Modern Optimization Methods for Big Data Problems

MATH11146
The University of Edinburgh
Spring 2017

Peter Richtárik

Lecture 2

Randomized Methods for Solving Linear Systems: Four Reformulations and Basic Method

January 18, 2017

Solving Very Large Linear Systems

In this lecture we are concerned with the problem of solving a **linear system.** In particular, consider the problem

$$\text{solve} \quad \mathbf{A}x = b, \tag{1}$$

where $0 \neq \mathbf{A} \in \mathbb{R}^{m \times n}$, and m **is very large.**

Let $\mathbf{A}_{i:}$ denote the ith row of $\mathbf{A}$, and $\mathbf{A}_{:j}$ denote the jth column of $\mathbf{A}$. Let $b = (b_1, \ldots, b_m)$. Problem (1) can also be written more explicitly as a system of m **linear equations:**

$$\begin{aligned} \mathbf{A}_{1:}x &= b_1 \\ \mathbf{A}_{2:}x &= b_2 \\ &\vdots \\ \mathbf{A}_{m:}x &= b_m. \end{aligned}$$

The ith equation in the system has the form

$$\sum_{j=1}^{m} \mathbf{A}_{ij}x_j = b_j.$$

Consistency

We shall assume throughout the lecture that:

Assumption 1

Linear system (1) *is* **consistent.** *In other words, it has a solution:*

$$\mathcal{L} \overset{def}{=} \{x \ : \ \mathbf{A}x = b\} \neq \emptyset.$$

Introduction

- We will present a fundamental and flexible way of **reformulating each consistent linear system** into a **stochastic problem.**
- Stochasticity is introduced in a controlled way, into an otherwise deterministic problem, as a decomposition tool which can be leveraged to design efficient, granular and scalable **randomized algorithms.**
- **Two parameters:**
  - **Distribution** $\mathcal{D}$ describing an ensemble of random matrices $\mathbf{S} \in \mathbb{R}^{m \times q}$.
  - **Symmetric positive definite matrix** $\mathbf{B} \in \mathbb{R}^{n \times n}$**.**
- Presented approach and underlying theory support virtually all thinkable distributions $\mathcal{D}$. The choice of the distribution should ideally depend on the problem itself, as it will affect the complexity of the associated algorithms.
- In this specific setup (=linear systems), we can study many popular stochastic methods used in optimization and machine learning in a **unified way.** You will thus get strong foundations in the field.

Positive Definite Matrices

Definition 1

Let $\mathbf{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix.

(i) We say that $\mathbf{M}$ is **positive semidefinite** if

$$x^\top \mathbf{M} x \geq 0 \qquad \forall x \in \mathbb{R}^n.$$

We write this concisely as $\mathbf{M} \succeq 0$.

(ii) We say that $\mathbf{M}$ is **positive definite** if

$$x^\top \mathbf{M} x > 0 \qquad \forall 0 \neq x \in \mathbb{R}^n.$$

We write this concisely as $\mathbf{M} \succ 0$.

Inner Products and Norms

Inner Product in $\mathbb{R}^n$

Given a symmetric positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, we equip the space $\mathbb{R}^n$ with the **Euclidean inner product** defined by

$$\langle x, y\rangle_{\mathbf{B}} \stackrel{\text{def}}{=} x^\top \mathbf{B} y = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \mathbf{B}_{ij} y_j, \qquad x, y \in \mathbb{R}^n.$$

Norm in $\mathbb{R}^n$

We also define the **induced norm:** $\|x\|_{\mathbf{B}} \stackrel{\text{def}}{=} \sqrt{\langle x, x\rangle_{\mathbf{B}}}$.

Remark: We also use the short-hand notation $\|\cdot\|$ to mean $\|\cdot\|_{\mathbf{I}}$, where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. We shall sometimes refer to the quantity $\|x\|_{\mathbf{M}}$ with matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ being merely positive definite.

Stochastic Reformulations

4 Reformulations

We reformulate (1) into 4 seemingly different, but equivalent **stochastic problems:**

1. **Stochastic optimization problem** (2)
2. **Stochastic linear system** (4)
3. **Stochastic fixed point problem** (5)
4. **Probabilistic intersection problem** (6)

Reformulation 1: Stochastic Optimization Problem

Consider the **stochastic optimization problem**

$$\text{minimize} \quad f(x) \overset{\text{def}}{=} \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[f_{\mathbf{S}}(x)\right], \tag{2}$$

where

$$f_{\mathbf{S}}(x) \overset{\text{def}}{=} \frac{1}{2}\|\mathbf{A}x-b\|^2_{\mathbf{H}} = \frac{1}{2}(\mathbf{A}x-b)^\top \mathbf{H}(\mathbf{A}x-b). \tag{3}$$

