Grad. Sch. Neur. Info. Proces. • 1-9
Lab Rotation Version

Graduate School of Neural Information Processing

Projection Methods for Large Linear Systems Without the Agonizing Pain

Review

Leonardo Casarsa[1]*

1 Probabilistic Inference Group, Max Planck Institute for Intelligent Systems, Spemannstr. 38, 72076, Tübingen, Germany

Abstract: The solution to large linear systems is computationally expensive to obtain through traditional methods, but can be iteratively estimated using projection methods. Projection methods, among which the conjugate gradient algorithm is one of the most popular, are usually presented in an algorithmic way which obscures the mathematical intuition behind the operations. The goal of this review is to introduce projection methods as a logical step from a few restrictions. We give particular focus on the interpretation of projection operators as optimizers for some problems and show how different interpretations of the same problem lead to different methods. As a working example, we derive the conjugate gradient method and provide a clear meaning for each term in its algorithm.

Keywords: linear systems • projection methods • conjugate gradient

1. Introduction

We want to solve the linear system

$$Ax = b \tag{1}$$

where is A is a large $n \times n$ symmetric positive definite matrix, $x, b \in \mathbb{R}^n$.

Classical solvers, such as Gaussian elimination, usually store all n^2 entries of A and can have $O(n^3)$ computational complexity. Moreover their only output is the exact solution $A^{-1}b$ after all the steps are over. In particular they do not fully explore special structures of the matrix A, such as sparseness. In contrast, matrix-vector products for sparse matrices require only $O(n)$ operations, and this can be used to produce more efficient algorithms in terms of work and storage. Additionally, we may be interested only in an approximated solution for (1). In such case, it would not be beneficial to wait for the exact solution.

* *E-mail: leonardo.azevedo@tuebingen.mpg.de*

Therefore we want solvers which

I - produce an estimator for the solution at each step

II - contain only matrix-vector or vector-vector multiplications

Most of the practical iterative solvers which satisfy the previous conditions extract an approximate solution at each step from an expanding subspace. These solvers are called Projection Methods, of which one of the most popular is the *conjugate gradient*. The goal of this review is to introduce a general framework for constructing projection methods and concludes with a derivation of the conjugate gradient as a special example. We provide an intuitive rather then rigorous introduction to the problem.

2. Projection Methods

An iterative estimator to (1) can be obtained by approximating $x = A^{-1}b$ from a search subspace $\mathcal{K} \subset \mathbb{R}^n$. In each step the search subspace must expand, such that $\mathcal{K}_m \subset \mathcal{K}_{m+1} \subset \mathrm{Im}(A^{-1})$. The updates for our estimator can then be written as a linear combination of the basis vectors $\{v_1, v_2, \ldots, v_m\} \in \mathbb{R}^n$ of $\mathcal{K}_m$,

$$\hat{x}_m - x_0 = V_m y_m \tag{2}$$

for some $y_m \in \mathbb{R}^m$, where $x_0 \in \mathbb{R}^n$ is an initial guess. Our goal is to find an $\hat{x}_m \in \mathcal{K}_m$ that minimizes the error $e(\hat{x}_m) := x - \hat{x}_m$, or the residual $r(\hat{x}_m) := b - A\hat{x}_m$ in the subspace. This property is fulfilled by projection operators, which will be better introduced in the next section and will guide the derivation of the methods in this review.

2.1. Projection Operators

A projection operator $M \in \mathcal{M}_{n\times n}$ is defined as an idempotent linear mapping from $\mathbb{R}^n$ to itself,

$$M^2 = M$$

It follows that $I - M$ is also an idempotent linear mapping and defines the complementary projection operator of M,

$$(I - M)^2 = I - M$$

$$\mathrm{Ker}(M) = \mathrm{Im}(I - M)$$

Therefore every $x \in \mathbb{R}^n$ can be uniquely decomposed as $x = Mx + (I - M)x$, with

$$Mx \in \mathrm{Im}(M)$$

$$(I - M)x \in \mathrm{Ker}(M)$$

In fact, an arbitrary projector M is uniquely identified by the two complementary subspaces: $\mathrm{Im}(M)$ and $\mathrm{Ker}(M)$. Equivalently M can be determined by $\mathcal{L}_m, \mathcal{K}_m$, with $\mathcal{K}_m := \mathrm{Im}(M)$ and $\mathcal{L}_m := \mathrm{Ker}^{\perp}(M)$, the orthogonal complement of the null space of M, such that

$$Mx \in \mathcal{K}_m \tag{3}$$

$$(I - M)x \perp \mathcal{L} \tag{4}$$

Conditions (3) and (4) provide a simpler intuition on what constitutes a projection mapping. If we want to project $x \in \mathbb{R}^n$ onto an m-dimensional subspace $\mathcal{K}$, there are m degrees of freedom to be determined. We must impose m additional restrictions, which can be expressed as an orthogonality condition between the complementary component $(I - M)x$ and an m-dimensional *subspace of constraints* $\mathcal{L}$.

