Sparse matrix optimization problems in computational color imaging

iPAL @ PENNSTATE
Information Processing and Algorithms Laboratory

Umamahesh Srinivas

iPAL Group Meeting

September 24, 2010

What is a sparse matrix?

- A matrix with significantly more number of 0's than non-zero elements
- In the words of J. H. Wilkinson, "... *any matrix with enough zeros that it pays to take advantage of them*"
- Structure lends itself to special techniques which improve computational efficiency: time and storage
- Examples: diagonal matrix (special case: identity matrix), band matrix
- Scenarios where sparse matrices appear: computational fluid dynamics (solution to partial differential equations), robotics, financial modeling, networks, computational color imaging

Examples from computational color imaging

1. Color look-up-table design via joint optimization of node locations and output values[1]

2. Image spatial resolution enhancement via multiple image captures[2]

---

[1] Monga and Bala, ICASSP 2010

[2] Monga and Srinivas, Invited paper, ASILOMAR Conference 2010

Brief digression: Optimization theory

- Consider an optimization problem in standard form:

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, m \\ & h_i(\mathbf{x}) = 0, i = 1, 2, \ldots, p. \end{array} \tag{1}$$

- Domain $\mathcal{D} = \bigcap_{i=0}^{m} \textbf{dom } f_i \cap \bigcap_{i=1}^{p} \textbf{dom } h_i \subset \mathbb{R}^n$.
- Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=0}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}). \tag{2}$$

Dual function

- Lagrange dual function $g : \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=0}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \right). \qquad (3)$$

- g is always concave in $\boldsymbol{\lambda}, \boldsymbol{\nu}$.
- A useful property:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^*, \text{ where } p^* = \inf_{\mathbf{x} \in \mathcal{D}} \{f_0(\mathbf{x})\}. \qquad (4)$$

- For convex optimization problems, solving Eq. (1) is equivalent to the following optimization problem:

$$\begin{array}{ll} \text{maximize} & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} & \boldsymbol{\lambda} \succeq \mathbf{0}. \end{array} \qquad (5)$$

Quadratic program (QP)

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + \mathbf{r} \\ \text{subject to} & \mathbf{G}\mathbf{x} \preceq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b}, \end{array} \tag{6}$$

where $\mathbf{P}$ is a $n \times n$ symmetric, positive semidefinite matrix, $\mathbf{G} \in \mathbb{R}^{m\times n}, \mathbf{A} \in \mathbb{R}^{p\times n}$.

- Objective function is (convex) quadratic
- Equality and inequality constraint functions are affine

Example 1: Color look-up-table design

Given: A training set of multi-dimensional color inputs and their "true" transformed values: $S = \{(\mathbf{x}_i, y_i)\}, i = 1, 2, \dots, K, \mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}$.

Choose look-up-table nodes $\chi^{nd} = \{\mathbf{x}_j^{nd}\}, j = 1, \dots, M$ and corresponding output values y_j^{nd} which minimize the following error function:

$$\mathcal{C}(\chi^{nd}, \mathbf{y}^{nd}) = \|\mathbf{y}^{tr} - \mathbf{W}_{\chi^{nd}}.\mathbf{y}^{nd}\|^2. \tag{7}$$

- Weight matrix $\mathbf{W}_{\chi^{nd}}$ induced by node locations χ^{nd}.
- For fixed $\mathbf{y}^{nd}$, cost function is convex in $\mathbf{W}_{\chi^{nd}}$.

Weight matrix optimization

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{y}^{tr} - \mathbf{W}_{\chi^{nd}}.\mathbf{y}^{nd^*}\|^2 \\ \text{subject to} \quad & \mathbf{w}_i^T \mathbf{e}_i = 0, i = 1, 2, \ldots, K \\ & w_{i,j} \geq 0, i = 1, 2, \ldots, T, j = 1, 2, \ldots, M \\ & \mathbf{W}_{\chi^{nd}}.\mathbf{1} = \mathbf{1}. \end{aligned} \quad (8)$$

Interpretation of constraints:

- $\mathbf{w}_i$ is the i-th row of $\mathbf{W}_{\chi^{nd}}$, $\mathbf{e}_i$ is a membership vector of zeros and ones. The membership constraint fixes the locations of the nodes with non-zero weights.
- $w_{i,j} \geq 0$: non-negativity of individual matrix elements, since interpolation weights are non-negative..
- Sum of elements in each row of $\mathbf{W}_{\chi^{nd}}$ is 1, since it is an interpolation matrix.

$\mathbf{W}_{\chi^{nd}}$ is a sparse matrix.