When solving the problem, we do not have (or do not wish to exercise, as it may be prohibitively expensive) explicit access to f, its gradient or Hessian. Rather, we can repeatedly sample $\mathbf{S}\sim\mathcal{D}$ and receive unbiased samples of these quantities at points of interest. That is, we may obtain local information about the **stochastic function $f_{\mathbf{S}}(x)$,** such as the **stochastic gradient $\nabla f_{\mathbf{S}}(x)$,** or the **stochastic Hessian $\nabla^2 f_{\mathbf{S}}(x)$.**

Reformulation 2: Stochastic Linear System

Consider the following **stochastic linear system:**

$$\text{solve} \quad \mathbf{B}^{-1}\mathbf{A}^\top \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{H}\right]\mathbf{A}x = \mathbf{B}^{-1}\mathbf{A}^\top \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{H}\right]b. \tag{4}$$

- The system arises by pre-multiplying the system (1) on both sides from the left by matrix $\mathbf{P} = \mathbf{B}^{-1}\mathbf{A}^\top \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{H}\right]$.
- The **preconditioner** $\mathbf{P}$ is not assumed to be known explicitly.
- Instead, when solving the problem, we are able to sample $\mathbf{S}\sim\mathcal{D}$, obtaining an unbiased estimate of the preconditioner (not necessarily explicitly), $\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}$, for which we coin the name **stochastic preconditioner**. This gives us access to a random sample of system (4):
$$\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}\mathbf{A}x = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}b.$$
- This information can be obtained by repeatedly querying the stochastic sampling $\mathbf{S}\sim\mathcal{D}$ and utilized by an iterative algorithm.

Reformulation 3: Stochastic Fixed Point Problem

Let $\Pi^{\mathbf{B}}_{\mathcal{L}_\mathbf{S}}(x)$ denote the projection of x onto $\mathcal{L}_\mathbf{S} \overset{\text{def}}{=} \{x \;:\; \mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b\}$, in the norm $\|x\|_\mathbf{B} \overset{\text{def}}{=} \sqrt{x^\top\mathbf{B}x}$.

Consider the **stochastic fixed point problem**

$$\text{solve} \quad x = \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\Pi^{\mathbf{B}}_{\mathcal{L}_\mathbf{S}}(x)\right]. \tag{5}$$

That is, we seek to find a **fixed point** of the mapping

$$x \to \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\Pi^{\mathbf{B}}_{\mathcal{L}_\mathbf{S}}(x)\right].$$

When solving the problem, we do not have an explicit access to the average projection map. Instead, we are able to repeatedly sample $\mathbf{S}\sim\mathcal{D}$, and use the stochastic projection map $x \to \Pi^{\mathbf{B}}_{\mathcal{L}_\mathbf{S}}(x)$.

Reformulation 4: Probabilistic Intersection Problem

Note that $\mathcal{L} \subseteq \mathcal{L}_{\mathbf{S}}$ for all $\mathbf{S}$. We would wish to design $\mathcal{D}$ in such a way that a suitably chosen notion of an intersection of the sets $\mathcal{L}_{\mathbf{S}}$ is equal to $\mathcal{L}$. The correct notion is what we call **probabilistic intersection,** denoted $\cap_{\mathbf{S}\sim\mathcal{D}}\mathcal{L}_{\mathbf{S}}$, and defined as the set of points x which belong to $\mathcal{L}_{\mathbf{S}}$ with probability one.

This leads to the problem:

$$\text{find} \quad x \in \cap_{\mathbf{S}\sim\mathcal{D}}\mathcal{L}_{\mathbf{S}} \overset{\text{def}}{=} \{x \ : \ \text{Prob}(x \in \mathcal{L}_{\mathbf{S}}) = 1\}. \tag{6}$$

As before, we typically do not have an explicit access to the probabilistic intersection when designing an algorithm. Instead, we can repeatedly sample $\mathbf{S} \sim \mathcal{D}$, and utilize the knowledge of $\mathcal{L}_{\mathbf{S}}$ to drive the iterative process. If $\mathcal{D}$ is a discrete distribution, probabilistic intersection reduces to standard intersection.

