Calculating Approximate GCD of Multiple Univariate Polynomials using Approximate Syzygies

Akira Terui
Faculty of Pure and Applied Sciences, University of Tsukuba
Tsukuba, 305-8571, Japan
terui@math.tsukuba.ac.jp, http://researchmap.jp/aterui/

For given n univariate polynomials with $n \geq 3$, we present a Symbolic-Numeric method for calculating approximate greatest common divisor (GCD) of them by calculating approximate Syzygies. This kind of GCD calculation can be used in application such as blind image deconvolution [1]. In such a case, it is especially effective when we try to restore the original image from several number of blurred images.

In our previous research, we have developed a method for calculating approximate GCD, called *GPGCD* [4]. Furthermore, we have extended the original method for n polynomial inputs [3] based on Rupprecht's first algorithm [2, Sect. 4]. However, this method is inefficient for large number and/or degree of input polynomials because, in such cases, the dimension of a generalized Sylvester matrix becomes large and sparse. While Rupprecht's second algorithm [2, Sect. 5] seems more efficient by using Syzygies with another generalization of Sylvester matrix whose dimension is much smaller than those used in the first algorithm, as for our GPGCD method, we have difficulty applying the method directly (we will explain the reason in detail below). We present a method to overcome the difficulty.

For $i = 1, \ldots, n$, let $P_i(x)$ be real univariate polynomial of degree $d_1 \leq \cdots \leq d_n$, respectively, with $d_1 > 0$, given as $P_i(x) = p_{d_i}^{(i)} x^{d_i} + \cdots p_1^{(i)} x + p_0^{(i)}$. At first we assume that $P_1, \ldots, P_n$ have a GCD. Let $H = \gcd(P_1, \ldots, P_n)$ and $d = \deg(H)$ with $d \leq d_1$.

For a real univariate polynomial $P(x)$ represented as $P(x) = p_n x^n + \cdots + p_0 x^0$, let $C_k(P)$ be a real $(n + k, k + 1)$ matrix (called "convolution matrix") defined as $C_k(P) = \big({}^t(p_n, \ldots, p_0, 0, \ldots, 0), {}^t(0, p_n, \ldots, p_0, 0, \ldots, 0), \ldots, {}^t(0, \ldots, 0, p_n, \ldots, p_0)\big)$ and let $\boldsymbol{p}$ be the coefficient vector of $P(x)$ defined as $\boldsymbol{p} = (p_n, \ldots, p_0)$, and vice versa.

As a generalized Sylvester matrix, we use the second definition by Rupprecht [2, Sect. 5]. For $k > d_1$, define the k-th Sylvester matrix of $P_1, \ldots, P_n$ as $N_k(P_1, \ldots, P_n) = \begin{pmatrix} C_{k-d_1}(P_1) & C_{k-d_2}(P_2) & \cdots & C_{k-d_n}(P_n) \end{pmatrix}$, where $C_{k-d_i}(P_i)$ has empty element for $k < d_i$.

If a vector $\boldsymbol{v} = {}^t \begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \cdots & \boldsymbol{r}_n \end{pmatrix}$ with $\dim(\boldsymbol{r}_i) = k - d_i + 1$ satisfies $N_k \boldsymbol{v} = \boldsymbol{0}$, then we see that the polynomials $R_1, \ldots, R_n$ whose coefficient vectors are $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n$, respectively, satisfy $R_1 P_1 + \cdots + R_n P_n = 0$. In such a case, we call a tuple of polynomials $(R_1, \ldots, R_n)$ a Syzygy of $P_1, \ldots, P_n$ of degree k.

In Rupprecht's second method, we first calculate Syzygies of $P_1, \ldots, P_n$, then calculate cofactors of $P_1, \ldots, P_n$ by using calculated Syzygies, as follows.

1. Calculate $n - 1$ "independent" (as elements in a module over polynomial ring $\mathrm{R}[x]$) Syzygies which satisfy the following condition on the degrees. For $j = 1, \ldots, n - 1$, let $R_j = (U_1^{(j)}, U_2^{(j)}, \ldots, U_n^{(j)})$ be a Syzygy of $P_1, \ldots, P_n$ of degree r_j. Then, we have

$$d = d_1 + \cdots + d_n - (r_1 + \cdots + r_{n-1}) \tag{1}$$

   [2, Lemma 5.3]. With numerical computation on coefficients, we calculate a Syzygy by the Singular Value Decomposition (SVD) on Sylvester matrix N_k by increasing the degree k by 1 from the initial value d_1, until we obtain $n - 1$ Syzygies satisfying condition (1).

