فصل ۲

کلیات نمونه‌برداری فشرده

۲-۱ مقدمه

در این فصل نمونه‌برداری فشرده را از چند دیدگاه بررسی می‌کنیم. ابتدا پیش زمینه‌های ریاضی که باعث پیدایش نظریه نمونه‌برداری فشرده شده است را بررسی می‌کنیم. عرض گلفاند مجموعه‌ها مبحثی است که سال‌ها قبل از نمونه‌برداری فشرده مطرح شده است و ارتباط تنگاتنگی با این نظریه دارد [۳۸]. از طرف دیگر، نمونه‌برداری فشرده از دیدگاه نظریه‌ی اطلاعات یک روش فشرده‌سازی تلقی می‌شود. بدین منظور نمونه‌برداری فشرده از دیدگاه نرخ اعوجاج[۱] را نیز بررسی می‌کنیم و در ادامه مرور مختصری بر روش‌های بازسازی ارائه می‌دهیم.

۲-۲ عرض گلفاند

فرض کنید S زیرمجموعه‌ای فشرده از $\mathbb{R}^n$ و m عددی طبیعی باشد. عرض گلفاند[۲] مجموعه S را به صورت زیر تعریف می‌کنیم:

$$p \geq ۱ : \quad d^m(S)_{\ell_p} = \inf_{Y} \ \sup\{\|\mathbf{x}\|_{\ell_p} \mid \mathbf{x} \in S \cap Y\} \tag{۱-۲}$$

که در آن infimum روی تمام زیرفضاهای Y ($\mathbb{R}^n$) با بعد حداقل $n-m$ گرفته می‌شود. نشان می‌دهیم که عملکرد نمونه‌برداری فشرده روی مجموعه S، رابطه بسیار نزدیکی با عرض گلفاند مجموعه دارد [۳۳]. فرض

[۱] Rate-Distortion
[۲] Gelfand's width

کنید S زیرمجموعه‌ای کراندار (با نرم ℓ_p) از $\mathbb{R}^n$ باشد به طوری که قرینه جمعی هر بردار داخل S عضوی از S باشد:

$$\forall\, s \in S: \quad -s \in S \tag{۲-۲}$$

همچنین فرض کنید جمع هر دو عضو S ، ضریبی (ثابت) از یک عنصر S باشد:

$$\exists\, c_\circ \in \mathbb{R}: \quad S + S \subset c_\circ S \tag{۳-۲}$$

مثلاً اگر S مجموعه بردارهای با نرم کمتر از واحد باشد، می‌توان $c_\circ = ۲$ اختیار کرد. می‌خواهیم اعضای S را توسط ماتریس $\mathbf{\Phi}_{m\times n}$ نمونه‌برداری کنیم (در حالت کلی اعضای $\mathbb{R}^n$ را نمونه‌برداری می‌کنیم اما در بازسازی به دنبال یافتن ورودی مناسب در S هستیم). برای بازسازی نیز از تابعی به نام D استفاده می‌کنیم که لزوماً خطی نیست:

$$\begin{cases} \mathbf{y}_{m\times ۱} = \mathbf{\Phi}_{m\times n} \cdot \mathbf{x}_{n\times ۱} \\ \hat{\mathbf{x}}_{n\times ۱} = D\big(\mathbf{y}_{m\times ۱}\big) \end{cases} \tag{۴-۲}$$

از آنجا که معیار سنجش ما نرم ℓ_p است، خطای حاصل از بازسازی توسط D به صورت زیر خواهد بود:

$$E\big(\mathbf{x}_{n\times ۱}, \mathbf{\Phi}_{m\times n}, D\big) = \big\|\mathbf{x}_{n\times ۱} - \hat{\mathbf{x}}_{n\times ۱}\big\|_{\ell_p} = \big\|\mathbf{x}_{n\times ۱} - D\big(\mathbf{\Phi}_{m\times n}\cdot \mathbf{x}_{n\times ۱}\big)\big\|_{\ell_p} \tag{۵-۲}$$

برای بررسی عملکرد نمونه‌برداری-بازسازی تولید شده توسط $\mathbf{\Phi}_{m\times n}$ وD، تنها خطای ایجاد شده روی یک عضو کفایت نمی‌کند بلکه باید به نوعی خطای ایجاد شده روی تمام حالات ورودی بررسی شود. مثلاً:

$$E\big(S, \mathbf{\Phi}_{m\times n}, D\big)_{\ell_p} = \sup_{\mathbf{x}_{n\times ۱}\in S} E\big(\mathbf{x}_{n\times ۱}, \mathbf{\Phi}_{m\times n}, D\big)_{\ell_p} \tag{۶-۲}$$

در نمونه‌برداری فشرده، در پی بهترین زوج D و $\mathbf{\Phi}_{m\times n}$ هستیم؛ به عبارت دیگر، می‌خواهیم ماتریس حسگر و روش بازسازی به نوعی انتخاب شوند که خطای ایجاد شده روی اعضایS به حداقل ممکن برسد. با انتخاب بهترین زوج D و $\mathbf{\Phi}_{m\times n}$ (با ثابت نگه داشتن m) $E\big(\mathbf{x}_{n\times ۱}, \mathbf{\Phi}_{m\times n}, D\big)_{\ell_p}$ به کمترین مقدار خود می‌رسد که این مقدار، بهترین سطح خطای ایجاد شده در بازسازی اعضای S است و فقط تابعی از S و m (و نرم ℓ_p) خواهد بود:

$$E_m(S)_{\ell_p} = \inf_{\mathbf{\Phi}_{m\times n}, D} E\big(S, \mathbf{\Phi}_{m\times n}, D\big)_{\ell_p} \tag{۷-۲}$$

اکنون می‌توان نشان داد که $E_m(S)_{\ell_p}$ به طور جالبی به $d^m(S)_{\ell_p}$ مربوط است : از آنجا که ماتریس $\mathbf{\Phi}_{m\times n}$ در نمونه‌برداری فشرده یک ماتریس مستطیلی افقی است ($m \le n$)، بعد فضای پوچ ($dim\ \mathcal{N}_{\mathbf{\Phi}}$) آن حداقل $n-m$ خواهد بود که انتخاب مناسبی برای Y در تعریف عرض گلفاند است:

$$d^m(S)_{\ell_p} = \inf_{Y} \sup_{\mathbf{x}\in S\cap Y} \|\mathbf{x}\|_{\ell_p} \le \sup_{\mathbf{x}\in S\cap N_{\mathbf{\Phi}}} \|\mathbf{x}\|_{\ell_p} \tag{۸-۲}$$

