Massachusetts Institute of Technology

Department of Electrical Engineering and Computer Science

6.242, Fall 2004: MODEL REDUCTION *

Objectives and challenges of model reduction[1]

This introductory lecture discusses general formulatiions and basic difficulties associated with model reduction. The problem of simplifying matrix-vector multiplication is used as an example.

1.1 Introduction

Model reduction is one of the most widely encountered "dynamical systems" tasks. In practice, it enables practical use of first principles models for physical phenomena described by partial differential equations, advanced control and signal processing algorithms. Model reduction can also be used to facilitate system identification, data compression, and knowledge extraction.

This class will concentrate on mathematical techniques of model reduction applicable to linear time invariant (LTI) systems, while also venturing into the field of nonlinear models whenever possible. Intentionally simplified, if not trivialized, application examples will be used for motivation and illustration of the theory. The lecture notes presentation will be rigorous, but many formal details will be skipped in the lectures.

1.1.1 Motivating example: the heat equation

Consider the task of modeling the dynamical dependence of time-varying temperature $y = y(t)$ at a point A of a thin non-homogeneous circular wire on the temperature $u = u(t)$

 Version of September 8, 2004

[1]The author is grateful to J. White and L. Daniel for many useful comments and suggestions

at the opposite point B, where the wire is being forcefully heated/cooled (see Figure 1.1). The relation between $u = u(t)$ and $y = y(t)$ can be written in the form of a partial

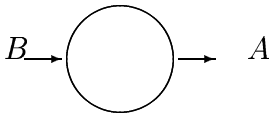

Figure 1.1: Example: the heat equation

differential equation

$$\frac{dv(t,\theta)}{dt} = K(\theta)\frac{d^2v(t,\theta)}{d\theta^2},$$

where $v(t,\theta)$ is temperature of the wire at time t at the point with angular position θ ($\theta = 0$ corresponds to point A), i.e.

$$v(t,-\pi) = v(t,\pi) = u(t), \quad y(t) = v(t,0),$$

and $K(\theta) > 0$ is a given position-dependent coefficient describing local properties of the wire. This model, while accurate subject to some idealizing assumptions, is not good for simulation or feedback control design. A model reduction technique would generate a low order LTI system providing an accurate approximation of the true dynamics.

1.1.2 Motivating example: system identification

An important task performed repeatedly by wireless communication devices is "channel identification", which essentially means finding a good model for electromagnetic signal propagation between two communication points. This can be accomplished by sending a white noise signal through the channel and calculating the statistical spectrum of the signal received. As the next step, this (noisy) spectral data has to be fitted by a low order rational function, as shown on If the order of approximation is allowed to be large (to match the number of data points available), the resulting model will try to fit the noise component of the data, which is highly undesirable. An optimal model reduction technique will help to avoide noise fitting by finding the best low order rational approximation.

1.1.3 Simplification of general system models

A general objective of model reduction as a mathematical discipline can be described as that of finding *efficient ways of deriving adequate simplified models of complex systems.*

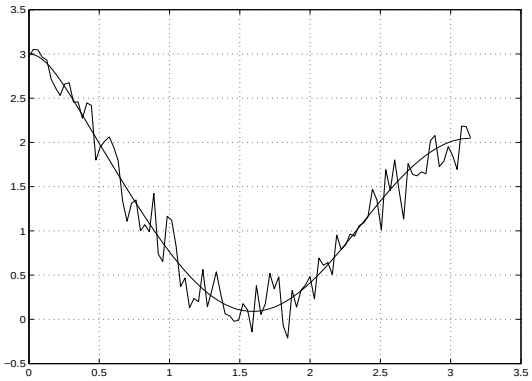

Figure 1.2: Example: noisy data fitting

Here the word *system* refers to a transformation which defines a family of real parameters (*output data*) as a function of another family of real parameters (*input data*). A major object of study in these lectures will be the class of *linear time-invariant dynamical systems*.

A particular technique of model reduction would deal with a specific class of system descriptions by

- specifying a quantative *system complexity measure* for models from the class;
- defining a notion of accuracy ("adequacy") for replacing one ("complex") model with another ("reduced") one;
- supplying a numerical algorithm which actually performs the reduction;
- presenting evidence (experimental or via a mathematical proof) that the algorithm produces accurate models of low complexity in reasonable time ("algorithm cost");

Accordingly, to compare two methods of model reduction, one has to take into account accuracy, system complexity reduction, and algorithm cost guarantees associated with them.

1.2 Example: reduction of matrix-vector products

Action of a given linear transformation $M : f \mapsto y$ on its unspecified input f is frequently represented as multiplication of a variable real m-vector f by a given real n-by-m matrix M: $y = Mf$. In some applications (image processing, optical simulations), it is desirable

to perform such matrix-vector multiplications, where M is known a-priori, as quickly as possible, which typically means minimization of the number of computer operations (multiplication, addition, copying) needed to compute $y = Mf$. It may also be admissible to have a certain amount r of relative error when calculating y quickly.

While the standard definition of a matrix-vector product involves nm operations, some linear transformations $f \mapsto y = Mf$ can be performed much faster, depending on M. Moreover, it could be possible to reduce the minimal number of operations by perturbing M slightly.

1.2.1 Matrix model reduction: a number-of-operations setup

The task of finding, for a given M, a faster way of producing an approximation of the product $y = Mf$, can be viewed as a model reduction problem, with the notions of "systems", "complexity", "accuracy", and "efficiency" defined as follows.

