# Πρόβλεψη της Εξέλιξης Χρονοσειρών μέσω Στοχαστικών Μοντέλων

### Κωνσταντίνος Παπαδημητρίου

## Η βασική ιδέα

Στα Νευτώνεια δυναμικά συστήματα η μελλοντική θέση ενός σώματος εξαρτάται από την παρούσα θέση του, την ταχύτητα και την επιτάχυνσή του, ή αλλιώς την πρώτη και δεύτερη χρονική παράγωγο. Αντλώντας έμπνευση από την κλασσική φυσική, το παρόν σύστημα προσπαθεί να προβλέψει την εξέλιξη μιας χρονοσειράς χρησιμοποιώντας σαν βάση την τιμή της στην παρούσα χρονική στιγμή, καθώς επίσης και τις τιμές των χρονικών παραγώγων της. Προφανώς, επειδή οι πραγματικές χρονοσειρές μπορεί να προέρχονται από μία πληθώρα φαινομένων, ελάχιστα από τα οποία υπακούουν στις αρχές τις Νευτώνειας δυναμικής, εδώ δεν υπάρχει κανένας λόγος το σύστημα να περιοριστεί στην πρώτη και δεύτερη παράγωγο, αλλά αντίθετα εξετάζει τις τιμές όλων των παραγώγων μέχρι μίας μέγιστης τάξης Νmax, που μάλιστα επιλέγεται από τον χρήστη ως ελεύθερη παράμετρος. Η βασική ιδέα στηρίζεται στο ότι το σύστημα θα δεχθεί μία χρονοσειρά (επαρκούς μήκους) ως input, θα εξετάσει όλες τις μεταβάσεις που έχουν λάβει χώρα σε αυτή, δηλαδή ποιά τιμή είχε η χρονοσειρά μία χρονική στιγμή t και σε ποιά κατέληξε την αμέσως επόμενη χρονική στιγμή t+1, και θα προσπαθήσει να “μάθει” από αυτές, ώστε ανά πάσα στιγμή στο μέλλον να μπορεί να προβλέψει (με ένα στατιστικό τρόπο) την επόμενη τιμή της.

## Φασικός χώρος

Έχοντας ορίσει την μέγιστη τάξη παραγώγισης, το σύστημα παράγει από την input χρονοσειρά τον φασικό χώρο μέσα στον οποίο θα διαδραματιστεί όλη η διαδικασία. Πρώτα απ’όλα, το πρόγραμμα παράγει τις χρονοσειρές που αντιστοιχούν στις χρονικές παραγώγους τις input σειράς. Ως proxy για την χρονική παραγώγιση θεωρείται η απλή διαφορική χρονοσειρά, η οποία σε κάθε σημείο της n δίνει την διαφορά της τιμής που είχε η αρχική σειρά στην θέση n από εκείνην που είχε στην θέση n-1. Οι ανώτερες τάξεις, παράγονται διαφορίζοντας την χρονοσειρά που αντιστοιχεί στην αμέσως προηγούμενη τάξη.

με d την τάξη της εκάστοτε διαφορικής χρονοσειράς (d=0 για την αρχική input χρονοσειρά) και Xi την τιμή της στην θέση (χρονική στιγμή) i. Ο φασικός χώρος του συστήματος ορίζεται ως ο χώρος Nmax+1 διαστάσεων των τιμών της αρχικής χρονοσειράς και όλων την παραγώγων της ( +1 γιατί για Nmax=0 δεν θα υπολογιστούν παράγωγοι και η πρόβλεψη θα γίνει μόνο με βάση τις τιμές της ίδιας της input χρονοσειράς, οπότε το φασικός χώρος γίνεται μονοδιάστατος).

## Ορισμός της κατάστασης

Έχοντας ορίσει τον φασικό χώρο της διαδικασίας, η “κατάσταση” την κάθε χρονική στιγμή μπορεί να απεικονιστεί ως ένα σημείο στον φασικό αυτό χώρο και η εξέλιξη της χρονοσειράς (και των παραγώγων της) θα σχηματίζει μία τροχιά στον χώρο αυτό. Επειδή οι τιμές της χρονοσειράς αναμένεται να είναι συνεχείς στο σύνολο των πραγματικών αριθμών, απαιτείται μία διακριτοποίηση των τιμών αυτών. Ο χρήστης δίνει τιμή σε μία δεύτερη ελεύθερη παράμετρο Nbins η οποία εκφράζει ακριβώς το σε πόσες διακριτές τιμές θα χωριστεί το πεδίο τιμών της χρονοσειράς και το πρόγραμμα, για κάθε διάσταση του φασικού χώρου (για κάθε παράγωγο δηλαδή) αντιστοιχεί την κάθε τιμή σε έναν ακέραιο από 0 έως Nbins-1 σύμφωνα με τη σχέση