2. For calculated Syzygies $R_j = (U_1^{(j)}, U_2^{(j)}, \ldots, U_n^{(j)})$, $j = 1, \ldots, n-1$, define a matrix $U = (u_{ij})$ as $u_{ij} = U_j^{(i)}$, and let Δ_i be the minor of U by deleting the i-th column (note that we must define U satisfying that $\Delta_i \neq 0$ for all i). Then, Δ_i is the cofactor of P_i satisfying $P_i = H \cdot \Delta_i$ [2, Lemma 5.2]. Thus, by calculating U and Δ_i, we obtain desired GCD H.

In our GPGCD method, we accept polynomials $P_1, \ldots, P_n$ that are pairwise relatively prime in general, then find "perturbation terms" ΔP_i, $i = 1, \ldots, n$, satisfying that "perturbed polynomials" $\tilde{P}_i = P_i + \Delta P_i$ have a nontrivial GCD H. With the original method by Rupprecht, we may encounter the following issue in Step 1. In our GPGCD method, we set up constrained optimization problem with constraints on the coefficients in input polynomials and their Syzygies that need coefficients in *all* input polynomials at once. On the other hand, Step 1 in the above may not involve *all* input polynomials from the beginning step(s), thus it is not clear if we can calculate appropriate perturbed terms incrementally to make all the perturbed polynomials satisfy the Syzygy relations in the final phase. Therefore, we modify the method so that we use Syzygy relations that involve *all* input polynomials from the beginning step, as follows.

1. Let $l = d_n$. For given polynomials $P_1, \ldots, P_n$, calculate perturbed polynomials $\tilde{P}_1, \ldots, \tilde{P}_n$ along with Syzygies $R_j = (U_1^{(j)}, U_2^{(j)}, \ldots, U_n^{(j)})$ of degree l satisfying $U_1^{(j)}\tilde{P}_1 + \cdots + U_n^{(j)}\tilde{P}_n = 0$, as follows. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ be the right singular vectors of $N_l(P_1, \ldots, P_n)$ calculated with the SVD. By optimization method (in our case we use so-called the modified Newton method; see our literature [4] for reference), we obtain $\tilde{P}_1, \ldots, \tilde{P}_n$ by perturbing coefficients in $P_1, \ldots, P_n$, and $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_m$ by perturbing $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$, respectively, satisfying $N_l(\tilde{P}_1, \ldots, \tilde{P}_n)\tilde{\boldsymbol{v}}_j = \boldsymbol{0}$. From vector $\tilde{\boldsymbol{v}}_j = \begin{pmatrix} \boldsymbol{r}_1^{(j)} & \boldsymbol{r}_2^{(j)} & \cdots & \boldsymbol{r}_n^{(j)} \end{pmatrix}$, we extract coefficients of a Syzygy $R_j = (U_1^{(j)}, U_2^{(j)}, \ldots, U_n^{(j)})$.

2. Using Syzygies R_j calculated in the above step, select and/or calculate Syzygies of appropriate degree satisfying (1) to make up matrix U with the following strategies.

   (a) If we need to calculate Syzygies of degree k smaller than l, make appropriate linear combination of the right singular vectors $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_m$ to eliminate coefficients of degrees greater than k in the corresponding Syzygy, as follows. Let M be a submatrix of $\begin{pmatrix} \tilde{\boldsymbol{v}}_1 & \cdots & \tilde{\boldsymbol{v}}_m \end{pmatrix}$ consisting of the rows corresponding the coefficients of $U_i^{(j)}$ of degree greater than k. Then, calculate the SVD on M to find basis of the null space of M. Repeat this step until we find appropriate Syzygies satisfying (1) along with $\Delta_i \neq 0$ for all i.

   (b) If we could not find $n-1$ independent Syzygies satisfying the degree condition (1) along with $\Delta_i \neq 0$ for degree of Syzygies smaller than or equal to l, then, for degree k greater than l, calculate new Syzygies from $N_k(\tilde{P}_1, \ldots, \tilde{P}_n)$ until we find $n-1$ independent Syzygies satisfying (1) along with $\Delta_i \neq 0$.

References

[1] Z. Li, Z. Yang, and L. Zhi. Blind image deconvolution via fast approximate GCD. In *Proceedings of ISSAC '10*, pages 155–162, 2010.

[2] D. Rupprecht. An algorithm for computing certified approximate GCD of n univariate polynomials. *J. Pure Appl. Algebra*, 139:255–284, 1999.

[3] A. Terui. GPGCD, an iterative method for calculating approximate GCD, for multiple univariate polynomials. In *Lecture Notes in Computer Science* 6244, pages 238–249. Springer, 2010.

[4] A. Terui. GPGCD: An iterative method for calculating approximate GCD of univariate polynomials. *Theoretical Computer Science*, 479:127–149, 2013. Symbolic-Numerical Algorithms.