از طرفی، برای هر $\mathbf{x} \in S\cap\mathcal{N}_{\mathbf{\Phi}}$ داریم: $\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1} = 0_{m\times 1}$ یعنی برای این بردارهای $\mathbf{x}$، $D(\mathbf{\Phi},\mathbf{x}) = D(0)$ و ثابت است.

$$\mathbf{x}\in S\cap\mathcal{N}_{\mathbf{\Phi}} \Rightarrow -\mathbf{x}\in S\cap\mathcal{N}_{\mathbf{\Phi}} \Rightarrow D(\mathbf{\Phi}\cdot\mathbf{x}) = D(-\mathbf{\Phi}\cdot\mathbf{x}) = D(0)$$

$$\begin{aligned} &\|\mathbf{x}-D(0)\|_{\ell_p} + \|-\mathbf{x}-D(0)\|_{\ell_p} \ge 2\|\mathbf{x}\|_{\ell_p} \\ &\Rightarrow \begin{cases} E(\mathbf{x}_{n\times 1},\mathbf{\Phi}_{m\times n},D)_{\ell_p} \ge \|\mathbf{x}\|_{\ell_p} \\ \quad or \\ E(-\mathbf{x}_{n\times 1},\mathbf{\Phi}_{m\times n},D)_{\ell_p} \ge \|\mathbf{x}\|_{\ell_p} \end{cases} \end{aligned} \tag{۹-۲}$$

در نتیجه برای هر $\mathbf{x}\in S\cap\mathcal{N}_{\mathbf{\Phi}}$ عنصری در این مجموعه وجود دارد که خطای بازسازی آن حداقل به اندازه $\|\mathbf{x}\|_{\ell_p}$ است:

$$\begin{aligned} E(S,\mathbf{\Phi}_{m\times n},D)_{\ell_p} &= \sup_{\mathbf{x}_{n\times 1}\in S} E(\mathbf{x}_{n\times 1},\mathbf{\Phi}_{m\times n},D)_{\ell_p} \\ &\ge \sup_{\mathbf{x}\in S\cap\mathcal{N}_{\mathbf{\Phi}}} E(\mathbf{x}_{n\times 1},\mathbf{\Phi}_{m\times n},D)_{\ell_p} \ge \sup_{\mathbf{x}\in S\cap\mathcal{N}_{\mathbf{\Phi}}} \|\mathbf{x}\|_{\ell_p} \end{aligned} \tag{۱۰-۲}$$

پس مشاهده شد که عرض گلفاند مجموعه، یک کران پایین برای خطای بازسازی در هر انتخاب $D,\mathbf{\Phi}_{m\times n}$ است. پس:

$$E_m(S)_{\ell_p} \ge d^m(S)_{\ell_p} \tag{۱۱-۲}$$

نامساوی فوق به تنهایی اهمیت خاصی ندارد زیرا فقط یک کران پایین است و معلوم نیست که تا چه حد میتوان به این کران نزدیک شد. نکته قابل توجه این است که $E_m(S)_{\ell_p}$ از بالا نیز محدود به ضریبی از $d^m(S)_{\ell_p}$ می‌شود: فرض کنید Y زیرفضایی با بعد $n-m$ از $\mathbb{R}^n$ باشد؛ در نتیجه بعد فضای متعامد Y ($Y^\perp$) برابر با m خواهد بود. فرض کنید $\mathbf{v}_1,\dots,\mathbf{v}_m$ پایه‌هایی برای این زیر فضا ($Y^\perp$) باشند. اکنون ماتریس $\mathbf{\Phi}$ نمونه‌برداری را بر حسب این

Y چنین می‌سازیم:

$$\mathbf{\Phi}_{m\times n}=\begin{bmatrix}\mathbf{v}_1^T\\ \vdots\\ \mathbf{v}_m^T\end{bmatrix} \tag{۱۲-۲}$$

همچنین تابع بازسازی D را چنین تعریف می‌کنیم: برای یک $\mathbf{u}_{m\times 1}\in\mathbb{R}^m$ دلخواه، اگر $\mathbf{u}_{m\times 1}\in\mathbf{\Phi}\cdot S$، $D(\mathbf{u}_{m\times 1})$ را به دلخواه یکی از اعضای S مانند $\mathbf{a}$ تعریف می‌کنیم که $\mathbf{u}_{m\times 1}=\mathbf{\Phi}\cdot\mathbf{a}$؛ درغیر این صورت $D(\mathbf{u}_{m\times 1})$ را بـه طور تصادفی یکی از اعضای S قرار می‌دهیم. با مفروضات فوق، $E_m(S)_{\ell_p}$ را بدست می‌آوریم. اگر $\mathbf{x}_{n\times 1}\in S$، داریم:

$$\begin{aligned}\mathbf{\Phi}_{m\times n}\big(\mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})\big) &= \mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1}-\mathbf{\Phi}_{m\times n}\cdot D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})\\ &= \mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1}-\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1}=0_{m\times 1}\\ \Rightarrow\quad & \mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})\in\mathcal{N}_{\mathbf{\Phi}_{m\times n}}\\ \Rightarrow\quad & \mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})\in Y\end{aligned} \tag{۱۳-۲}$$

از طرفی طبق فرض اولیه، تفاضل هر دو عضو S ضریبی از یک عضو S خواهد بود:

$$\exists c_0\in\mathbb{R}:\ \frac{\mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})}{c_0}\in S\Rightarrow\frac{\mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})}{c_0}\in S\cap Y \tag{۱۴-۲}$$

