به نام خدا

درس جداسازی کور منابع

خلاصه مقاله

عنوان مقاله:

Off-Grid Direction of Arrival Estimation Using Sparse Bayesian Inference

دانشگاه تهران

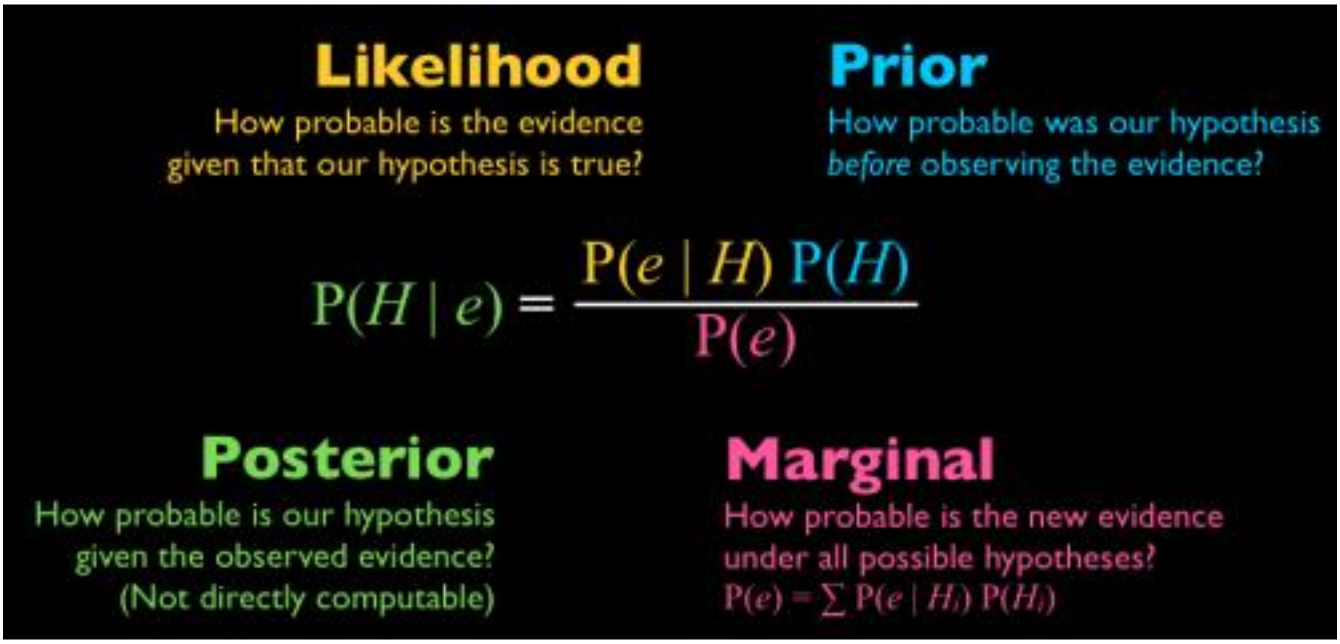
1402/03/20

جدول محتویات

مقدمه:

در این مقاله، روشی جدید برای حل مسئله‌ی DoA ارائه شده است که مبتنی بر استفاده از دانش اولیه نسبت به مسئله بر اساس چهارچوب Bayesian است. در این چهارچوب که در دانش آمار پایه‌ای ارائه می‌شود، هر رخداد به عنوان دانش اولیه برای پیش‌بینی رخداد بعدی استفاده می‌شود.

در واقع به صورت زیر می‌توان داشت:



شکل 1 فرمت مدل کردن منطق بیضی

در این چهارچوب، دانش اولیه به عنوان توزیع اولیه برای ما نقش ایفا می‌کند و سپس با وقوع اتفاق جدید دانش ما به روزمی‌شود. در واقع برای حل مسئله‌ی DoA که همان Direction of Arrival یا جهت ورود یا جهت رسیدن هست، دانش ابتدائی ما می‌تواند Sparse بودن جواب باشد که همان در واقع کم بودن تعداد اهداف در یک زاویه خاص است.

در این مقاله سعی بر ارائه روشی شده است که به صورت off-Grid عمل کند! به این صورت که دیگر محدود به ماتریس Dictionary انتخاب شده‌ی اولیه نخواهیم بود. در واقع نشان داده می‌شود که این محدود بودن به ماتریس Dictionary اولیه ناشی از حذف تقریب مرتبه اول از بسط تیلور خواهد بود. در واقع مشکلی که در روش‌های On-Grid بود، رعایت یک trade-off بین استقلال یا ناهمبستگی ستون‌ها و همچنین پوشش حداکثری و افزایش رزولوشن زاویه‌یابی بود. با افزایش

رزولوشن، شرط ناهمبستگی بین ستون‌ها به مرور رنگ می‌بازد و دیگر به شرایط RIP یا همان Restricted isometry property نخواهیم رسید.