X είναι η χρονοσειρά (ή οποιαδήποτε από τις παραγώγους της), Xi η τιμή της στην χρονική θέση i, τα min(X) και max(X) δίνουν αντίστοιχα την ελάχιστη και μέγιστη τιμή της χρονοσειράς, ενώ η συνάρτηση floor() στρογγυλοποιεί το αποτέλεσμα της πράξης προς το 0. Η τελική τιμή si θα είναι τώρα ένας ακεραιος με τιμές από 0 ως Nbins-1. Υπάρχει μόνο μία ειδική περίπτωση, όταν το Xi είναι ίσο με την μέγιστη τιμή max(X), όπου το αποτέλεσμα θα δώσει floor((max(X) – min(X)) / (max(X) – min(X)) \* Nbins) = floor(Nbins) = Nbins, οπότε και ο αλγόριθμος θα μετατρέψει την τιμή αυτή σε Nbins-1 κατά παράβαση της ανωτέρω σχέσης, συμπεριλαμβάνοντας και τις τιμές αυτές στο τελευταίο bin. Έτσι, από εκεί όπου κάθε χρονική στιγμή η κατάσταση οριζόταν από τις τιμές της χρονοσειράς και των παραγώγων της από το διάνυσμα με , τώρα η ίδια κατάσταση έχει διακριτοποιηθεί στο άνυσμα με .

## Απεικόνιση των καταστάσεων

Μέχρι το σημείο αυτό κάθε χρονική στιγμή έχει απεικονιστεί ώς ένα σημείο ενός διακριτοποιημένου Nmax-1διάστατου χώρου (grid). Η διαχείριση ενός τέτοιου συστήματος προγραμματιστικά δημιουργεί πολλά προβλήματα, κυρίως γιατί η μέγιστη τάξη διαφόρισης Nmax είναι ελεύθερη παράμετρος που θα δοθεί από τον χρήστη. Όταν το πρόγραμμα προσπαθήσει να κάνει μετρήσει το πόσες φορές εμφανίστηκε η κάθε κατάσταση θα χρειαστεί να κάνει initialize έναν πίνακα με διαστάσεις Nbins x Nbins x Nbins x … x Nbins, τάξης ίσης με Nmax. Στην C όμως (και σε άλλες γλώσσες προγραμματισμού) ένας πίνακας θα απεικονίζεται ως pointer σε pointers, ένας τανυστής τρίτης τάξης ως pointers σε pointers σε pointers κ.ο.κ και είναι δύσκολο να γραφεί ο κώδικας και οι μέθοδοι από τις οποίες θα απαρτίζεται χωρίς να είναι ήδη γνωστή η μέγιστη αυτή τάξη. Έτσι απαιτείται διαφορετική διαχείριση του προβλήματος. Ένας εύκολος τρόπος να γίνει αυτό είναι να αποδοθεί σε κάθε κατάσταση ένας μοναδικός αριθμός (id) οπότε ο οποιοσδήποτε πίνακας Ν τάξης θα μετατραπεί σε μία απλή, μονοδιάστατη λίστα αριθμημένων καταστάσεων. Για να γίνει αυτό, το πρόγραμμα συμπεριφέρεται στο καταστατικό διάνυσμα των διακριτοποιημένων θέσεων ως αριθμό σε ένα μη-δεκαδικό σύστημα. Για παράδειγμα, αν έχει επιλεγεί Nmax=1 και Nbins=2, τότε κάθε κατάσταση θα ορίζεται από ένα ζεύγος αριθμών (διάσταση = Nmax+1 = 2) που θα παίρνουν τιμές από 0 ως Nbins-1=1, οπότε μία κατάσταση θα είναι η (0,0) μία άλλη η (0,1) κτλ. Τα ανύσματα αυτά μπορούν να αντιμετωπιστούν ως αριθμοί στο δυαδικό σύστημα και να αντικατασταθούν με τον αντίστοιχο δεκαδικό αριθμό τους, πχ (0,0) = 0, (0,1) = 1 κτλ. Στη γενική περίπτωση, κάθε άνυσμα ακεραίων με τιμές από 0 ως Nbins-1 μετατρέπεται από το Nbins-ικό σύστημα στο δεκαδικό και αποδίδεται στην εκάστοτε κατάσταση αυτός ο αριθμός ως unique state id. Έχοντας τα Nmax και Nbins ο αλγόριθμος μπορεί εύκολα να υπολογίσει τον μέγιστο δυνατό αριθμό καταστάσεων Nstates = Nbins^(Nmax+1) και να ετοιμάσει την λίστα που θα υποδεχθεί τις καταστάσεις, τους counters που θα απαριθμήσουν το πόσες φορές εμφανίστηκε η κάθε μία και γενικά όλες τις σχετικές δομές χωρίς να χρειάζεται να ανατρέχει σε περίπλοκες αναπαραστάσεις.

