبی محدب و خطی از همین نرم استفاده میشود. بنابراین یکی از انتخاب‌های خوب ما برای تابع f نرم l_1 خواهد بود.
ایده نرم آخر این است که تابع f را به شکلی تعریف کنیم که در آن از مقادیر Bid و Ask نیز استفاده کرده باشیم. زیرا هنگام ترمیم داده ما میخواهیم تا جای ممکن قیمت‌هایمان در بازه bid-ask قرار بگیرد. با توجه به اینکه با اضافه کردن این قیود ممکن است مساله ما ناشدنی (infeasible) شود، بنابراین نمیتوانیم صرفا به اضافه کردن آنها به لیست قیود اکتفا کنیم. بلکه سعی میکنیم با تغییر تابع هدف، این قیود را به شکل soft constraints درون مساله‌مان جا دهیم. کافی است تابع هدف ما ۵ خاصیت زیر را دارا باشد:

۱. $f(\epsilon) = \sum f_j(\epsilon_j)$

۲. $f_j(0) = inf_x f_j(x) = 0$

۳. f در مقادیر مثبت صعودی و در مقادیر منفی نزولی باشد

۴. $f_j(-\delta_j^b) = f_j(\delta_j^a) = \delta_0$

۵. $\frac{df_j(x)}{d|x|} = 1$ به ازای مقادیر خارج بازه bid-ask

که در آن δ_j^b و δ_j^a به ترتیب مقادیر bid reference spread و ask reference spread می باشند. حال یک تابع معرفی میکنیم که قیود بالا را ارضا خواهد کرد:

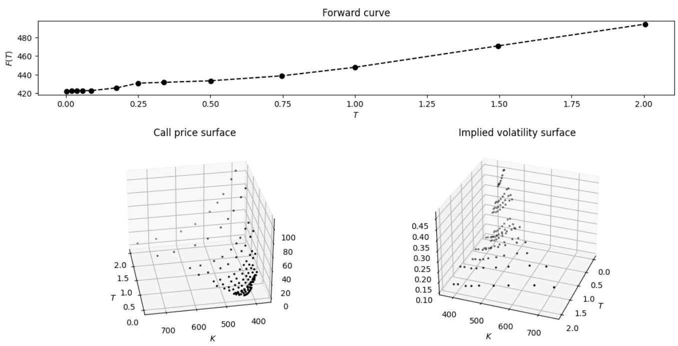
$$f_j(x) = max(-x - \delta_j^b + \delta_0, x - \delta_j^a + \delta_0, -\frac{\delta_0}{\delta_j^b}x, \frac{\delta_0}{\delta_j^a}x)$$

که در آن $\delta_0 \leq min(\delta_j^b, \delta_j^a)$. این تابع هدف را نیز میتوان به شکل یکسری تابع خطی بازنویسی کرد و به این ترتیب به یک مساله برنامه‌ریزی خطی تبدیل کرد که به راحتی قابل حل است.

۴ کاربرد عملی الگوریتم

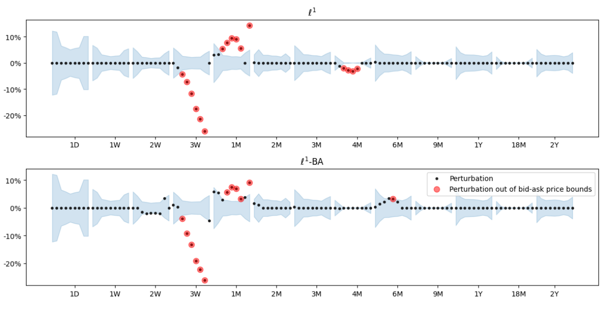
اهمیت روش‌های اصلاح حذف آربیتراژ داده، از جایی نشات می‌گیرند که با بررسی داده‌های تاریخی بازارهای جهانی متوجه می‌شویم که آربیتراژ در داده‌های تاریخی وجود دارد. در واقع اگر روی داده‌های خام در دسترس، الگوریتم را اجرا کنیم مشاهده می‌کنیم که قیمت‌ها هر کدام تعدادی از شروط بدست آمده برای نبود آربیتراژ را نقض می‌کنند. برای استفاده از الگوریتم بدست آمده، در این مقاله از داده‌های موجود قیمت قراردادهای اختیار خرید بازار FOREX استفاده کردیم.

در شکل زیر می‌توان نمودار قیمت قرارداد آتی روی دارایی پایه بر حسب زمان را مشاهده کرد. در دو نمودار دیگر نیز، نقاط متناظر هر قراداد را در فضای سه بعدی مشخص کردیم. در یکی از آن‌ها، قیمت قرارداد بر حسب قیمت اعمال و زمان سررسید نمایش داده شده و در دیگری، کمیت نوسان ضمنی برای هر قراداد بر حسب قیمت اعمال و زمان سررسید مشخص شده است.



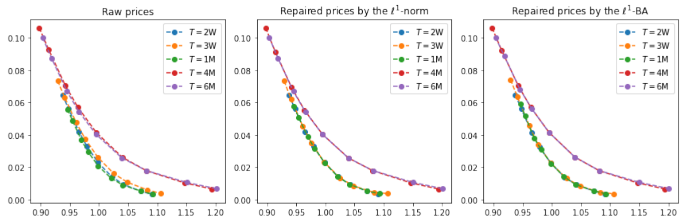
لازم به ذکر است که از قیمت قراردادهای آتی برای نرمال کردن متغیرهای قیمت قراردادهای اختیار و قیمت اعمال آنها استفاده می‌شود. خود قرارداد آتی نیز به عنوان یک اختیار خرید با قیمت اعمال 0 و قیمت قرارداد $F(T)$ در نظر گرفته می‌شود تا از استراتژی‌های آربیتراژ به وسیله آن‌ها نیز جلوگیری شود.

الگوریتم را روی داده‌ها به دو روش اجرا می‌کنیم. در یک روش از تابع هدف $\ell^1 - norm$ و در روش دیگر از تابع هدف $\ell^1 - BA$ استفاده کردیم. در نمودار زیر می‌توان تفاوت این دو روش را در میزان تغییری که روی دیتا اعمال می‌کنند مشاهده کرد.



در هر بخش آبی، نقطه‌ها متناظر تغییر ایجاد شده در داده‌هایی با زمان سررسید یکسان و قیمت اعمال‌های متفاوت است. این بازه نمایانگر bid-ask spread است و بنابراین نقاط خارج آن‌ها، درایه‌هایی از ϵ هستند که مقدار قابل توجهی دارند. همان‌طور که مشاهده می‌کنیم در روش $\ell^1 - BA$ تغییرات قابل توجه کمتری اعمال شده است.

در شکل زیر نیز، برای بخشی از قراردادها، نمودار قیمت قرارداد بر حسب قیمت اعمال برای تاریخ سررسیدهای مختلف رسم کردیم. سپس با دو روش گفته شده الگوریتم حذف آربیتراژ را اجرا کرده و نتایج را در دو نمودار نشان دادیم:



۵ کاربرد الگوریتم در افزایش دقت مدلسازی‌ها

در انتها برای دیدن تاثیر الگوریتم حذف آربیتراژر، مدل هستون را در بررسی کردیم. در مدل هستون قبل از شروع، الگوریتم را روی داده‌ها اعمال کردیم تا آربیتراژ را از بین ببرد و در نهایت دیدیم که دقت پیش‌بینی در این مدل نسبت به حالتی که داده‌ها دارای آربیتراژ بودند بیشتر شد.