حال داریم :

$$\begin{aligned}&E(S,\mathbf{\Phi}_{m\times n},D)_{\ell_p}=c_0\sup_{\mathbf{x}_{n\times 1}\in S}\left\|\frac{\mathbf{x}_{n\times 1}-D(\mathbf{\Phi}_{m\times n}\cdot\mathbf{x}_{n\times 1})}{c_0}\right\|_{\ell_p}\le c_0\sup_{\mathbf{x}_{n\times 1}\in S\cap Y}\|\mathbf{x}_{n\times 1}\|_{\ell_p}\\ &\Rightarrow\inf_{\mathbf{\Phi}_{m\times n},D}E(S,\mathbf{\Phi}_{m\times n},D)_{\ell_p}\le c_0\inf_{Y}\sup_{\mathbf{x}_{n\times 1}\in S\cap Y}\|\mathbf{x}_{n\times 1}\|_{\ell_p}\\ &\Rightarrow E_m(S)_{\ell_p}\le c_0\ d^m(S)_{\ell_p}\end{aligned} \tag{۱۵-۲}$$

پس در کل خواهیم داشت:

$$d^m(S)_{\ell_p}\le E_m(S)_{\ell_p}\le c_0\ d^m(S)_{\ell_p} \tag{۱۶-۲}$$

در نتیجه، یافتن عملکرد بهترین نحوه نمونه‌برداری–بازسازی روی S تقریباً معادل با یافتن عرض گلفاند این مجموعه است. متاسفانه یافتن عرض گلفاند مجموعه‌ها جز در حالاتی خاص (و آن هم نه به سادگی) هنوز

مساله‌ای باز محسوب می‌شود [۶۳]. از موارد خاص می‌توان به عرض گلفاند گوی واحد n بعدی با نرم ℓ_p اشاره کرد $(U(\ell_p^n))$:

$$U(\ell_p^n) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_{\ell_p} = ۱\} \tag{۲–۱۷}$$

در چنین حالتی $d^m\big(U(\ell_p^n)\big)_{\ell_q}$ ، بین دو کران بالایی و پایینی محدود می‌شود که کران بالایی ضریبی از کران پایینی است اما مقدار دقیقی برای این ضریب بدست نیامده است؛ تنها مرتبه بزرگی $d^m\big(U(\ell_p^n)\big)_{\ell_q}$ مشخص است [۶۳]. به هر حال، عرض گلفاند را می‌توان به عنوان مرجع سنجشی برای خطای ایجاد شده در یک روش نمونه‌برداری فشرده تلقی کرد (مانند باند Cramer-Rao در تخمین؛ با این تفاوت که در کمترین خطا، لزوماً به عرض گلفاند دست نخواهیم یافت).

۲–۳ بررسی نمونه‌برداری فشرده از دیدگاه نظریه نرخ–اعوجاج

در نمونه‌برداری فشرده، هـدف ذخیره‌سازی دسته خاصی از سیگنال‌ها به کمک تعداد نمونه کم است. پس می‌توان به نوعی این روش را جزء روش‌های کدگذاری منبع طبقه بندی کرد. از آنجا که این روش لزوماً به بازسازی دقیق سیگنال منجر نمی‌شود، یک روش فشرده‌سازی با اتلاف محسوب می‌شود؛ در نتیجه بررسی عملکرد آن باید توسط ابزارهای مطرح شده در نظریه نرخ–اعوجاج[۳] صورت پذیرد. فرض کنید معیار سنجش اعوجاج $(\mathcal{D})$ به صورت MSE باشد :

$$\mathcal{D} = \mathscr{E}\{(x - \hat{x})^۲\} \tag{۲–۱۸}$$

همچنین فرض کنید نرخ ذخیره‌سازی اطلاعات بر حسب بیتR باشد؛ مثلاً اگر سیگنالی را که در بازه‌ای به طول c توزیع یکنواخت دارد، با استفاده از چندی‌سازی[۴] با پله‌های Δ گسسته کنیم :

$$R = \log_۲\left(\frac{c}{\Delta}\right) \tag{۲–۱۹}$$

اگر توزیع ورودی را مشابه فرض فوق یکنواخت فرض کنیم، متغیر $x - \hat{x}$ (اختلاف مقدار اصلی با مقدار چندی‌شده) نیز در بازه $[-\frac{\Delta}{۲}, \frac{\Delta}{۲}-]$ توزیع یکنواخت خواهد داشت:

$$\mathcal{D} = \mathscr{E}\{(x - \hat{x})^۲\} = \frac{\Delta^۲}{۱۲} \tag{۲–۲۰}$$

[۳] Rate-Distortion

[۴] quantization

و در نتیجه، رابطه‌ی بین نرخ ذخیره‌سازی بیت و اعوجاج به صورت $\mathcal{D} = \frac{c^۲}{۱۲}۲^{-۲R}$ خواهد بود. در حالت کلی تر، اگر سیگنال ورودی توزیع دلخواه داشته باشد، در [۵۳] نشان داده شده است که بهترین (کمترین) میزان اعوجاج در چندی‌کردن این سیگنال با نرخ ثابت R در رابطه‌ی زیر صدق می‌کند:

$$\frac{۲^{۲h}}{۲\pi e}۲^{-۲R} \leq \mathcal{D} \leq \sigma^۲ ۲^{-۲R} \qquad (۲۱-۲)$$

که در آن h و $\sigma^۲$ به ترتیب آنتروپی پیوسته و واریانس سیگنال ورودی هستند. جمله‌ی اصلی در روابط اعوجاج که از نرخ حاصل می‌شود، $۲^{-۲R}$ است؛ یعنی وابستگی اعوجاج به نرخ بیت به صورت ضریبی از $۲^{-۲R}$ خواهد بود. اکنون مسأله‌ی نمونه‌برداری فشرده را برای تحلیل نرخ–اعوجاج مدل می‌کنیم: فرض کنید ماتریس $\mathbf{\Psi}_{n\times n}$ یک ماتریس متعامد–یکه تصادفی است که تنها در گیرنده (عملیات بازسازی) معلوم است و همچنین فرض کنید بردار تنک $\mathbf{s}_{n\times ۱}$ از kدرایه ناصفر تشکیل شده باشد که انتخاب این k محل با توزیع یکنواخت بین $\binom{n}{k}$ حالت مختلف صورت گرفته است و درایه‌های ناصفر به طور مستقل از هم، تـوزیع گـوسی با متوسط صفر و واریانس یک دارند. بردار زمانی نیز به صورت $\mathbf{x}_{n\times ۱} = \mathbf{\Psi}_{n\times n} \cdot \mathbf{s}_{n\times ۱}$ در نظر گرفته می‌شود که پس از نمونه‌برداری توسط ماتریس $\mathbf{\Phi}_{m\times n}$ به $\mathbf{y}_{m\times ۱} = \mathbf{\Phi}_{m\times n} \cdot \mathbf{x}_{n\times ۱}$ تبدیل می‌شود. اکنون باید نمونه‌های بدست آمده (y_i) را چندی‌کرد:

$$\begin{aligned} \hat{y}_i &= Q(y_i) \\ \beta = \frac{\mathscr{E}\{|y_i - \hat{y}_i|^۲\}}{\mathscr{E}\{|y_i|^۲\}} &\quad , \quad \rho = ۱ - \beta \\ \Rightarrow \hat{y}_i &= \rho y_i + \nu_i \end{aligned} \qquad (۲۲-۲)$$