معرفی روش Off-Grid.

به صورت مرسوم، برای مدل کردن سیگنال دریافتی در یک ارایه داریم:

$$y(t) = A(\theta)s(t) + e(t); \quad t = 1, \dots, T$$

که در آن داریم:

$$y(t) = [y_1(t), \dots, y_M(t)]; \quad \theta = [\theta_1, \dots, \theta_K]^T; \qquad s(t) = [s_1(t), \dots, s_K(t)]^T;$$

و همچنین برای ماتریس manifold داریم:

$$A(\theta) = [a(\theta_1), \dots, a(\theta_K)]$$

که در آن $a(\theta_K)$ همان steering vector های ما هستند!

اگر حال فرض کنیم که برای زاویه‌ی فوق بتوانیم تخیمی بزنیم خواهیم داشت:

$$a(\theta_K) \approx a(\hat{\theta}_{n_K}) + b(\hat{\theta}_{n_K})(\theta_K - \hat{\theta}_{n_K})$$

که همان تخمین مرتبه‌ی اول تیلور هست! پس به این ترتیب $b(\hat{\theta}_{n_K}) = a'(\hat{\theta}_{n_K})$ که با جایگذاری و در نظر گرفتن فرم ماتریسی داریم:

$$y(t) = \Phi(\beta)x(t) + e(t); \quad t = 1, \dots, T$$

که در آن:

$$\beta_n = \theta_K - \hat{\theta}_{n_K}, x_n(t) = s_{n_K}(t) \; if \;\; n = n_K \; for \; any \; K \in \{1, \dots, K\}$$

در واقع با قرار دادن مقدار $\beta = 0$ در معادله‌ی فوق، همان تقریب مرتبه 0 تیلور که معادله‌ی فرم اولیه برای گیرنده بود، به دست خواهد آمد.

- *Off-Grid* را می‌توان به عنوان تقریب مرتبه اول برای مدل مشاهدات در گیرنده در نظر گرفت. به این ترتیب با مقدار مشخصی نمونه برداری، به خطای کمتری به نسبت *On-Grid* خواهیم رسید که برای خطای برابر مقدار کمتر *WorkLoad* را نتیجه می‌دهد.

صرفا متغیر جدیدی به نام β به این مسئله اضافه شده و باید علاوه بر مقادیر sparse در x(t) به دنبال این مقدار هم باشیم. در ادامه این مسئله به صورت چهارچوب بیضی فرمول‌بندی شده و به صورت الگوریتم تکرارشونده حل خواهد شد.

مدل OGSBI یا همان Off-Grid Sparse Bayesian Inference

در این مدل، فرمول‌بندی برای حالت MMV یا همان Multiple Measurement Vector در نظر گرفته خواهد شد. با درنظر گرفتن نویز به صورت گوسی داریم:

$$Y = \Phi(\beta)X + E; \qquad \Phi(B) = A + B * diag(\beta)$$

که با در نظر داشتن گریدبندی به صورت یکنواخت با طول r به صورت

$$r = \widetilde{\theta_2} - \widetilde{\theta_1} \propto N^{-1}$$

در نظر گرفته می‌شود. پس برای β داریم:

$$\beta \in \left[-\frac{1}{2}r, \frac{1}{2}r\right]^N$$

برای مدل کردن سیگنا به صورت *Sparse* داریم:

$$p(X|\alpha) = \prod_{t=1}^{T} \mathcal{CN}(x(t)|0, \Lambda)$$

که در واقع در آن توزیع α را به صورت توزیع گاما در نظر گرفته ایم که در آن $\alpha_0 = \sigma^2$ همان واریناس نویز است.$(\Lambda = diag(\alpha))$

به صورت کلی خواهیم داشت:

$$p(X|Y,\alpha_0,\alpha,\beta)=\prod_{t=1}^{T}\mathcal{CN}(x(t)|\mu(t),\Sigma)$$

With $\mu(t)=\alpha_0\Sigma\Phi^H y(t), t=1,\dots,T;\quad \Sigma=(\alpha_0\Phi^H\Phi+\Lambda^{-1})^{-1}$.

