Basis Inverse

Ashutosh Mahajan

Department of Industrial and Systems Engineering
Lehigh University

COR@L Seminar Series, Spring 2005

References

Istvan Maros, Computational Techniques of the Simplex Method, Kluwer Academic Publishers, 2003. (Chapter 8)

THE ROAD TO WISDOM

The road to wisdom? – Well, it's plain
and simple to express:
Err
and err
and err again
but less
and less
and less.
–A Grook by Piet Hein.

The Basis Inverse

- $\hat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q$
- $\pi_B^T = \mathbf{c}_B^T B^{-1}$.

Storing Inverse

- Explicitly – What is the number of non-zeroes?
- What happens when we update?
- Product Form
- Elimination Form

Product Form of Inverse - PFI

- Start with a given B^{-1}, usually $\mathcal{I}$.
- Iteratively replace columns of B and update B^{-1}.
- Each update of B^{-1} is just a premultiplier matrix E_i
- $\hat{B}^{-1} = E_k E_{k-1} \dots E_1 \mathcal{I}$.
- Just need to store E_i.

What are the benefits?

Elimination Form of Inverse - EFI

- Do we really need B^{-1}?
- $\hat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q \Rightarrow B\hat{\mathbf{a}}_q = \mathbf{a}_q$
- $\pi_B^T = \mathbf{c}_B^T B^{-1} \Rightarrow B^T\pi_B = \mathbf{c}_B$
- $\hat{\mathbf{a}}_q$ and π_B^T are merely solutions of systems of linear equations.
- $B = LU \Rightarrow B^{-1} = U^{-1}L^{-1}$

What are the properties of $U^{-1}L^{-1}$? Can we store them easily? How easy are updates?

Elementary Transformation Matrix

- Suppose we have a basis B and its inverse B^{-1}.
- We wish to replace a column of B, $\mathbf{b}_p$ with $\mathbf{a}$.
- Let the new basis be B_a.
- Let $\mathbf{a} = \sum_{i=1}^{m} v_i \mathbf{b}_i$. Is $v_p = 0$?
- $\mathbf{b}_p = \frac{1}{v_p}\mathbf{a} - \sum_{i \neq p} \frac{v_i}{v_p}\mathbf{b}_i$.
- $\eta = \left[-\frac{v_1}{v_p}, \ldots, -\frac{v_{p-1}}{v_p}, \frac{1}{v_p}, -\frac{v_{p+1}}{v_p}, \ldots, -\frac{v_m}{v_p}\right]^T$.
- $E = \left[\mathbf{e}_1, \ldots, \mathbf{e}_{p-1}, \eta, \mathbf{e}_{p+1}, \ldots, \mathbf{e}_m\right]$.
- $B = B_a E$
- $B_a^{-1} = EB^{-1}$.

Updating columns of A

- $v_i = B^{-1}\mathbf{b}_i$.
- $\mathbf{a}_p = 0 \Rightarrow Ea = ?$
- What happens if B is lower triangular?
- What happens to E_i if we wish to have $\mathbf{e}_i$ in the basis?
- Fillin may happen in the updated columns
  - Difficult to foresee this beyond the next step.
  - *Markowitz merit number* $m_j^i = (r_i - 1)(c_j - 1)$.
- Numerical issues.

Triangularization in PFI

- If B can be made nearly triangular, PFI can become really fast.
- This usually differentiates a good implementation from not so good ones.

PFI operations

- $B^{-1} = E_k \dots E_1$.
- Thus, $\hat{\mathbf{a}}_j = E_k \dots E_1 \mathbf{a}_j$: FTRAN
- $\pi^T = \mathbf{c}_B^T E_k \dots E_1$: BTRAN
- Problems with BTRAN.
- Storing E_i
- When to reinvert?

Elimination form of Inverse

- Recall that if $B = LU$, then $B^{-1} = U^{-1}L^{-1}$.
- Thus, $B^{-1} = U^1 \dots U^m L^m \dots L^1$. A product form!
- What happens on premultiplying by E?
- We need hybrid procedures.

How to maintain sparsity?

- Method of Suhl and Suhl.
- Rearrange (not physically) rows and columns to get a triangular structure.
- Inversion of L and U is easy.
- Inverting kernel is important – fillin and numerical stability.
- Implementation can again make a difference.

How to triangularize?

- $\bar{B} = B + (\mathbf{a}_q - B\mathbf{e}_p)\mathbf{e}_p^T$.
- or $L^{-1}\bar{B} = U + (L^{-1}\mathbf{a}_q - U\mathbf{e}_p)\mathbf{e}_p^T$.
- So we may get a *spike* in U.
- Permute rows and columns to get upper Hessenberg matrix.
- Apply some E's to restore triangularity.
- $(L^k)^{-1}B^k = R^k U^k (R^k)^T$.
- Thus, we maintain L^{-1}, and U for each required basis.