که در آن ν_i نویز چندی‌کردن است و واریانس آن برابر با $\beta(۱-\beta)\mathscr{E}\{|y_i|^۲\}$ است (ν_i با y_i نابسته خواهد بود اما $y_i - \hat{y}_i$ نسبت به y_i نابسته نیست):

$$\hat{\mathbf{y}} = \underbrace{\rho\mathbf{\Phi} \cdot \mathbf{\Psi}}_{\mathbf{A}} \cdot \mathbf{s} + \nu = \mathbf{A} \cdot \mathbf{s} + \nu \qquad (۲۳-۲)$$

فرض کنیم برای ذخیره هر نمونه ناصفر، R بیت اختصاص یافته باشد؛ در این صورت برای ذخیره‌سازی بردار $\mathbf{s}$ و در نتیجه $\mathbf{x}$، در کل $R.k$ بیت اختصاص یافته‌است. اکنون درپی محاسبه اعوجاج هستیم. اگر از ابتدا تمام نمونه‌های $\mathbf{x}$ را بدون توجه به شرط تنک‌بودن $\mathbf{s}$ چندی می‌کردیم، به هر درایه $\frac{kR}{n}$ بیت اختصاص می‌یافت و در نتیجه:

$$\mathcal{D}_{direct} \propto ۲^{-۲\frac{kR}{n}} \qquad (۲۴-۲)$$

اگر می‌شد فقط نمونه‌های ناصفر s را ذخیره کرد، به غیر از k نمونه ناصفر، باید مکان‌های ناصفر نیز ذخیره می‌شد؛ پس بطور متوسط $R_* = \frac{\log_2 \binom{n}{k}}{k}$ بیت از بیت‌های اختصاص داده شده به هر نمونه، باید به این امر اختصاص یابد. بنابراین:

$$\mathcal{D}_{adaptive} \propto 2^{-2(R-R_*)} \tag{۲۵–۲}$$

در نمونه‌برداری فشرده که m نمونه در اختیار داریم، به هر نمونه $\frac{kR}{m}$ بیت اختصاص خواهد یافت. بدیهی است که با افزایش m (تعداد نمونه‌ها)، بازسازی از روی نمونه‌های چندسطحی نشده دقیق‌تر می‌شود؛ حال آنکه با افزایش m، هر نمونه با کیفیت پایین‌تری ذخیره می‌شود (نرخ کلی ثابت) و در نهایت اعوجاج سیگنال نهایی افزایش می‌یابد. در [۴۷] نشان داده شده که در نرخ‌های R متوسط به بالا، مقدار بهینه m از رابطه زیر بدست می‌آید:

$$m \approx \frac{2kR\ln 2}{2R\ln 2 - 1} \tag{۲۶–۲}$$

برای m رابطه فوق چنین خواهیم داشت:

$$\mathcal{D}_{CS} \propto 2^{-2(R-R^*)} \tag{۲۷–۲}$$

که :

$$R^* = \frac{\log_2(2eR\ln 2)}{2} \tag{۲۸–۲}$$

مقایسه $\mathcal{D}_{CS}$ با $\mathcal{D}_{adaptive}$ حاکی از آن است که کیفیت بطور قابل ملاحظه‌ای افت کرده که تقریباً نتیجه واضحی است زیرا $\mathcal{D}_{adaptive}$ با تقریب خوبی بهترین کیفیت قابل دسترسی است. در بازسازی، ابتدا باید محل نمونه‌های ناصفر s تخمین زده شود و سپس مقادیر آن‌ها بدست آید؛ نکته منفی که در [۵۲] ذکر شده این است که حتی با افزایش تعداد نمونه‌ها (m) تضمینی برای تخمین درست از این مکان‌ها بدست نمی‌آید.

۲–۴ ارتباط نمونه‌برداری فشرده با نظریه کدگذاری

نویز جمعی مشکلی غیرقابل پرهیز در انواع مخابرات محسوب می‌شود. در مخابرات دیجیتال، به دلیل وجود نویز، تعدادی از بیت‌های ارسالی دستخوش تغییر می‌شوند که در صورت تصحیح نشدن، می‌توانند در گیرنده

مشکلات بسیاری بوجود آورند. سازوکار رایج در آشکارسازی و تصحیح بیت‌های خطا، استفاده از کدگذاری است. در روش کدگذاری قالبی[۵]، یک قالب kتایی از بیت‌های داده، با افزودن تعدادی بیت توازن[۶] به یک قالب nتایی تبدیل می‌شود. از آن‌جا که تعداد حالات قالب kتایی ورودی ۲^k است، تنها ۲^k حالت از ۲^n بردار ممکن در فضای nتایی متعلق به بردارهای کد هستند. در نتیجه هنگامی که تعدادی از بیت‌های یک بردار کد بر اثر نویز جمعی تغییر می‌کنند، بردار نهایی با احتمال زیاد، بردار کد نخواهد بود.

در اکثر کانال‌های مخابراتی، هنگامی که توان سیگنال به اندازه کافی بالا باشد، احتمال تغییر یک بیت توسط نویز، پایین است. از این‌رو در بردار نویزی دریافتی، انتظار تعداد بیت غلط کمی داریم. به عبارت بهتر، اختلاف بین بردارهای ارسالی و دریافتی، یک بردار تنک است. پس، برای بازیابی بردار کد ارسالی، باید نزدیک‌ترین بردار کد به بردار نویزی دریافتی را بیابیم؛ البته در این‌جا، نزدیک و دور بودن بردارها براساس فاصله همینگ[۷] سنجیده می‌شود.