Formulation as a QP

$$\begin{aligned}\arg\min_{\mathbf{W}} \|\mathbf{y}^{tr} - \mathbf{W}\mathbf{y}^{nd}\|^2 &= \arg\min_{\mathbf{W}} \left((\mathbf{y}^{tr} - \mathbf{W}\mathbf{y}^{nd})^T(\mathbf{y}^{tr} - \mathbf{W}\mathbf{y}^{nd})\right) \\ &= \arg\min_{\mathbf{W}} \left((\mathbf{y}^{nd})^T\mathbf{W}^T\mathbf{W}\mathbf{y}^{nd} - 2(\mathbf{y}^{tr})^T\mathbf{W}(\mathbf{y}^{nd})\right)\end{aligned}$$

Let $\mathbf{Y} := \mathbf{y}^{nd}(\mathbf{y}^{nd})^T$ and $\mathbf{w}_i^T$ denote the i-th row of $\mathbf{W}$. Then,

$$\begin{aligned}(\mathbf{y}^{nd})^T\mathbf{W}^T\mathbf{W}\mathbf{y}^{nd} &= tr((\mathbf{y}^{nd})^T\mathbf{W}^T\mathbf{W}\mathbf{y}^{nd}) = tr(\mathbf{W}\mathbf{y}^{nd}(\mathbf{y}^{nd})^T\mathbf{W}^T) \\ &= tr(\mathbf{W}\mathbf{Y}\mathbf{W}^T) = \sum_{i=1}^{K} \mathbf{w}_i^T\mathbf{Y}\mathbf{w}_i \\ &= vec(\mathbf{W}^T)^T(\mathbf{I}_m \otimes \mathbf{Y})vec(\mathbf{W}^T) \\ &= vec(\mathbf{W}^T)^T\tilde{\mathbf{Y}}vec(\mathbf{W}^T),\end{aligned}$$

where $vec(\cdot)$ is the vectorizing operator, $\mathbf{I}_m$ is the $m \times m$ identity matrix, and $\otimes$ represents the Kronecker product.

Formulation as a QP (contd.)

Similarly,

$$\begin{aligned}(\mathbf{y}^{tr})^T\mathbf{W}\mathbf{y}^{nd} &= tr((\mathbf{y}^{tr})^T\mathbf{W}\mathbf{y}^{nd}) = tr(\mathbf{W}\mathbf{y}^{nd}(\mathbf{y}^{tr})^T) \\ &= tr((\mathbf{y}^{nd}(\mathbf{y}^{tr})^T)^T\mathbf{W}^T) = vec(\mathbf{y}^{nd}(\mathbf{y}^{tr})^T)^T\ vec(\mathbf{W}^T).\end{aligned}$$

With $\mathbf{z} := vec(\mathbf{W}^T)$ and $\mathbf{c} := -2vec(\mathbf{y}^{nd}(\mathbf{y}^{tr})^T)$, the original cost function becomes

$$\arg\min_{\mathbf{z}} \mathbf{z}^T\tilde{\mathbf{Y}}\mathbf{z} + \mathbf{c}^T\mathbf{z}. \tag{9}$$

- $\mathbf{Y}$ positive semidefinite, $\mathbf{I}_m$ positive definite $\Rightarrow$ $\tilde{\mathbf{Y}}$ is positive semidefinite (Kronecker product preserves positive definiteness).
- Cost function is quadratic in $\mathbf{z}$.

What about the constraints?

- Membership constraint: Let $\mathbf{0} \in \mathbb{R}^M$ denote the vector with all zeros. Define $\mathbf{E} \in \mathbb{R}^{K \times KM}$ such that

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1^T & \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{e}_2^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T & \mathbf{e}_K^T \end{bmatrix} \tag{10}$$

$$\mathbf{w}_i^T \mathbf{e}_i = 0, i = 1, 2, \ldots, K \Leftrightarrow \mathbf{Ez} = \mathbf{0} (\in \mathbb{R}^K). \tag{11}$$

- Non-negativity:

$$w_{i,j} \geq 0 \Leftrightarrow \mathbf{z} \succeq \mathbf{0} (\in \mathbb{R}^{KM}). \tag{12}$$

- Interpolation constraint: Let $\mathbf{1} \in \mathbb{R}^M$ denote the vector with all ones. Define $\mathbf{F} \in \mathbb{R}^{K \times KM}$ such that

$$\mathbf{F} = \begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{1}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T & \mathbf{1}^T \end{bmatrix} \tag{13}$$

$$\mathbf{W}.\mathbf{1} \ = \ \mathbf{1} \Leftrightarrow \mathbf{Fz} \ = \mathbf{1} (\in \mathbb{R}^K). \tag{14}$$