Orthogonal projection mapping occur whenever $\mathcal{L} = \mathcal{K}$. Consequently $Mx \perp (I - M)x$. In a similar fashion we can define an A-*orthogonal* projection operator, given that A is a symmetric positive-definite matrix, whenever $(I - M)x$ and $\mathcal{L}_m$ are A-orthogonal. Two vectors v, w are A-orthogonal if

$$\langle v, w \rangle_A := v^{\top} A w = 0 \tag{5}$$

Orthogonal projection operators play an important role in optimization under constrained subspaces.

Theorem 2.1.
The distance between two vectors $x \in \mathbb{R}^n$ and $y \in \mathcal{K} \subset \mathbb{R}^n$ is minimized by the orthogonal projection mapping M of x onto $\mathcal{K}$

$$\arg\min_{y \in \mathcal{K}} \|x - y\|_2 = Mx$$

Proof. The idea is to decompose $(x - y)$ into orthogonal components. Since $\langle x - Mx, Mx - y \rangle = 0$, we have

$$\begin{aligned} \|x - y\|^2 &= \|(x - Mx) + (Mx - y)\|^2 \\ &= \|x - Mx\|^2 + \|Mx - y\|^2 \end{aligned}$$

Now $\|x - Mx\|^2$ does not depend on y, so

$$\arg\min_{y \in \mathcal{K}} \|x - y\|_2 = \arg\min_{y \in \mathcal{K}} \|Mx - y\|^2 = Mx$$

□

A similar optimization property follows from the last theorem for A-orthogonal projections, by just changing the Euclidean norm to the norm induced by the A-inner product (5)

Corollary 2.1.
If A is a symmetric positive-definite matrix, the A-distance between two vectors $x \in \mathbb{R}^n$ and $y \in \mathcal{K} \subset \mathbb{R}^n$ is minimized by the A-orthogonal projection mapping M^{A} of x onto $\mathcal{K}$

$$\arg\min_{y \in \mathcal{K}} \|x - y\|_A = M^{\mathrm{A}} x$$

Now we have all the elements we need to write a general framework for projection methods.

2.2. General Framework

A projection method constructs an estimator $\hat{x}_m$ to (1) by constraining the updates from an initial guess x_0 to $\mathcal{K}_m$, and the residual vector $r_m := b - Ax_m$ to a subspace perpendicular to $A\mathcal{L}_m$

$$\hat{x}_m - x_0 \in \mathcal{K}_m \quad \text{and} \quad b - Ax_m \perp A\mathcal{L}_m$$

Let V_m, W_m be two bases for $\mathcal{K}_m$ and $A\mathcal{L}_m$ respectively, then

$$\begin{cases} \hat{x}_m - x_0 = V_m y_m \\ W^\top (b - Ax_m) = 0 \end{cases}$$

Substituting the first on the second equation,

$$\begin{aligned} W_m^\top (b - A\hat{x}_m) = 0 \;&\Rightarrow\; W_m^\top ((b - Ax_0) + A(x_0 - \hat{x}_m)) = 0 \\ &\Rightarrow\; W_m^\top (r_0 + AV_m y_m) = 0 \\ &\Rightarrow\; y_m = (W_m^\top A V_m)^{-1} W_m^\top r_0 \end{aligned}$$

y_m will be uniquely identified whenever $W_m^\top A V_m$ is invertible. This is true for all cases in this review. It follows that

$$\hat{x}_m = x_0 + V_m (W_m^\top A V_m)^{-1} W_m^\top r_0$$

This is the general update rule for a projection method. It is easier to inteprete it by noticing that $r_0 = A(x - x_0)$ so that the previous equation becomes

$$(\hat{x}_m - x_0) = M_m (x - x_0), \quad \textit{with } M_m = V_m (W_m^\top A V_m)^{-1} W_m^\top A \tag{6}$$

So the updates are projections of the error onto $\mathcal{K}_m$ orthogonal to $A\mathcal{L}_m$.

