

## استفاده از آمارگان مقاوم در جدا سازی کورسیگنال برای کاهش اثر داده های پرت

کریم سلیمی، محمد امیر فلاح

۱- دانشگاه آزاد اسلامی واحد نیریز

۲- پژوهشکده هیدروفیزیک شیراز

### چکیده:

در این مقاله مساله جداسازی کورسیگنال را با استفاده از روشهای آمارگان مقاوم بررسی میکنیم. الگوریتمهای متداول جداسازی کورسیگنال نسبت به انحراف داده از مدل فرضی بسیار حساس هستند. به این معنی که وجود تعدادی داده پرت در میان داده ها، باعث واگرایی الگوریتم و یا کند شدن همگرایی آن میشود. برای مقابله با این مشکل از روشهای آمارگان مقاوم استفاده میکنیم. به این صورت که ابتدا داده های مورد استفاده در الگوریتم از یک بخش پردازش اولیه عبور میکنند تا اثر داده های پرت تا حد ممکن کاهش یابد. سپس این داده های جدید به الگوریتم داده میشود. تا کنون از این روش در جدا سازی کور استفاده نشده است. با شبیه سازیهای انجام داده شده نشان داده ایم که همگرایی الگوریتمها با داده های جدید بهتر شده است. همچنین نشان داده ایم که در محیطهای نویزی کارایی روش جدید از روشهای قبلی بهتر است. البته این بهبود کارایی به بهای کاهش سرعت همگرایی بدست میآید.

**کلمات کلیدی:** جداسازی کور، داده های پرت، آمارگان مقاوم

### ۱- مقدمه

جدا سازی کور سیگنال مساله ای است که با آن در حوزه های مختلف علوم کاربردی و مهندسی روبرو میشویم که از جمله آنها میتوان به پردازش صحبت، انتقال داده، پردازش آرایه ای، داده های مربوط به زلزله و پردازش داده های حیاتی اشاره کرد. به دلیل کاربردهای وسیع جدا سازی کور سیگنال الگوریتمها و روشهای زیادی برای آن ابداع شده است. [۳ و ۷ و ۱۰] مساله مورد بحث در جدا سازی کور سیگنال این است که با در دست داشتن تعدادی ترکیب خطی یا فیلتر شده از منابع مستقل بتوانیم منابع اولیه را استخراج کنیم.

به زبان ریاضی فرض کنید  $s_1(t), \dots, s_n(t)$  سیگنال مستقل آماری باشند که  $n$  ترکیب  $x_1(t), \dots, x_n(t)$  از آنها را داریم. این ترکیبها خطی و لحظه ای هستند یعنی برای هر  $i=1, \dots, n$  داریم:  $x_i(t) = \sum_{j=1}^n a_{ij} s_j(t)$ . این رابطه را میتوان به صورت ماتریسی زیر

بیان کرد:

$$x(t) = As(t) \quad (1)$$

که  $s(t) = [s_1(t), \dots, s_n(t)]^T$  و  $x(t) = [x_1(t), \dots, x_n(t)]^T$ .

به ماتریس مربعی  $n \times n$ ،  $A$  ماتریس ترکیب گفته میشود. برای استخراج منابع اولیه، بردار مشاهده را در یک ماتریس جداسازی  $B$  ضرب میکنیم:

$$y(t) = Bx(t) \quad (2)$$

از آنجا که مشاهدات ترکیب خطی از منابع هستند منطقی است که برای استخراج منابع از ترکیب خطی آنها استفاده کنیم. برای تعیین  $B$  روشهای زیادی وجود دارد. از جمله میتوان به [۱۲ و ۱۱ و ۹] اشاره کرد.