Systems: every real n-by-m matrix M defines a *matrix model system* with the set of admissible inputs defined as the set $\mathbf{R}^n = \{f\}$ of all real m-vectors f, and outputs $y \in \mathbf{R}^n$ defined by $y = Mf$.

System complexity: define complexity $\mathcal{N} = \mathcal{N}(M)$ of a matrix model system as the minimal number of binary addition, multiplication, and memory copy operations needed to implement the input-to-output transformation $f \mapsto y = Mf$. Note that, while it is easy to *define* $\mathcal{N}(M)$ it is usually quite difficult to *calculate* the quantity.

Accuracy: define a numerical measure of error of approximating an original matrix M by a "reduced" matrix $\hat{M}$ as the maximal Euclidean norm $|e|$ of the output matching error vector $e = Mf - \hat{M}f$ when the Euclidean norm of input f is bounded by $|f| \leq 1$. In linear algebra, this quantity is known as the *operator norm* $\|M - \hat{M}\|$, or *largest singular value* $\sigma_{\max}(M - \hat{M})$ of $M - \hat{M}$.

Algorithm cost: define efficiency of a model reduction algorithm as the maximal number of operations needed to produce a reduced model $\hat{M}$ for a given M.

Even after the basic notions of system class, complexity, accuracy, and efficiency are defined, there is a variety of possible approaches to follow. For example, one can formulate an optimal matrix reduction problem (given M and $r > 0$, find $\hat{M} = \hat{M}(M, r)$ such that $\|M - \hat{M}\| \leq r$, and $\mathcal{N}(\hat{M})$ is as small as possible). Alternatively, one can pick a transformation $M \mapsto \mathcal{F}(M)$ and try to prove, either formally or via numerical experiments, that, for most M, $\mathcal{N}(\mathcal{F}(M)) < clN(M)$, and $\|M - \mathcal{F}(M)\|$ is small. However, despite the number of options available, most of them are likely to end up nowhere, due to the difficulty of working with the "uncomputable" complexity measure $\mathcal{N}(M)$.

1.2.2 Matrix model reduction: a rank setup

As it frequently happens, a relatively minor modification of the matrix reduction setup leads to a computationally efficient algorithm.

Let us note that the rank of M defines an upper bound on the minimal number of operations needed to implement transformation $f \mapsto Mf$. Indeed, if $k = \mathrm{rank}(M)$ then $M = VU$ where V is an n-by-k matrix, and U is a k-by-m matrix. Hence, Mf can be found by forming Uf first (km operations) and then calculating VUf (kn operations), which yields a total of $k(n+m)$ operations. When $k \ll n$ and $k \ll m$, this constitutes a substantial reduction in the number of operations.

Thus, it appears to be natural to consider a modified matrix reduction setup, in which $\mathcal{N}(\hat{M})$ is replaced by $\mathrm{rank}(\hat{M})$ as a system complexity measure. This will be referred to as the *matrix rank reduction*. Note that this is not equivalent to the number-of-operations setup, because multiplication by some matrices of full rank can be performed very quickly (for example, multiplication by an upper triangular n-by-n matrix with 1's above the diagonal can be performed in n steps).

1.2.3 Optimal matrix rank reduction

A solution of the optimal matrix rank reduction problem is a standard part of introductory linear algebra. Indeed, let $\sigma_k^2 = \lambda_k$, where $k \in \{1, 2, \ldots, m\}$ and $\sigma_k \geq 0$, be the ordered eigenvalues of $M'M$ (i.e. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$). Then $\|M - \hat{M}\| \geq \sigma_k$ for every matrix $\hat{M}$ of rank less than k. Moreover, a matrix $\hat{M}$ of rank less than k such that $\|M - \hat{M}\| = \sigma_k$ can be defined by $\hat{M} = VV'MUU'$, where the columns of U are the first $k-1$ orthonormal eigenvectors of $M'M$ (i.e. those corresponding to eigenvalues $\lambda_1, \ldots, \lambda_{k-1}$), and the columns of V are the first $k-1$ orthonormal eigenvectors of MM'.

Thus, when the accuracy measure is defined as the operator norm, optimal matrix rank reduction can be implemented via the so-called *singular value decomposition*, which is, essentially eigenvalue decomposition of positive semidefinite matrices. It is interesting to note that for many other accuracy measures there is still no efficient solution available for the optimal matrix rank reduction problem.

1.3 Challenges of model reduction

There is a number of formulation changes which will complicate the matrix reduction process dramatically. It is instructive to discuss some of these modifications, as they are typical for the general model reduction research area.

1.3.1 Partially defined systems

We started with the assumption that all coefficients of M are known precisely. This is frequently not the case: some of the coefficients may be unknown, while a general constraint is placed on their dependence on the index (say, $|M_{i,j} - M_{i,j+1}| \leq 0.01$). Alternatively, a noise factor may be present in *all* coefficient data.

1.3.2 A system which is too large

A very large matrix can be defined "analytically", while being too large to be stored in the memory of a computer. For example, what about a 10^{20}-10^{20} matrix M with entries

$$M_{ij} = \frac{1}{2 + i^2 + j^2 + \cos(i)}?$$

1.3.3 Non-linear and uncertain systems

Already a very simple type of nonlinearity – parameter dependence – makes a model reduction problem much harder. Model reduction of more general nonlinear systems remains largely an uncharted territory.

1.3.4 How to compare model reduction methods?

Model reduction methods have to be compared with respect to many parameters (accuracy, algorithm cost, type of complexity measure, etc.) An algorithm which is optimal accuracy-wise may be prohibitevily expensive, and a cheap algorithm may produce extremely inadequate reduced models. Moreover, theoretical proofs of performance are rare.