Reformulations: Remarks

- All of the above formulations have a common feature: they all involve an expectation over $\mathbf{S} \sim \mathcal{D}$, and we either do not assume this expectation is known explicitly, or even if it is, we prefer, due to efficiency or other considerations, to sample from unbiased estimates of the objects (e.g., stochastic gradient $\nabla f_{\mathbf{S}}$, stochastic preconditioner $\mathbf{B}^{-1}\mathbf{A}^{\top}\mathbf{H}$, stochastic projection map $x \to \Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}}}(x)$, random set $\mathcal{L}_{\mathbf{S}}$) appearing in the formulation.
- As we shall see later, all these stochastic formulations are equivalent. In particular, the following sets are identical: the set of minimizers of the stochastic optimization problem (2), the solution set of the preconditioned system (4), the set of fixed points of the stochastic fixed point problem (5), and the probabilistic intersection (6).
- Further, we give necessary and sufficient conditions for this set to be equal to $\mathcal{L}$. Distributions $\mathcal{D}$ satisfying these conditions always exist, independently of any assumptions on the system beyond consistency. The simplest, but also the least useful choice of a distribution is to pick $\mathbf{S} = \mathbf{I}$ (the $m \times m$ identity matrix), with probability one. In this case, all of our reformulations become trivial.

Three Algorithms

Besides proposing a family of stochastic reformulations of (1), we also propose several stochastic algorithms for solving them:

- **Basic Method:** Algorithm 1
- **Parallel Method:** Algorithm 2
- **Accelerated Method:** Algorithm 3

Each method can be interpreted naturally from the viewpoint of each of the reformulations.

Basic Method

Basic Method

We shall now discuss some of the interpretations of the **basic method,** which performs updates of the form

$$x_{k+1} \overset{\text{def}}{=} \underbrace{x_k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x_k - b)}_{\phi_\omega(x_k, \mathbf{S}_k)}, \qquad (7)$$

where $\mathbf{S}_k \sim \mathcal{D}$ is sampled afresh in each iteration, and † denotes the **Moore-Penrose pseudoinverse.**

Algorithm 1 Basic Method

1: **Parameters:** distribution $\mathcal{D}$ from which to sample matrices; positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$; stepsize/relaxation parameter $\omega \in \mathbb{R}$
2: Choose $x_0 \in \mathbb{R}^n$ ▷ Initialization
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Draw a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
5: Set $x_{k+1} = x_k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x_k - b)$

Stochastic Gradient Descent

Algorithm 1 can be seen as **stochastic gradient descent,** with fixed stepsize, applied to (2).

In iteration k of the method, we sample $\mathbf{S}_k \sim \mathcal{D}$, and compute $\nabla f_{\mathbf{S}_k}(x_k)$, which is an unbiased stochastic approximation of $\nabla f(x_k)$. We then perform the step

$$x_{k+1} = x_k - \omega \nabla f_{\mathbf{S}_k}(x_k), \qquad (8)$$

where $\omega > 0$ is a stepsize.

Stochastic Newton Method

The method can also be seen as a **stochastic Newton method.**

At iteration k we sample $\mathbf{S}_k \sim \mathcal{D}$, and instead of applying the inverted Hessian of $f_{\mathbf{S}_k}$ to the stochastic gradient (this is not possible as the Hessian is not necessarily invertible), we apply the $\mathbf{B}$-pseudoinverse. That is, we perform the step

$$x_{k+1} = x_k - \omega(\nabla^2 f_{\mathbf{S}_k}(x_k))^{\dagger_{\mathbf{B}}} \nabla f_{\mathbf{S}_k}(x_k), \tag{9}$$

where $\omega > 0$ is a stepsize, and the $\mathbf{B}$-pseudoinverse of a matrix $\mathbf{M}$ is defined as $\mathbf{M}^{\dagger_{\mathbf{B}}} \overset{\text{def}}{=} \mathbf{B}^{-1}\mathbf{M}^\top(\mathbf{M}\mathbf{B}^{-1}\mathbf{M}^\top)^\dagger$.

Remark: One may wonder, why are methods (8) and (9) equivalent? Certainly, in general, stochastic gradient descent and stochastic Newton methods are not equivalent. It turns out that the stochastic gradient is always an eigenvector of the $\mathbf{B}$-pseudoinverse Hessian, with eigenvalue 1 (see Lemma **??**).

Stochastic Fixed Point Method

From the perspective of the stochastic fixed point problem (5), Algorithm 1 can be interpreted as a **stochastic fixed point method, with relaxation.**

We first reformulate the problem into an equivalent form using relaxation, which is done to improve the contraction properties of the map. We pick a relaxation parameter $\omega > 0$, and instead consider the equivalent fixed point problem

$$x = \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\omega\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}}}(x) + (1-\omega)x\right].$$

Now, at iteration k, we sample $\mathbf{S}_k \sim \mathcal{D}$, which enables us to obtain an unbiased estimate of the new fixed point mapping, and then simply perform one step of a fixed point method on this mapping:

$$x_{k+1} = \omega\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}_k}}(x_k) + (1-\omega)x_k. \tag{10}$$

Stochastic Projection Method

Algorithm 1 can also be seen as a **stochastic projection method** applied to the probabilistic intersection problem (6).