## ۲- برخی الگوریتمهای استفاده شده در جدا سازی کور

### ۲-۱- پردازش اولیه

معمولا سیگنالهای مشاهده  $x(t)$  پیش از پردازش سفید میشوند. نشان داده شده است که این کار در افزایش سرعت همگرایی موثر است. [۶] و [۱۱] برای سفید کردن داده فرض کنید متوسط  $X$  صفر باشد و تجزیه ماتریس کوواریانس  $X$  را در نظر بگیرید  $C_x = E\{xx^T\} = Q\Lambda Q^T$  برای سفید کردن داده کافی است آن را در  $V = Q\Lambda^{-1/2}Q^T$  ضرب کنیم. زیرا با فرض  $z(t) = Vx$  داریم:

$$E\{zz^T\} = E\{Vxx^TV^T\} = (VQ)\Lambda(VQ)^T = (Q\Lambda^{-1/2})\Lambda(Q\Lambda^{-1/2})^T = QQ^T = I$$

از این به بعد در این مقاله با داده سفید شده  $Z$  سروکار داریم.

### ۲-۲- روشهای مبتنی بر کورتوسیس (Kurtosis)

میتوان نشان داد که هر چه توزیع  $y(t)$  ها از توزیع گوسی دور تر باشد عمل جدا سازی بهتر صورت میگیرد [۲ و ۸]. لذا برای جدا سازی کافی است معیاری از نزدیکی توزیع یک سیگنال به توزیع گوسی داشته باشیم. یکی از این معیارها کورتوسیس است که به صورت زیر تعریف میشود:

$$Kurt(x) = E\{x^4\} - 3E\{x^2\}^2 \quad (3)$$

این تابع تنها برای توزیع گوسی صفر میشود و قدر مطلق آن میتواند معیاری از جدا سازی سیگنالها باشد.

فرض کنید  $W$  ماتریس جداسازی مربوط به  $Z$  باشد در این صورت برای ارضای شرط جداسازی باید رابطه زیر برقرار باشد:

$$Wz = PDs \quad (4)$$

در این رابطه  $P$  یک ماتریس جایگشت (ماتریسی با تنها یک عدد یک در هر سطر و ستون و بقیه عناصر صفر) و  $D$  یک ماتریس قطری است که نشان دهنده مقیاس است.

برای سادگی توجه خود را به حالتی معطوف میکنیم که تنها جدا سازی یکی از منابع مورد نظر باشد. در حالت کلی میتوان نشان داد با انجام مکرر این کار و اعمال شرط تعامد بر بردارهای بدست آمده میتوان تمام منابع را استخراج کرد.

لذا اگر  $w^T$  یکی از سطرهای  $W$  باشد برای جداسازی صحیح باید داشته باشیم:  $\hat{s}_i = w^T z$ ، که  $\hat{s}_i$  تخمین یکی از منابع است. (فرقی نمیکند کدام منبع باشد) حال معیار کورتوسیس را بر  $\hat{s}_i$  اعمال میکنیم:

$$Kurt(\hat{s}_i) = Kurt(w^T z) = E\{(w^T z)^4\} - 3E\{(w^T z)^2\}^2 \quad (5)$$

با مشتق گیری نسبت به  $w$  بدست میاوریم [۱۱]:

$$\frac{\partial |Kurt(w^T z)|}{\partial w} = 4 \text{sign}(Kurt(w^T z)) \times [E\{z(w^T z)^3\} - 3w\|w\|^2] \quad (6)$$

و با اعمال چند شرط ساده کننده الگوریتم گرادیان به صورت زیر در می آید:

$$\begin{aligned} w_{k+1} &= E\{z(w_k^T z)^3\} - 3w_k \\ \|w_{k+1}\| &= 1 \end{aligned} \quad (7)$$

این الگوریتم یکی از الگوریتمهای مهم در جداسازی کور است.

در قمت بعد به بررسی اثر داده های پرت و ارایه روشی جهت کاهش آنها میپردازیم.