An additional interpretation is possible by seeing that $A(\hat{x}_m - x_0) = r_0 - r_m$, therefore

$$r_m = (I - N_m)(r_0), \quad with\ N_m = AV_m(W_m^\top AV_m)^{-1}W_m^\top \tag{7}$$

So the residual at step m is the projection of the initial residual onto a shrinking subspace $\mathbb{R}^n \setminus A\mathcal{K}_m$ orthogonal to $\mathbb{R}^n \setminus \mathcal{L}_m$. It is easy to verify that both M_m and $I - N_m$ are projection mappings, since they are idempotent linear mappings.

Different ways of picking $\mathcal{L}_m$ and $\mathcal{K}_m$ will lead to distinct methods. We will focus on methods that minimize either the error or the residual at each step. Other optimization algorithms are possible, but they are not guaranteed to converge to an exact solution for every case, so we will not discuss them further in this review [1].

2.3. Search subspaces $\mathcal{K}_m$

The first step to the solver is picking an appropriate set of search subspaces. Following restriction (II), a natural choice for $\mathcal{K}_m$ is some *Krylov subspace*, such that $\mathcal{K}_m(A, b) := span\{b, Ab, \dots, A^{m-1}b\}$. For m large enough, $\mathcal{K}_m(A, b)$ spans the exact solution $A^{-1}b$, since by the Cayley-Hamilton Theorem

$$A^{-1} = c_0 + c_1 A + \cdots + c_{n-1}A^{n-1}, \quad \text{with } c_i \in \mathbb{R}$$

Therefore $A^{-1}b \in \mathcal{K}_n(A, b)$. In fact, the exact solution can be achieved in only k steps, where k is the number of distinct eigenvalues of A. [1]

Lemma 2.1.
If λ_j, with $j = 1, \dots, k$ are the distinct eigenvalues of A, then $x = A^{-1}b \in \mathcal{K}_k(A, b)$.

It is therefore a good strategy to constrain $\hat{x}_m \in \mathcal{K}_m$. From now on, whenever we refer to $\mathcal{K}_m$, we mean $\mathcal{K}_m(A, b)$.

One disadvantage of the Krylov subspace is that $\{b, Ab, A^2b, \dots\}$ become less linearly independent at each step. A general way to address this is to apply Gram-Schmidt orthogonalization to generate an orthonormal basis for $\mathcal{K}_m$. This is called the *Arnoldi process*. In case A is symmetric, the algorithm is called *Lanczos process*. Both Arnoldi and Lanczos processes can be used to solve linear systems, but a direct implementation of them does not lead to practical estimators. Instead better algorithms such as the generalized minimal residual (GMRES)[2] and the conjugate gradient can be derived from them. In this review we will not derive conjugate gradient from Lanczos, but as a general projection method with special subspace of constraints.

2.4. Subspaces of constraints $\mathcal{L}_m$

$\mathcal{L}_m$ is bounded by similar restrictions as $\mathcal{K}_m$ (II). Therefore it is reasonable to try $\mathcal{L}_m = \mathcal{K}_m$. As seen before (6), this amounts to projecting $(x - x_0)$ A-orthogonally onto $\mathcal{K}_m$ and, by Theorem 2.1 to minimizing the A-norm of the error. Alternatively, we can pick $\mathcal{L}_m = A\mathcal{K}_m$, and equation (7) is the orthogonal projection of $r(x_0)$ onto $\mathbb{R}^n \setminus A\mathcal{K}_m$, therefore minimizing the residual in that subspace.

Different choices for $\mathcal{L}_m$ induce distinct optimization conditions which lead to different solvers. We will see next that some of these solvers can be derived from different interpretations of the linear problem (1).

2.4.1. Least Squares

A generalized version of equation (1) is the linear least squares problem for finding

$$\hat{x} = \arg\min_{x \in \mathbb{R}^n} \|Ax - b\| \tag{8}$$

Whenever A is full-rank, the solution for (1) and (8) will clearly be the same. Otherwise, there would be no solution for (1), but (8) would still deliver a good approximation in the least squares sense.