By sampling $\mathbf{S}_k \sim \mathcal{D}$, we are one of the sets defining the intersection, namely $\mathcal{L}_{\mathbf{S}_k}$. We then project the last iterate onto this set, in the $\mathbf{B}$-norm, followed by a relaxation step with relaxation parameter $\omega > 0$. That is, we perform the update

$$x_{k+1} = x_k + \omega(\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}_k}}(x_k) - x_k). \tag{11}$$

This is a randomized variant of an alternating projection method. Note that the representation of $\mathcal{L}$ as a probabilistic intersection of sets is not given to us. Rather, we construct it with the hope to obtain faster convergence.

Technicalities

Moore-Penrose Pseudoinverse - I

Let $\mathbf{M} \in \mathbb{R}^{n \times n}$. If $\mathbf{M}$ is invertible, then there exists a matrix, denoted by $\mathbf{M}^{-1} \in \mathbb{R}^{n \times n}$, called the **inverse matrix,** with the properties:

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{I}, \qquad \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}.$$

Not every square matrix has an inverse.

There is a generalization of the concept of the inverse, called **(Moore-Penrose) pseudoinverse.** The nice thing about it is that every matrix, even rectangular matrices, have a unique pseudoinverse.

Exercise 1

Use one of the properties of the pseudoinverse listed on the next slide to show that the pseudoinverse of a real number $\alpha \in \mathbb{R}$ is given by:

$$\alpha^{\dagger} = \begin{cases} \frac{1}{\alpha}, & \text{if} \quad \alpha \neq 0, \\ 0, & \text{if} \quad \alpha = 0. \end{cases} \tag{12}$$

Moore-Penrose Pseudoinverse - II

Fact 2

Every matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ has a unique pseudoinverse $\mathbf{A}^{\dagger} \in \mathbb{R}^{n \times m}$. Among others, this matrix satisfies the following properties:

(i) $\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{A}$

(ii) $\mathbf{A}^{\top} = \mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\top}$

(iii) $\mathbf{A}^{\top} = \mathbf{A}^{\top}\mathbf{A}\mathbf{A}^{\dagger}$

(iv) $\mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{A}^{\dagger}$

(v) $(\mathbf{A}^{\dagger})^{\top} = (\mathbf{A}^{\top})^{\dagger}$

(vi) $(\mathbf{A}\mathbf{C})^{\dagger} = \mathbf{C}^{\dagger}\mathbf{A}^{\dagger}$

(vii) $\mathbf{A}^{\top} = \mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\top}$

(viii) $\mathbf{A}^{\top} = \mathbf{A}^{\top}\mathbf{A}\mathbf{A}^{\dagger}$

Exercise 2

Use the above fact to show that: i) the pseudoinverse of a symmetric matrix is symmetric, ii) the pseudoinverse of a positive semidefinite matrix is positive semidefinite, iii) if $\mathbf{A}$ is invertible, then $\mathbf{A}^{\dagger} = \mathbf{A}^{-1}$.

Assumption on $\mathcal{D}$

From now on we will enforce the following assumption on $\mathcal{D}$:

Assumption 2 (Finite mean)

The random matrix

$$\mathbf{H} = \mathbf{H}_{\mathbf{S}} \stackrel{def}{=} \mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top \tag{13}$$

has a mean. That is, the following matrix has finite entries:

$$\mathrm{E}\left[\mathbf{H}\right] = \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{H}\right] = \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top\right]$$

Remark:

(i) $\mathbf{H} = \mathbf{H}_{\mathbf{S}}$ is a random matrix because it depends on the random matrix $\mathbf{S}$. However, in order to simplify notation, we will drop the subscript highlighting this dependency and will simply write $\mathbf{H}$.