### ۳- الگوریتم پیشنهادی برای کاهش اثر داده های پرت

#### ۳-۱- داده های پرت و اثر آنها

منظور از داده پرت داده ای است که از مدل مفروض برای داده ها پیروی نکند [۷۵]. اثر این نوع داده ها را در این مقاله به صورت نویز جمع شونده با داده اصلی در نظر گرفته ایم. یعنی سیگنال مشاهده سفید شده به صورت زیر است:

$$z_{ol}(t) = z(t) + \eta(t)$$

که  $\eta(t)$  نویز مورد نظر است. داده های پرت اغلب باعث واگرایی الگوریتمهای جدا سازی کور میشوند و در بهترین حالت همگرایی آنها را کند میکنند.

دلیل آن این است که تمام الگوریتمهای جدا سازی کور به نوعی (مستقیم یا غیر مستقیم) تابع چگالی منابع مستقل و بردار مشاهده ارتباط دارند و برای انجام جدا سازی از تخمینهای این توابع چگالی استفاده میکنند.

داده های پرت در حالت کلی باعث ایجاد اختلاف زیاد بین توابع چگالی مدل فرضی با توابع چگالی داده های مشاهده شده و لذا منجر به واگرایی یا کند شدن همگرایی الگوریتمهای جداسازی کور میشوند.

روشهای robust (مقاوم) مختلفی برای پیش پردازش داده ها وجود دارد [۸] که از جمله ساده ترین آنها استفاده از فیلترهای میانه (Median filter) و میانگین (Mean filter) است که در این مقاله از آنها استفاده کرده ایم. یک فیلتر میانه به طول  $2L+1$  فیلتری است که بردار ورودی  $Z$  را میگیرد و با شروع از عنصر اول، میانه  $L$  داده قبل و بعد داده، به همراه خود داده را به عنوان خروجی برمیگرداند. یعنی اگر  $y_{median}$  خروجی فیلتر میانه باشد داریم:

$$y_{i_{median}} = \text{median}\{z_{i-L}, \dots, z_i, \dots, z_{i+L}\} \quad (8)$$

فیلتر میانگین نیز با رابطه زیر بیان میشود:

$$y_{i_{mean}} = \text{mean}\{z_{i-L}, \dots, z_i, \dots, z_{i+L}\} = \frac{1}{2L+1} \sum_{k=-L}^{k=L} z_{i+k} \quad (9)$$

### ۲-۳- روش پیشنهادی

ما در روش پیشنهادی خود ابتدا داده های مشاهده را از یک فیلتر مقاوم (Robust) عبور دادیم. سپس این داده ها به الگوریتم اصلی داده میشود و الگوریتم از روی داده های جدید تخمینی از بردار  $w$  را که جدا سازی را انجام میدهد بدست می آورد. این الگوریتم آنقدر

تکرار میشود تا  $w$  همگرا شود. منابع اولیه نیز از رابطه  $\hat{s}_i = w_{conv}^T z$  بدست می آیند. ( $w$ ،  $w_{conv}$  هنگام همگرایی است).

در بدست آوردن تخمین منبع، از بردار اولیه  $Z$  استفاده میشود ولی در تخمین  $w$  از  $Z$  فیلتر شده استفاده میشود.

با توجه به آنچه گفته شد الگوریتم مورد نظر به صورت زیر در می آید:

الگوریتم پیشنهادی:

(10)

$$z_f = \text{filter}(z)$$

for  $k = 1, \dots$

$$1: w_{k+1} = E\{z_f (w_k^T z_f)^3\} - 3w_k$$

$$w_{k+1} = w_{k+1} / \|w_{k+1}\|$$

if  $w$  not converges goto 1

$$\hat{s}_i = w_{conv}^T z$$

### ۴- معیار جدا سازی

فرض کنید جداسازی کامل انجام گرفته باشد، در این صورت  $\hat{s}_i = w^T z = w^T Vx = w^T VAs = p^T s$  که  $p = w^T VA$ .