Restricted to $\mathcal{K}_m$, the least squares estimator can be written as,

$$\text{Find } \hat{x}_m \text{ such that } A\hat{x}_m = \arg\min_{Ax \in A\mathcal{K}_m} \|Ax - b\|$$

This is equivalent to picking $\mathcal{L}_m = A\mathcal{K}_m$. So we can write the residual projection equation (7) as

$$r_m = (I - AV_m(V_m^\top A^\top A V_m)^{-1} V_m^\top A^\top) r_0 \tag{9}$$

$V_m^\top A^\top A V_m$ is nonsingular whenever A is nonsingular.

Examples of least squares algorithms are the GMRES[2], LSQR[4] and MINRES[3]. GMRES and LSQR are very general iterative solvers, since they are able to provide an estimator to non-symmetric indefinite non-square matrices. However they can become computationally expensive, as there is no way to implement them as a short recurrence[5]. MINRES for symmetric positive-definite problems is equivalent to the CG algorithm and should be the preferred choice to solve problems with symmetric indefinite matrices.

2.4.2. Quadratic optimization

If A is symmetric and positive definite, we can interpret the linear problem (1) as the minimization of a quadratic function $\phi(x) := \frac{1}{2} x^\top A x - b^\top x$. The next theorem guarantees that minimizing $\phi(x)$ in any subspace of $\mathbb{R}^n$ is equivalent to minimizing the A-norm of the error in that same subspace.

Theorem 2.2.

$$\arg\min_{y \in \mathcal{K}_m} \|x - y\|_A = \arg\min_{y \in \mathcal{K}_m} \phi(y)$$

Proof.

$$\|x-y\|_A^2 = (x-y)^\top A(x-y) = x^\top Ax + y^\top Ay - 2y^\top Ax = 2\phi(y) + x^\top Ax$$

□

This is equivalent to picking $\mathcal{L}_m = \mathcal{K}_m$. So we can write the error projection equation (6) as

$$\hat{x}_m - x_0 = V_m(V_m^\top A V_m)^{-1} V_m^\top A e_0 \tag{10}$$

where $e_0 := e(x_0)$.

Since A is a symmetric and positive-definite matrix, $V_m^\top A V_m$ is as well. The previous equation characterizes the update for the conjugate gradient.

Conjugate Gradient

Conjugate gradient follows

$$\text{Find } \hat{x}_m \in \mathcal{K}_m \text{ such that } r_m \perp A\mathcal{K}_m \tag{11}$$

with update equation (10). The most computationally expensive step is finding the inverse of matrix $G := V_m^\top A V_m$. This issue is addressed by picking a different basis P_m for $\mathcal{K}_m$, such that that G, and therefore G^{-1}, are diagonal.

$$G^{-1} = \begin{bmatrix} \frac{1}{\delta_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\delta_2} & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\delta_m} \end{bmatrix}, \qquad \text{with} \quad \delta_i = p_i A p_i$$

We can write

$$\hat{x}_m = x_0 + y_{m,1}p_1 + y_{m,2}p_2 + \cdots + y_{m,m}p_m \qquad \text{with } y_{m,i} = \frac{p_i^\top A e_0}{\delta_i}$$

Since the directions are conjugate,

$$\begin{aligned} p_k^\top A e_0 &= p_k^\top A(x - x_0) = p_k^\top A(x - x_k + x_k - \cdots + x_1 - x_0) \\ &= p_k^\top A e_m \end{aligned}$$

So the update equation is

$$\hat{x}_m = \hat{x}_{m-1} + \alpha_m p_m, \qquad \text{with } \alpha_m := \frac{p_m^\top r_m}{p_m^\top A p_m} \tag{12}$$

The conjugate gradient method picks a basis of A-conjugate search directions for $\mathcal{K}_m$. Therefore estimator updates will only occur once in each direction. In particular, the updates will be an A-orthogonal projection of e_m onto p_m.

From the previous equation we can determine the update formula for the residual,

$$\begin{aligned} r_m &= r_{m-1} - A(\hat{x}_{m-1} - \hat{x}_m) \\ &= r_{m-1} - \alpha_m A p_m \end{aligned} \tag{13}$$

Finally, we construct the set of search directions with the conjugate Gram-Schmidt process.