برای به دست آوردن پارامترهای اساسی مدل که همان α_0,α,β هستند از تخمین‌گر MAP استفاده می‌شود که منتج به قواعد به‌روزرسانی زیر خواهد شد:

$$\alpha_n^{new}=\frac{\sqrt{1+4\underline{\rho}E\left\{\|\underline{\boldsymbol{X}}^n\|_2^2\right\}}-1}{2\underline{\rho}},\quad n=1,\cdots,N,$$

$$\alpha_0^{new}=\frac{M+\frac{c-1}{T}}{E\left\{\|\underline{\boldsymbol{Y}}-\boldsymbol{\Phi}\underline{\boldsymbol{X}}\|_{\mathrm{F}}^2\right\}+\frac{d}{T}},$$

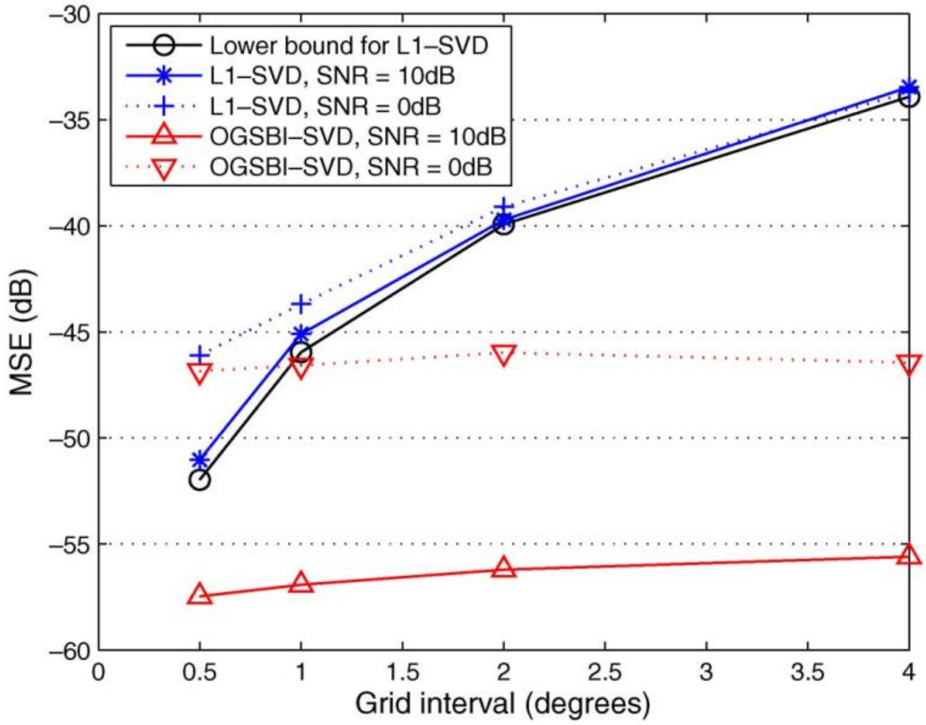
$$\boldsymbol{\beta}^{new}=\arg\min_{\boldsymbol{\beta}\in\left[-\frac{1}{2}r,\frac{1}{2}r\right]^N}\left\{\boldsymbol{\beta}^T\boldsymbol{P}\boldsymbol{\beta}-2\boldsymbol{v}^T\boldsymbol{\beta}\right\}.$$

با داشتن این پارامترها و قواعد به روزرسانی هر یک از آنها، مدل همگرا خواهد شد.(به صورت تضمینی به دلیل انتخاب درست تابع هزینه و ویژگی‌های آن)

- در ادامه می‌توان از SVD برای denoise کردن و کاهش ابعاد مسئله به فضای سیگنال، استفاده کرد.

نتایج مدل

در نهایت، مدل پیشنهادی با یک مدل که به خوبی عمل کرده و مشهور شده است مقایسه کردیده:



شکل 2 مقایسه‌ی بین عملکرد الگوریتم L1-SVD و روش ارائه شده

همچنین برای مقایسه‌ی زمان اجرای این الگوریتم‌ها در جدول زیر خواهیم داشت:

SNR = 10dB				
	$r = 0.5^\circ$	$r = 1^\circ$	$r = 2^\circ$	$r = 4^\circ$
ℓ_1-SVD	0.601	0.413	0.324	0.291
OGSBI-SVD	10.2	0.782	0.096	0.025
SNR = 0dB				
	$r = 0.5^\circ$	$r = 1^\circ$	$r = 2^\circ$	$r = 4^\circ$
ℓ_1-SVD	0.413	0.295	0.218	0.190
OGSBI-SVD	10.9	0.831	0.104	0.024

3 شکل

که با توجه به انتخاب مقادیر بیشتر از 2 برای r بر حسب درجه، مدل پیشنهادی بسیار بهتر و سریعتر از $SVD - \ell_1$ عمل می‌کند.

در زیر مقایسه‌ی بین عملکرد و زمان اجرا به صورت میانگین بر روی CPU را مشاهده خواهیم کرد:

	MSE (dB)		Time (sec)	
	$r = 2°$	$r = 4°$	$r = 2°$	$r = 4°$
STLS	-36.5	-36.6	5.31	1.77
OGSBI	-45.1	-43.3	0.098	0.028

شکل 4

پایان