از دیدگاه قابلیت تصحیح خطا، کد مناسب کدی است که فاصله بین بردارهای کد تا حد امکان زیاد باشد؛ معیار سنجش مورد استفاده، اغلب حداقل فاصله بین بردارهای کد است. به عنوان مثال، اگر حداقل فاصله بین بردارهای کد ۵ باشد (حداقل تعداد بیت‌های متفاوت بین دو بردار کد)، به هر نحوی که حداکثر ۴ بیت از یک بردار کد تغییر کنند، بردار نهایی متعلق به فضای کد نخواهد بود و وجود خطا قابل آشکارسازی است. همچنین اگر حداکثر ۲ بیت خطا وجود داشته باشد، بردار کد اولیه در فاصله حداکثر ۲ از بردار نویزی قرار دارد، حال آن‌که سایر بردارهای کد حداقل فاصله ۳ را نسبت به این بردار دارند. بنابراین با یافتن نزدیک‌ترین بردار کد به بردار دریافتی، خطای ایجاد شده به طور صحیح برطرف می‌شود.

نکته جالب توجه در نظریه کدگذاری این است که اگر یک نحوه کدگذاری برای طول ورودی k و طول خروجی n با حداقل فاصله d_{min} بین بردارهای کد وجود داشته باشد، حتماً یک کدگذاری خطی با همین طول‌ها و حداقل فاصله وجود دارد. در نتیجه، از لحاظ قابلیت تصحیح خطا، کدهای خطی بهینه‌اند [۶۶]. در کدگذاری قالبی خطی، ارتباط بین بردار بیت ورودی ($\mathbf{u}_{k\times ۱}$) و بردار بیت خروجی ($\mathbf{c}_{n\times ۱}$) توسط ماتریس مولد کد قابل

[۵] Block Coding
[۶] Parity
[۷] Hamming Distance

بیان است:

$$\mathbf{c}_{n\times ۱} = \mathbf{G}_{n\times k} \cdot \mathbf{u}_k \qquad (۲۹-۲)$$

در رابطه ماتریسی فوق، تمام اعمال (جمع و ضرب) در میدان $GF(۲)$ صورت می‌گیرند. همچنین تاثیر نویز بر روی بردار ارسالی را می‌توان توسط بردار خطا ($\mathbf{w}_{n\times ۱}$) به صورت زیر بیان کرد:

$$\mathbf{r}_{n\times ۱} = \mathbf{c}_{n\times ۱} + \mathbf{w}_{n\times ۱} \qquad (۳۰-۲)$$

(جمع در میدان $GF(۲)$) از آن‌جا که رتبه ماتریس $\mathbf{G}_{n\times k}$، حداکثر k است، ماتریس $\mathbf{H}_{(n-k)\times n}$ وجود دارد که $\mathbf{H}_{(n-k)\times n} \cdot \mathbf{G}_{n\times k} = ۰_{(n-k)\times k}$. این ماتریس به ماتریس آزمون توازن شهرت دارد. اکنون با ضرب کردن طرفین رابطه (۲-۳۰) در $\mathbf{H}_{(n-k)\times n}$ داریم:

$$\mathbf{s}_{(n-k)\times ۱} = \mathbf{H}_{(n-k)\times n} \cdot \mathbf{r}_{n\times ۱} = \mathbf{H}_{(n-k)\times n} \cdot \mathbf{w}_{n\times ۱} \qquad (۳۱-۲)$$

به بیان بهتر، بردار $\mathbf{s}_{(n-k)\times ۱}$ تنها متاثر از بردار خطاست. عمل بازیابی بردارکد که هم ارز با یافتن بردار خطاست، در حقیقت معادل به دست‌آوردن بردار تنک $\mathbf{w}_{n\times ۱}$ از دستگاه فرو-معین $\mathbf{s}_{(n-k)\times ۱} = \mathbf{H}_{(n-k)\times n} \cdot \mathbf{w}_{n\times ۱}$ است. این مسأله به وضوح منطبق بر مسأله نمونه‌برداری فشرده است. علاوه بر این، در نظریه کدگذاری، مشابه نمونه‌برداری فشرده، هدف طراحی ماتریس $\mathbf{G}_{n\times n}$ و متعاقباً $\mathbf{H}_{(n-k)\times}$ است به نحوی که قابلیت بازسازی بردارهای تنک خطا را فراهم کند. شایان ذکر است که در هر دو مبحث، ماتریس‌های تصادفی عملکردی مناسب و فراتر از ماتریس‌های یقینی از خود نشان می‌دهند.

با وجود شباهت بسیار زیاد مسأله نمونه‌برداری فشرده با مسأله طراحی کد، تفاوت عمده در نحوه محاسبات ماتریسی است. در مبحث نمونه‌برداری فشرده، تمامی اعمال در میدان اعداد حقیقی و یا مختلط صورت می‌گیرند در حالی که در نظریه کدگذاری، میدان‌های متناهی مورد توجه هستند. در میدان‌های متناهی، به دلیل ساختار گسسته المان‌ها، مفهوم فاصله قابل تعریف نیست؛ به عبارت دیگر، برخلاف میدان اعداد حقیقی، میدان‌های متناهی توپولوژی مناسبی ندارند. برای روشن‌تر شدن این موضوع، یک دستگاه معادلات خطی n-معادله n-مجهول را در میدان اعداد حقیقی در نظر بگیرید؛ درصورتی که دترمینان ماتریس ضرایب ناصفر باشد، این دستگاه به‌طور یکتا قابل حل است اما بسته به عدد حالت این ماتریس، ممکن است جواب بسیار ناپایدار و یا از نظر عددی، غیرقابل محاسبه باشد. در مقابل، به دلیل متناهی بودن مرتبه میدان، چنین مشکلی

هیچ‌گاه در میدان‌های متناهی رخ نمی‌دهد. برای مثال، ماتریس‌های واندرموند از جمله گزینه‌های خوب برای ماتریس آزمون توازن به‌شمار می‌روند، حال آن‌که به دلیل مشکل ناپایداری، برای ماتریس حسگر انتخاب‌های مناسبی به حساب نمی‌آیند.