Complete optimization problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{z}^T\tilde{\mathbf{Y}}\mathbf{z} + \mathbf{c}^T\mathbf{z} \\ \text{subject to} \quad & \mathbf{E}\mathbf{z} = \mathbf{0} \\ & \mathbf{z} \succeq \mathbf{0} \\ & \mathbf{F}\mathbf{z} \ = \mathbf{1} \end{aligned} \tag{15}$$

where

- $\mathbf{z} = vec(\mathbf{W}^T)$
- $\tilde{\mathbf{Y}} = \mathbf{I}_m \otimes (\mathbf{y}^{nd}(\mathbf{y}^{nd})^T)$
- $\mathbf{c} = -2vec(\mathbf{y}^{nd}(\mathbf{y}^{tr})^T)$
- $\mathbf{E} \in \mathbb{R}^{K \times KM}$ is defined in Eq. (10)
- $\mathbf{F} \in \mathbb{R}^{K \times KM}$ is defined in Eq. (13)
- K is the number of training points.

Example 2: Spatial resolution enhancement

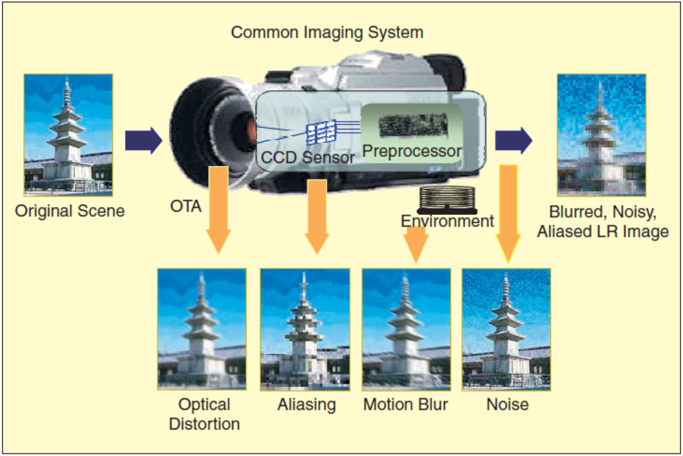

Figure: Digital image acquisition system[3].

[3] Park et al., IEEE Signal Processing Magazine, 2003

Model of the forward imaging process

$$\mathbf{y}_k = \mathbf{D}\mathbf{B}\mathbf{T}_k\mathbf{x} + \mathbf{n}_k, \quad 1 \le k \le K \tag{16}$$

where

- $\mathbf{x} \in \mathbb{R}^n$ is the unknown high-resolution image
- $\mathbf{y}_k \in \mathbb{R}^m$ $(m < n)$ represents the k-th low-resolution image
- $\mathbf{T}_k \in \mathbb{R}^{n \times n}$ is the k-th geometric warping matrix
- $\mathbf{B} \in \mathbb{R}^{n \times n}$ describes camera optical blur
- $\mathbf{D} \in \mathbb{R}^{m \times n}$ is a downsampling matrix of $1s$ and $0s$
- $\mathbf{n}_k \in \mathbb{R}^m$ is the noise vector that corrupts $\mathbf{y}_k$.

Final optimization problem

$$
\begin{array}{ll}
\text{minimize} & \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{D}\mathbf{B}\mathbf{T}_k\mathbf{x}\|_p + \lambda\rho(\mathbf{x}) \\
\text{subject to} & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \\
& \mathbf{0} \leq \mathbf{D}\mathbf{B}\mathbf{T}_k\mathbf{x} \leq \mathbf{1}, \quad 1 \leq k \leq K \\
& \mathbf{T}_k.\mathbf{1} = \mathbf{1}, \quad 1 \leq k \leq K \\
& \mathbf{B}.\mathbf{1} = \mathbf{1} \\
& \mathbf{t}_{k,i}^T\mathbf{m}_{k,i} = 0, \quad 1 \leq i \leq n, \quad 1 \leq k \leq K \\
& \mathbf{b}_i^T\mathbf{e}_i = 0, \quad 1 \leq i \leq n
\end{array}
\qquad (17)
$$

$\mathbf{D}, \mathbf{B}$ and $\mathbf{T}_k$ are sparse matrices.

Open problems

- Dual function analysis for the optimization in Example 1, i.e., solution to Eq. (5)
- Does there exist a closed-form solution to the optimization problems?
  - If not, can a closed-form solution be obtained by dropping any of the constraints?
  - In that case, will the resulting solution be physically meaningful?
- How can the sparse matrix optimization be performed more efficiently?