(ii) If you worry about what expectation of a random matrix is, do not. By $\mathrm{E}\left[\mathbf{H}\right]$ we simply mean the matrix whose (i,j) entry is the mean of the (i,j) entry of $\mathbf{H}$:

$$\left(\mathrm{E}\left[\mathbf{H}\right]\right)_{ij} = \mathrm{E}\left[\mathbf{H}_{ij}\right].$$

Assumption on $\mathcal{D}$: Exercises

Exercise 3

(i) *Show that the matrix* $\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}$ *is symmetric and positive semidefinite.*

(ii) *It is known (see Exercise 2) that the pseudoinverse of a symmetric and positive semidefinite matrix is again symmetric and positive semidefinite. Show that* $\mathbf{H}$ *is symmetric and positive semidefinite.*

(iii) *Show that* $\mathrm{E}\left[\mathbf{H}\right]$ *is symmetric and positive semidefinite.*

Assumption on $\mathcal{D}$: Examples

Let $e_1, e_2, \ldots, e_m$ be standard basis vectors (aka coordinate vectors) in $\mathbb{R}^m$. That is, e_i is the vector whose all entries are zeros, except for the ith entry, which is equal to 1.

Example 3 (Uniform sampling unit of basis vectors)

Let $\mathcal{D}$ be the uniform distribution over $\{e_i\}$. That is, for all $i = 1, 2, \ldots, m$ we let

$$\mathbf{S} = e_i \quad \text{with probability} \quad 1/m.$$

We can then compute:

$$\mathrm{E}\left[\mathbf{H}\right] = \sum_{i=1}^{m} \tfrac{1}{m} e_i (\mathbf{A}_{i:}\mathbf{B}^{-1}\mathbf{A}_{i:}^\top)^\dagger e_i^\top = \tfrac{1}{m}\mathbf{Diag}\left(\alpha_1, \ldots, \alpha_m\right),$$

where

$$\alpha_i \overset{\text{def}}{=} (\mathbf{A}_{i:}\mathbf{B}^{-1}\mathbf{A}_{i:}^\top)^\dagger \overset{(12)}{=} 1/\|\mathbf{A}_{i:}^\top\|^2_{\mathbf{B}^{-1}}, \qquad i = 1, 2, \ldots, m,$$

and $\mathbf{Diag}\left(\alpha\right)$ is the diagonal matrix with vector α on the diagonal.

Note that if $\mathbf{A}$ has nonzero rows, then $\mathrm{E}\left[\mathbf{H}\right] \succ 0$.

Is f well defined?

We may wonder: does the expectation in (2) exist? That is, is f well defined? The next result says that all is fine.

Lemma 4

Let x_* be any solution of the linear system $\mathbf{A}x = b$ (that is, let $x_* \in \mathcal{L}$). Then

$$f_{\mathbf{S}}(x) = \frac{1}{2}(x - x_*)^\top \mathbf{A}^\top \mathbf{H}\mathbf{A}(x - x_*). \tag{14}$$

Moreover,

$$f(x) = \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[f_{\mathbf{S}}(x)\right] = \frac{1}{2}(x - x_*)^\top \mathbf{A}^\top \mathrm{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{H}\right]\mathbf{A}(x - x_*), \tag{15}$$

and hence $f(x)$ is finite for all $x \in \mathbb{R}^n$. Thus, f is well defined.

Proof of Lemma 4

Step 1: Since $x_* \in \mathcal{L}$, we have $\mathbf{A}x_* = b$. Plugging this into (3) gives (14).

Step 2: It remains to establish (15). In order to do so, we will use two facts.

Fact 5

For any $\mathbf{X} \in \mathbb{R}^{n\times n}$ and $h \in \mathbb{R}^n$, we have[1] $h^\top \mathbf{X} h = \mathrm{Trace}\left(\mathbf{X} h h^\top\right)$.

Fact 6

Fix any $\mathbf{M} \in \mathbb{R}^{n\times n}$. The map $\mathbf{X} \mapsto \mathrm{Trace}\left(\mathbf{X}\mathbf{M}\right)$ is linear.

Now back to the proof. Let $h = \mathbf{A}(x - x_*)$. Utilizing the above two facts, we get

$$f(x) \overset{(2)}{=} \mathrm{E}\left[f_{\mathbf{S}}(x)\right] \overset{(14)}{=} \tfrac{1}{2}\mathrm{E}\left[h^\top \mathbf{H} h\right] \overset{\text{(Fact 1)}}{=} \tfrac{1}{2}\mathrm{E}\left[\mathrm{Trace}\left(\mathbf{H} h h^\top\right)\right]$$
$$\overset{\text{(Fact 2)}}{=} \tfrac{1}{2}\mathrm{Trace}\left(\mathrm{E}\left[\mathbf{H}\right] h h^\top\right) \overset{\text{(Fact 1)}}{=} \tfrac{1}{2} h^\top \mathrm{E}\left[\mathbf{H}\right] h,$$

which gives (15). Note that when applying Fact 2, we have also used linearity of expectation.

[1] Recall that **trace** of a matrix, denoted $\mathrm{Trace}\left(\cdot\right)$, is the sum of its diagonal elements.