برای اینکه  $w^T z$  یکی از منابع باشد لازم است که  $p$  تنها شامل یک عنصر غیر صفر باشد یعنی:

$$p = [0 \dots p_i \dots 0]^T \quad \hat{s}_i = p_i s_i \text{ تخمین منبع } \hat{s}_i \text{ خواهد بود.}$$

با توجه به آنچه گفتیم میتوان نسبت جدا سازی را به صورت زیر تعریف کرد:

$$SepI = \frac{[Max(p_i)]^2}{\sum_{i=1}^n p_i^2} \quad (11)$$

در حالت جداسازی کامل  $SepI = 1$  خواهد بود.

نکته مهم این است که تنها همگرایی الگوریتم برای جداسازی کافی نیست؛ بلکه علاوه بر آن بردار  $p$  باید به فرم ذکر شده باشد.

در بخش بعد نتایج شبیه سازی ارایه میشود.

### ۵- نتایج شبیه سازی

الگوریتم پیشنهادی برای دونوع داده اجرا شده است یکی داده عادی  $z$  (فیلتر نشده) و دیگری داده فیلتر شده  $z_f$ . استفاده از داده عادی

برای مقایسه روش پیشنهادی با روش عادی است.

در شبیه سازی این روش میتوان حالت های مختلفی از توزیع منابع، داده های پرت و نوع فیلتر کردن داده ها در نظر گرفت. نتایجی که ارائه میشوند متوسط نتایج در دو حالت زیر هستند:

۱- تابع چگالی منابع یکنواخت و داده های پرت با توزیع گوسی یا پواسن

۲- تابع چگالی منابع لاپلاسی و داده های پرت با توزیع گوسی یا لاپلاسی

برای هر حالت الگوریتم ۱۰۰ بار اجرا شده است و در نهایت از نتایج میانگین گرفته شده است.

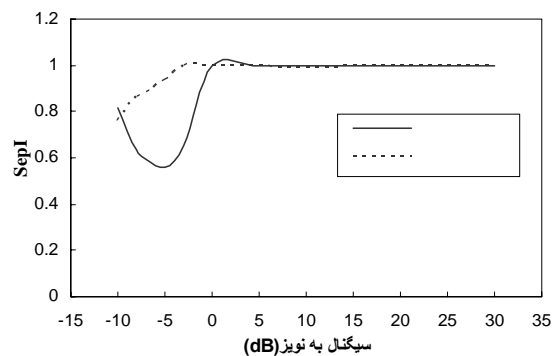
در شکل های داده شده نمودار های مربوط به نسبت جداسازی (*Sepl*) را در سیگنال به نویز های مختلف و همچنین نمودار های مربوط به همگرایی  $w$  را در هر حالت مشاهده میکنید. منظور از نویز در اینجا همان داده های پرت است یعنی سیگنال به نویز به صورت  $SNR = \frac{\sigma_s^2}{\sigma_n^2}$  تعریف میشود که  $\sigma_s^2$  و  $\sigma_n^2$  به ترتیب واریانس های منابع و داده های پرت هستند.

در حالت اول توزیع داده ها یکنواخت است. همانطور که در شکل ۱ مشاهده میکنید الگوریتم با داده عادی در سیگنال به نویز های زیر 0dB همگرا نمیشود ولی الگوریتم پیشنهادی که با داده فیلتر شده کار میکند همگرا میشود.

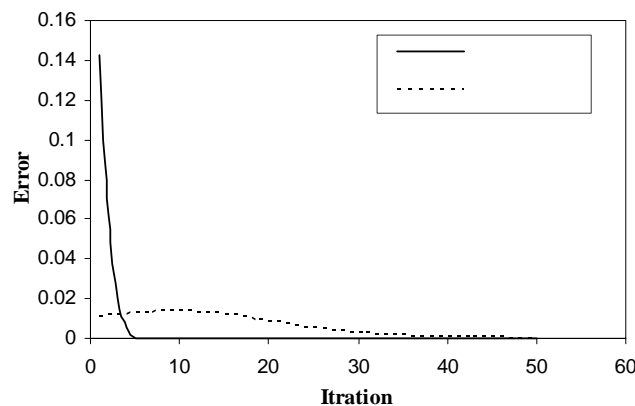
در شکل های ۲ و ۳ نمودار های همگرایی  $w$  را برای دو سیگنال به نویز مختلف مشاهده میکنید.