۲-۵ مروری بر روش‌های بازسازی

شرط RIP از مرتبه $2k$ با $\delta_{2k} \leq 0$ تضمین می‌کند که در حالت بدون نویز هیچ دو بردار k-تنک‌ای نمونه‌های یکسان تولید نمی‌کنند و در نتیجه بردار k-تنک به طور یکتا از روی نمونه‌ها قابل بازسازی است. اما نحوه‌ی بازسازی بهینه هنوز مشخص نیست. خوشبختانه روش‌هایی که در زیر به آن‌ها اشاره می‌شود، در حد قابل قبولی عمل بازسازی را میسر می‌سازند.

۲-۵-۱ کمینه کردن نرم ℓ_1

همانطور که قبلاً اشاره شد، روش بهینه در بازسازی سیگنال‌های تنک، کمینه کردن نرم ℓ_0 است که در عمل ممکن نیست. به علاوه، اگر چنین کاری ممکن باشد، به دلیل ناپایدار بودن نرم ℓ_0 وجود نویز می‌تواند تفاوت بسزایی در عملکرد این روش ایجاد کند. دو قضیه‌ی زیر نشان می‌دهند که کمینه کردن نرم ℓ_1 می‌تواند تحت شرایطی، جایگزین مناسبی برای این روش باشد.

قضیه ۲-۱ فرض کنید $\mathbf{A}_{m\times n}$ ماتریسی باشد که شرط *RIP* از مرتبه $2k$ را با ثابت $\delta_{2k} < \sqrt{2}-1$ ارضا کند؛ همچنین فرض کنید $\mathbf{x}_{n\times 1}$ بردار دلخواهی باشد و $\mathbf{y}_{m\times 1} = \mathbf{A}_{m\times n}\mathbf{x}_{n\times 1}$. اگر $\mathbf{x}^*_{n\times 1}$ بردار با حداقل نرم ℓ_1 باشد که $\mathbf{y}_{m\times 1} = \mathbf{A}_{m\times n}\mathbf{x}^*_{n\times 1}$ و $\mathbf{x}^{(k)}_{n\times 1}$، بهترین تقریب k-تنک بردار $\mathbf{x}_{n\times 1}$ باشد، داریم:

$$\|\mathbf{x}^* - \mathbf{x}\|_{\ell_2} \leq \frac{C_0}{\sqrt{k}}\|\mathbf{x} - \mathbf{x}^{(k)}\|_{\ell_1} \qquad (۳۲-۲)$$

که C_0 ضریبی ثابت (مستقل از $\mathbf{A}$ و $\mathbf{x}$) است [۲۴].

قضیه فوق نشان می‌دهد که اگر بردار $\mathbf{x}_{n\times 1}$ k-تنک باشد و $\mathbf{A}$ شرط RIP مورد نظر را ارضا کند، کمینه کردن نرم ℓ_1 (BP) این سیگنال را دقیقاً بازسازی خواهد کرد. در [۴۸] نشان داده شده است که نتیجه قضیه فوق برای محدوده وسیع تری از δ_{2k} نیز کماکان معتبر است.

قضیه ۲-۲ فرض کنید $\mathbf{A}_{m\times n}$ ماتریسی باشد که شرط *RIP* از مرتبه $2k$ را با ثابت $0 \leq \delta_{2k} < \sqrt{2}-1$ ارضا کند، همچنین فرض کنید $\mathbf{x}_{n\times 1}$ بردار دلخواهی باشد و $\mathbf{y}_{m\times 1} = \mathbf{A}_{m\times n}\mathbf{x}_{m\times 1} + \mathbf{n}_{m\times 1}$ که $\|\mathbf{n}_{m\times 1}\|_{\ell_2} \leq \epsilon$ (بردار نویز جمعی). اگر $\mathbf{x}^*_{n\times 1}$ بردار با حداقل نرم ℓ_1 باشد که $\|\mathbf{A}\mathbf{x}^* - \mathbf{y}\|_{\ell_2} \leq \epsilon$ و $\mathbf{x}^{(k)}_{n\times 1}$، بهترین تقریب k-تنک بردار $\mathbf{x}_{n\times 1}$ باشد، داریم:

$$\|\mathbf{x}^* - \mathbf{x}\|_{\ell_2} \leq \frac{C_0}{\sqrt{k}}\|\mathbf{x} - \mathbf{x}^{(k)}\|_{\ell_1} + C_1\epsilon \qquad (۳۳-۲)$$

که C_0, C_1 ضرایب ثابتی هستند [۲۴].

قضیه فوق نشان می‌دهد که روش کمینه کردن با نرم ℓ_1 نه تنها نسبت به نویز سیستم (k-تنک نبودن بردار) پایدار است، بلکه نسبت به نویز جمعی نیز پایداری خطی دارد. شرط استفاده شده در قضیه ۲-۲ ($\|\mathbf{A}\mathbf{x}^* - \mathbf{y}\|_{\ell_2} \leq \epsilon$) به شرط LASSO شهرت دارد [۸۸].

از آن‌جا که نرم ℓ_1 تابعی محدب است، کمینه کردن این نرم را می‌توان توسط روش‌های بهینه‌سازی محدب انجام داد. نکته مهم‌تر در مورد نرم ℓ_1 این است که مسأله کمینه کردن این نرم را می‌توان به صورت یک مسأله خطی بیان کرد. در نتیجه روش‌های برنامه نویسی خطی[۸] به طور موثری در پیدا کردن نقطه بهینه این مسأله می‌توانند مورد استفاده قرار گیرند. روش‌های معروف در این زمینه Simplex و Interior Point هستند که نقطه بهینه را با پیچیدگی محاسباتی از مرتبه n^3 (بعد بردار$\mathbf{x}$) پیدا می‌کنند.