Given an arbitrary set of vectors $\{d_1, \ldots, d_m\}$ which span $\mathcal{K}_m$, p_m is obtained by eliminating from d_m all the components which are not A-orthogonal to the previous directions. This is an A-orthogonal projection of d_m onto $\mathbb{R}^n \setminus span\{p_1, ..., p_{m-1}\}$. Starting with $p_1 = d_1$,

$$p_m = (I - N_m)d_m \quad \text{with } N_m = P_m(P_m^\top A P_m)^{-1} P_m^\top A$$

Again $(P_m^\top A P_m)^{-1}$is a diagonal matrix, and that leads to simpler computations. A further simplification can nonetheless be achieved by choosing $\{d_1, \ldots, d_m\}$, such that d_m is A-orthogonal to all previous search directions. Condition (11) guarantees that for $d_m = r_m$. So we can write the previous equation as

$$p_{m+1} = r_m - \beta_m p_m, \qquad \text{with } \beta_m := \frac{p_m^\top A r_m}{p_m^\top A p_m} \tag{14}$$

Collecting the updates for the error (12), residual (13) and direction (14), we can write the conjugate gradient algorithm as

Algorithm 1 Conjugate Gradient

INPUT: $x_0 \in \mathbb{R}^n$

$r_0 \leftarrow Ax_0 - b$, $p_1 \leftarrow r_0$
for $m = 1, \ldots$ **do**
  until stopping condition
  $\alpha_m \leftarrow \frac{p_m^\top r_m}{p_m^\top A p_m}$
  $x_m \leftarrow x_{m-1} + \alpha_m p_m$
  $r_m \leftarrow r_{m-1} - \alpha_m A p_m$
  $\beta_m \leftarrow \frac{p_m^\top A r_m}{p_m^\top A p_m}$
  $p_{m+1} \leftarrow r_m - \beta_m p_m$

Further improvements can be done to this algorithm but they are not in the scope of this review. More interesting is to provide an interpretation for $\alpha_m p_m$ and $\beta_m p_m$ in terms of projections:

$$\begin{aligned} \alpha_m p_m &= \frac{p_m p_m^\top A}{p_m^\top A p_m} e_m \\ \beta_m p_m &= \frac{p_m p_m^\top A}{p_m^\top A p_m} r_m \end{aligned}$$

Therefore both of the previous terms are A-orthogonal projections of the error and residual (respectively) onto the current search direction.

3. Conclusion

The main advantage of deriving the conjugate gradient through projection methods is a better understanding of what each update means. Although there are many essays deriving the conjugate gradient, few [6] present more than an algorithmic point of view. A general understanding of projection methods allows an adaptation of the existing methods for specific needs. For example, one could choose to design an algorithm analagous to the CG but in nested subspaces other than Krylov. This could be done with little effort by just choosing the proper projection operators. Another example are algorithms for solving different optimizations in every subspace, like a weighted or regularized least squares. Although the solution to these problems cannot be always written as projections, since they have a known closed form, the methods could be easily adapted. A final possible application for this review is in the further development of Probabilistic Numerics, a framework which applies gaussian inference to infer the uncertainty over the result, given a prior uncertainty. Since linear mappings of Gaussians are still Gaussians, the author believe the constructing general probabilistic projection methods should be relatively straightforward.

Acknowledgements

The author would like to thank his lab rotation supervisor Philipp Hennig for all support during the last month. He is also grateful to fellow lab members Edgar Klenske, Michael Schober and Maren Mahsereci for their patience when answering questions, as well as their tolerance to change. Thanks, guys! It has been lots of fun.

References

[1] Saad Y., ”Iterative Methods for Sparse Linear Systems”, 2nd Edition, SIAM, 2003

[2] Saad Y. & Schultz M.H., ”GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”, SIAM J. Sci. Stat. Comput., 7:856-869, 1986.

[3] Paige C.C. & Saunders M.A., ”Solution of sparse indefinite systems of linear equations”, SINUM 12, 1975, 617–629

[4] Paige C.C. & Saunders M.A., ”LSQR: An algorithm for sparse linear equations and sparse least squares”, TOMS 8(1), 1982, 43-71.

[5] Faber V., Liesen J. and Tichý P., ”The Faber–Manteuffel Theorem for Linear Operators”, J. on Num. Analysis, SIAM, 2008, 46:3, 1323-1337

[6] Gower R.M., ”Conjugate Gradients: The short and painful explanation with oblique projections”, *not published*, http://www.maths.ed.ac.uk/~s1065527/pdf/GowerR_Painful_PCG_projections, 2014

[7] Hennig P., ”Probabilistic Interpretation of Linear Solvers”, SIAM J. on Optimization, 25:235-260, 2015