در شکل ۴ هم نسبت همگرایی برای داده های با توزیع لاپلاسی دیده میشود. و شکل های ۵ و ۶ نیز نمودار های همگرایی  $w$  در دو سیگنال به نویز خاص و برای داده های با توزیع لاپلاسی هستند.

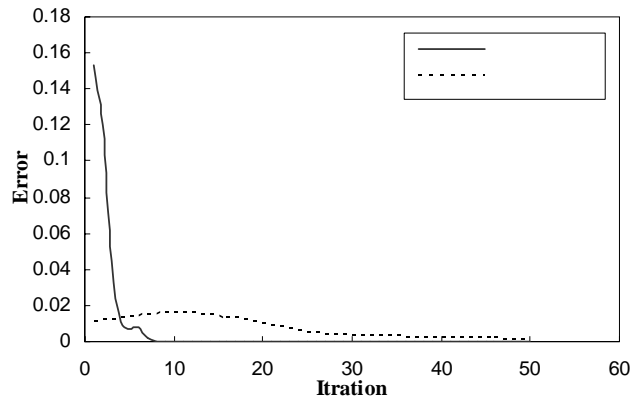
به طوری که دیده میشود و انتظار داریم، همگرایی ضرایب  $w$  در الگوریتم پیشنهادی کندتر از حالت عادی است و هر چه سیگنال به نویز کمتر باشد اختلاف سرعت همگرایی بیشتر میشود. ولی نکته این است که در  $SNR$  پایین الگوریتم عادی عمل جدا سازی را انجام نمیدهد.



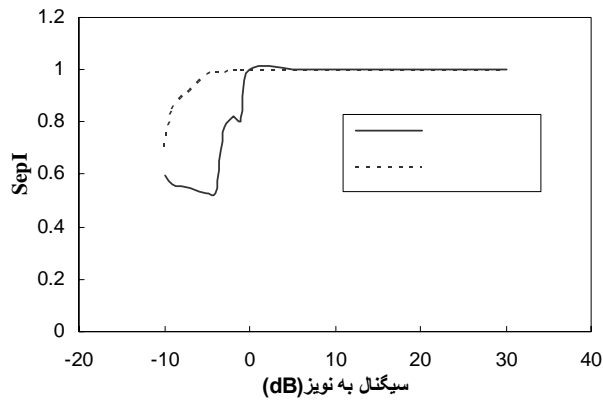
شکل ۱: نسبت جدا سازی الگوریتم پیشنهادی و الگوریتم استاندارد برای داده با توزیع یکنواخت



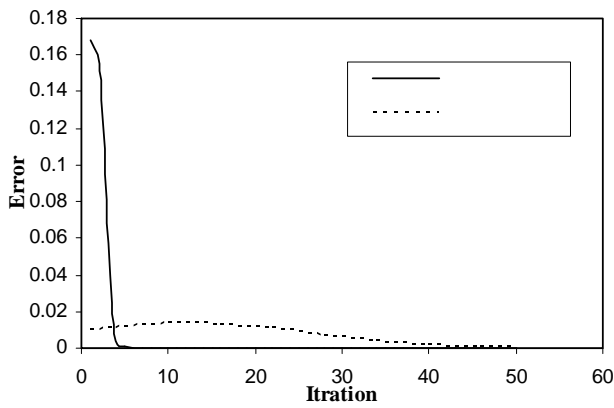
شکل ۲: خطای همگرایی ضرایب  $w$  در  $SNR = -4dB$  و توزیع داده یکنواخت