در هر دو مسأله کمینه کردن نرم ℓ_1 (با و بدون نویز) با استفاده از ضرایب لاگرانژ می‌توان نشان داد که بردار حداقل کننده[۹] با بردار حداقل کننده

$$\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2_{\ell_2} + \lambda\|\mathbf{x}\|_{\ell_1} \qquad (۳۴-۲)$$

در صورت انتخاب صحیح λ برابر است. به عنوان مثال، در مسأله نویزی هنگامی‌که واریانس نویز σ_n^2 باشد، $\lambda = \sigma_n\sqrt{2\log n}$ گزینه نزدیکی به مقدار بهینه خواهد بود [۳۱] . با وجود آن که روش برنامه نویس خطی در زمان متناهی جواب مسأله را می‌یابد، در بسیاری از موارد کاربردی، پیچیدگی محاسباتی از مرتبه n^3 غیرعملی است. از این رو، روش‌های دیگری معرفی شده‌اند که به صورت تکراری به سمت جواب بهینه میل می‌کنند. با وجود این که این روش‌ها احتمالاً در زمان متناهی به جواب دقیق دست نمی‌یابند، اما خیلی سریع به جواب

[۸] Linear Programming

[۹] Minimizer

اصلی نزدیک می‌شوند و به کمک این روش‌ها می‌توان در زمانی بسیار کوتاه، به جواب نسبتاً دقیقی دست یافت. از روش‌های معروف در این راستا می‌توان به SPGL1 [۳۷] و GPSR [۴۶] اشاره کرد. در هر تکرار از این روش‌ها، حداقل کننده مسأله زیر محاسبه می‌شود :

$$\arg\min_{\mathbf{x}} \quad \|\mathbf{Ax}-\mathbf{b}\|_{\ell_2} \quad \text{subjectto } \|\mathbf{x}\|_{\ell_1} \leq \tau \tag{۳۵-۲}$$

که τ از روی λ محاسبه می‌شود و در هر تکرار مقدار λ بروز می‌شود.

در هنگامی‌که مقدار λ به نحوی از پیش معلوم باشد، روش‌های بسیار سریعی برای پیدا کردن حداقل کننده (۲-۳۴) وجود دارد. به عنوان مثال، در روش Reweighted LS در هر تکرار و بر حسب تقریب کنونی از جواب، جمله $\|\mathbf{x}\|_{\ell_1}$ با یک جمله مربعی جایگزین می‌شود و در نتیجه در هر تکرار یک مسأله حداقل مربعات حل می‌گردد [۳۵]. روش‌های ISTA و FISTA [۱۲، ۹۴] که اخیراً توجه زیادی را به خود جلب کرده‌اند، بر پایه تعویض جمله $\|\mathbf{Ax}-\mathbf{b}\|_{\ell_2}$ و جایگزاری یک جمله مناسب، مسأله کمینه‌سازی برداری در n بعد را به n مسأله کمینه‌سازی تک بعدی تقلیل می‌دهند. این روش‌ها نیز سرعت بالایی در همگرایی به جواب اصلی دارند .

۲-۵-۲ روش‌های حریص

روش‌های حریص[۱۰] بر پایه این اصل استوار است که در هر مرحله بدون آینده نگری، بهترین انتخاب ممکن صورت گیرد. همان طور که پیش بینی می‌شود، عدم وجود آینده نگری در این روش‌ها، در بسیاری از موارد منجر به شکست می‌شود اما در مورد بازسازی بردارهای تنک موفقیت قابل توجهی کسب کرده‌اند. برای توضیح این دسته از روش‌ها، از ساده‌ترین عضو این خانواده که Matching Pursuit است شروع می‌کنیم. فرض کنید $\mathbf{y}_{m\times 1} = \mathbf{A}_{m\times n}\mathbf{x}_{n\times 1}$ و $\mathbf{x}$ یک بردار k–تنک است. در این صورت $\mathbf{y}_{m\times 1}$ را می‌توان به صورت ترکیب خطی kتا از ستون‌های $\mathbf{A}_{m\times n}$ تلقی کرد و عمل بازسازی معادل با یافتن این k ستون و ضرایب آن‌ها در ترکیب خطی است. بر خلاف روش‌های مبتنی بر نرم ℓ_1 که یک تابع هزینه را کمینه می‌کنند، در این روش ابتدا ستون‌های مورد استفاده از $\mathbf{A}_{m\times n}$ در ترکیب خطی آشکار می‌شوند[۱۱] و سپس به کمک حل یک مسأله حداقل مربعات، ضرایب این ستون‌ها که همان مقادیر ناصفر بردار $\mathbf{x}_{n\times 1}$ هستند محاسبه می‌شوند. در هر مرحله از روش MP یکی از ستون‌های $\mathbf{A}_{m\times n}$ به عنوان عضوی فعالی در ترکیب خطی آشکار می‌شود؛ در این مرحله، ضرب داخلی $\mathbf{y}_{m\times 1}$ با

[۱۰] Greedy

[۱۱] Support Recovery

تمام ستون‌های $\mathbf{A}_{m\times n}$ محاسبه می‌شود و ستونی که بیشترین اندازه را حاصل کند (که بیشترین شباهت را به $\mathbf{y}_{m\times ۱}$ دارد) به عنوان عضو فعال شناسایی می‌شود و مقدار ضرب داخلی که به اندازه این ستون تقسیم شده است را به عنوان ضریب آن لحاظ می‌کنیم. اکنون تقریبی ۱–تنک از بردار اصلی بدست آورده‌ایم که در این مرحله انتخاب بهینه به شمار می‌رود ($\hat{\mathbf{x}}^{(۱)}_{n\times ۱}$) . برای ادامه، با فرض این که محاسبات تا این لحظه صحیح است، بردار $\mathbf{r}_{m\times ۱} = \mathbf{y}_{m\times ۱} - \mathbf{A}_{m\times n}\hat{\mathbf{x}}^{(۱)}_{n\times ۱}$ را به عنوان باقیمانده[۱۲] در نظر می‌گیریم و تمامی‌مراحل قبلی را طی می‌کنیم تا تخمین ۱–تنک جدیدی بدست آید. اکنون حاصل جمع این بردار ۱–تنک جدید و $\hat{\mathbf{x}}^{(۱)}_{n\times ۱}$ را به عنوان تقریب جدید $\hat{\mathbf{x}}^{(۲)}_{n\times ۱}$ تلقی می‌کنیم و این مراحل را مجدداً تا رسیدن به شرط پایانی (نرم باقیمانده کوچک یا تعداد مرحله معلوم و یا ترکیبی از هر دو) ادامه می‌دهیم. این روش از آن‌جا که در به روز رسانی تقریب‌های $\mathbf{x}$ از همبستگی ستون‌های $\mathbf{A}_{m\times n}$ صرف نظر شده است (ستون‌ها عمود فرض می‌شوند) ضعف‌های زیادی دارد. در روش OMP در مرحله‌ی i و پس از آشکار شدن i ستون فعال از ماتریس $\mathbf{A}$ به کمک یک مسأله حداقل مربعات، بهترین ضرایب را برای این بردارها مستقل از نتایج مراحل قبلی حساب می‌کنیم؛ به عبارت بهتر از نتایج قبلی تنها در یافتن مکان‌های ناصفر بردار $\mathbf{x}$ بهره می‌جوییم و نه بیشتر. از آنجا که روش‌های مذکور بر اساس شباهت بردار نمونه‌ها و باقیمانده به ستون‌های $\mathbf{A}$ پایه ریزی شده‌اند، ضریب همدوسی ماتریس $\mathbf{A}$ نقش مهم تری نسبت به پارامتر RIP آن ایفا می‌کند. همچنین روش‌های مذکور در برخی موارد نتایج ضعیفی در حضور نویز از خود نشان می‌دهند. از جمله روش‌های قوی در این خانواده می‌توان از CoSaMP نام برد؛ در این روش هر بار به جای انتخاب یک ستون از ماتریس $\mathbf{A}$ تعداد بیشتری انتخاب می‌شوند که پایداری نسبتاً خوبی را فراهم می‌کند، برای این روش تضمین مشابهی نسبت به کمیته کردن نرم $\ell_۱$ در مورد کیفیت بازسازی هنگام وجود نویز و یا k–تنک نبودن منبع اثبات شده است [۷۴]. روش‌های حریص به طور عام بسیار سریع‌تر از روش‌های دیگر هستند که این سرعت معمولاً به قیمت کیفیت تمام می‌شود.

۲–۵–۳ روش‌های آستانه‌ای

در روش‌های حریص، هر بار شدت ستون‌های ماتریس نسبت به بردار باقیمانده را حساب می‌کنیم و براساس یک قاعده، یک و یا تعدادی از ستون‌ها را به عنوان کاندیداهای ستون‌های فعال شناسایی می‌کنیم. نکته مهم این است که ستون‌های انتخابی در مرحله ثابت است و به روش بستگی دارد . دسته دیگری از روش‌ها وجود دارند

[۱۲] Residual

که مشابه روش‌های حریص به نوعی از شباهت بردارها سود می‌جویند ولی در هنگام انتخاب به جای معیار تعداد از یک سطح آستانه استفاده می‌کنند. یعنی در هر مرحله هر تعدادی از ستون‌ها که شباهت کافی به بردار مورد نظر را دارند انتخاب می‌شوند. در این صورت ممکن است در مرحله‌ای هیچ ستونی انتخاب نشود و یا تمامی‌ستون‌ها انتخاب می‌شوند .

از نماینده‌های معروف این دسته می‌توان به روش IHT[۱۳] اشاره کرد. در این روش برای حل

$$\arg\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_{\ell_2}^2 + \lambda\|\mathbf{x}\|_{\ell_0} \tag{۲-۳۶}$$

(ضرایب لاگرانژ برای نرم ℓ_0) هر بار از تکرارهای زیر استفاده می‌کنیم:

$$\mathbf{x}^{(i+1)} = \mathcal{H}\big(\mathbf{x}^{(i)} + \mathbf{A}^H(\mathbf{y}-\mathbf{A}\mathbf{x}^{(i)})\big) \tag{۲-۳۷}$$

که منظور از عملگر $\mathcal{H}$ مقایسه با سطح آستانه $\sqrt{\lambda}$ است:

$$\mathcal{H}(x_i) = \begin{cases} 0 & |x_i| \le \sqrt{\lambda} \\ x_i & |x_i| > \sqrt{\lambda} \end{cases} \tag{۲-۳۸}$$

در [۱۵] نشان داده شده است که روش فوق به سمت یکی از حداقل‌های موضعی مسأله (۲-۳۶) میل می‌کند.

در روش IHT هر بار مقایسه با یک سطح آستانه ثابت صورت می‌گیرد. روش IMAT [۷۰، ۸۶] تعمیمی‌از این روش است که سطح آستانه به طور وفقی بر اساس نتایج بدست آمده تا کنون تغییر می‌کند و در نتیجه سرعت همگرایی را افزایش می‌دهد. در [۹۷] نیز روشی مشابه بر مبنای الگوریتم CFAR برای آشکارسازی و حذف نویز ضربه‌ای ارائه شده است.

۲-۵-۴ روش‌های تقریب نرم

دسته کوچکی از روش‌ها مانند SLO[۱۴] [۷۳] براساس تقریب زدن نرم ℓ_0 با توابع دیگر شکل گرفته‌اند. در این روش‌ها در هر تکرار، به جای کمینه کردن نرم ℓ_0، تابع دیگری که احتمالاً یافتن حداقل کننده آن ساده‌تر است مورد استفاده قرار می‌گیرد. در طول تکرارها، توابع مورد استفاده به سمت نرم ℓ_0 نزدیک می‌شوند و در نتیجه جواب نهایی حداقل کننده نرم ℓ_0 خواهد بود. از آن‌جا که نرم ℓ_0 محدب نیست، توابع مورد استفاده نیز نمی‌توانند همگی محدب باشند و در نتیجه حداقل کردن آن‌ها نیز کار دشواری است . نکته اصلی در این روش‌ها این است

[۱۳] Iterative Hard Thresholding

[۱۴] Smoothed ℓ_0

که مقدار حداقل کننده تابع در مرحله کنونی به عنوان نقطه ابتدایی مرحله بعدی مورد استفاده قرار می‌گیرد. از این رو اگر تغییرات توابع به کندی صورت گیرد، هر بار نقطه ابتدایی و نقطه بهینه به اندازه کافی به یکدیگر نزدیک هستند و روش‌های مبتنی بر گرادیان برای یافتن حداقل کننده‌های این توابع غیرمحدب کفایت می‌کنند.