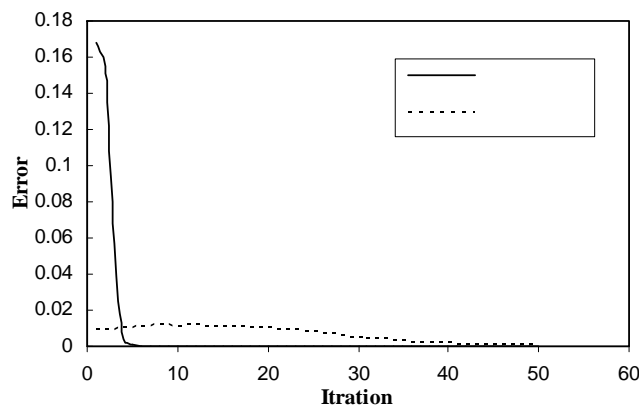
شکل ۳: خطای همگرایی ضرایب  $w$  در  $SNR=10dB$  و توزیع داده یکنواخت



شکل ۴: نسبت جدا سازی الگوریتم پیشنهادی و الگوریتم استاندارد برای داده با توزیع لاپلاسی



شکل ۵: خطای همگرایی ضرایب  $w$  در  $SNR=-4dB$  و توزیع داده لاپلاسی



شکل ۶: خطای همگرایی ضرایب  $w$  در  $SNR=10dB$  و توزیع داده لاپلاسی

## ۶- نتیجه

الگوریتم پیشنهادی در محدوده وسیعی از سیگنال به نویزها (یا به طور معادل تغییرات داده های پرت) عمل جداسازی را با موفقیت انجام میدهد، در حالی که الگوریتم استاندارد (عادی) برای سیگنال به نویزهای پایین بسیار ضعیف عمل میکند. البته برای تمام سیگنال به نویزها، سرعت همگرایی الگوریتم پیشنهادی از الگوریتم عادی کمتر است. در نهایت این الگوریتم روش قابل اطمینانی برای حصول به جداسازی کور سیگنال در محدوده وسیعی از سیگنال به نویزها ارائه می کند.

## ۷- مراجع

- [1] S.I. Amari, "Natural gradient works efficiently in learning," Neural Computation, vol. 10, pp. 251-276, 1998
- [2] K. Anand, G. Mathew, and V. Reddy, "Blind separation of multiple co-channel BPSK signals arriving at an antenna array," IEEE Signal Processing Lett., vol. 2, pp. 176-178, Sept. 1995.
- [3] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," Neural Computation, vol. 7, no. 6, pp. 1004-1034, 1995.
- [4] C. Jutten and J. Herault, "Blind separation of sources. An adaptive algorithm based on neuro mimetic architecture," Signal Processing, vol. 24, no. 1, pp. 1-10, July 1991.
- [5] J. Karhunen, A. Hyvarinen, R. Vigario, J. Hurri, and E. Oja, "Applications of neural blind separation to signal and image processing," in Proc. ICASSP, vol. 1, 1997, pp. 131-134.
- [6] S. Makeig, A. Bell, T. P. Jung, and T. J. Sejnowski, "Independent component analysis of electroencephalographic data," in Advances in Neural Information Processing Systems, vol. 8. Denver, CO: MIT Press, 1995.

- [7] I.Pitas and A. N. Venetsanopoulos. Nonlinear digital filters and applications. Boston MA.Kluwer Academic, 1990
- [8] G. Arce and N. C. Gallagher, “Stochastic analysis for the recursive median filter process” ,IEEE Trans. Inform. Theory, vol. 34,pp. 669-679,July 1988.
- [9] A. Nandi, editor. Blind estimation using higher order statistics, kluwer 1999.
- [10] S. degerine and R.malki. “Second order blind separation of sources based on canonical partials”.IEEE Trans. On Signal processing, vol. 48, pp. 629-641, 2000.
- [11] A. J. Bell. Information theory, Independent component analysis, and applications. In S. haykin, editor, Unsupervised adaptive filtering, vol. 1, pp. 237-264,Wiley,2000.