## Πρόβλεψη της επόμενης κατάστασης

Η βασική αρχή του συστήματος έγκειται στο ότι προσπαθεί από τη γνώση των προηγούμενων καταστατικών μεταβάσεων να προβλέψει τις επόμενες. Ένα μειονέκτημα που προκύπτει από αυτό (και είναι κοινό σε όλες τις μεθόδους που ακολουθούν την ίδια αρχή) είναι η αδυναμία του συστήματος να προβλέψει μεταβάσεις που δεν έχει δει ποτέ πριν. Για παράδειγμα, έστω ότι η αρχική χρονοσειρά είναι η (1, 2, 3, 4, 5). Οποιοσδήποτε άνθρωπος δει αυτή τη σειρά θα προβλέψει αμέσως ότι οι επόμενες τιμές της θα είναι κατά πάσα πιθανότητα οι 6, 7, 8 κτλ. Ένας αλγόριθμος όμως θα διαβάσει τη σειρά αυτή αποθηκεύοντας ως πληροφορία ότι “η κατάσταση (2) ακολουθεί την κατάσταση (1)”, “η κατάσταση (3) ακολουθεί την κατάσταση (2)”, “η κατάσταση (4) ακολουθεί την κατάσταση (3)” κ.ο.κ. Αν ξαναδεί ως input την κατάσταση (2) θα προβλέψει σωστά ως πιθανή εξέλιξη την (3), αν όμως δει την κατάσταση (5) δεν θα μπορέσει ποτέ να κάνει οποιαδήποτε πρόβλεψη, γιατί με βάση την input χρονοσειρά δεν έχει υπάρξει στο ιστορικό κάποια μετάβαση από την κατάσταση (5) σε οποιαδήποτε άλλη.

Ένας τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα προκύπτει από την χρήση των παραγώγων. Στην χρονοσειρά του παραδείγματος είναι προφανές ότι η παράγωγος παραμένει παντού σταθερή και ίση με 1. Με βάση αυτή την πληροφορία, το σύστημα θα μπορούσε να προβλέψει, όχι την επόμενη θέση, αλλά την επόμενη διαφορά ως ίση με 1, και με βάση αυτό και την παρούσα κατάσταση που είναι η 5 να αποφανθεί ότι η επόμενη θέση θα είναι η 6. Έτσι, στα πλαίσια του παρόντος συστήματος, η πρόβλεψη αναφέρεται πάντα στην διαφορική χρονοσειρά μέγιστης τάξης και με βάση αυτήν και τις τιμές των διαφορικών μικρότερης τάξης προκύπτει αναδρομικά η μελλοντική τιμή της αρχικής χρονοσειράς.

## Πίνακας πιθανοτήτων μετάβασης

Η μέθοδος με την οποία το σύστημα διδάσκεται από το παρελθόν είναι η κατασκευή του πίνακα πιθανοτήτων μετάβασης. Για κάθε χρονική στιγμή, το σύστημα διαβάζει την τιμή της αρχικής χρονοσειράς και των παραγώγων της, μετατρέπει το άνυσμα των τιμών σε άνυσμα στον διακριτοποιημένο χώρο και από εκεί υπολογίζει τον id αριθμό της εν λόγω κατάστασης. Έχοντας αυτό ως βάση, διαβάζει την επόμενη τιμή της διαφορικής χρονοσειράς μέγιστης τάξης και υπολογίζει σε ποιό bin αντιστοιχεί. Έχοντας αυτή τη μετάβαση, αυξάνει τον αντίστοιχο counter σε έναν Nstates x Nbins πίνακα και μετά προχωρά στην επόμενη χρονική στιγμή. Επειδή στο τέλος η πρόβλεψη θα γίνει στο ανώτατης-τάξης-διαφορικό, όπως περιγράφηκε στην προηγούμενη παράγραφο, το σύστημα κρατά επίσης και την τιμή της μετάβασης (την πραγματική τιμή, όχι απλά το bin) και στο τέλος υπολογίζει για κάθε bin του διαφορικού μέγιστης τάξης, την μέση τιμή.

Η δεύτερη αδυναμία που χαρακτηρίζει συστήματα τέτοιου τύπου είναι ότι ορισμένες φορές δεν εμφανίζονται όλες οι δυνατές καταστάσεις καθώς το σύστημα διατρέχει το ιστορικό της χρονοσειράς και κατά συνέπεια μεγάλα τμήματα του πίνακα μπορεί να παραμείνουν κενά. Το γεγονός αυτό θα οδηγήσει σε αδυναμία πρόβλεψης, οπότε απαιτείται η αντιμετώπισή του, η οποία επιτυγχάνεται εδώ με ένα interpolation schema. Προφανώς η επιλογή της κατάλληλης interpolation μεθόδου είναι ένα μεγάλο ζήτημα που εξαρτάται και από την φύση των χρονοσειρών που θα επιδιώξει να αναλύσει ο αλγόριθμος, αλλά ως ένα απλό και εύκολο στην υλοποίηση scheme, με άμεση φυσική ερμηνεία, επιλέχθηκε το Laplacian Interpolation, το οποίο ονομάζεται έτσι καθώς το αποτέλεσμά του δίνει τις λύσεις της Λαπλασιανής εξίσωσης . Ο αλγόριθμος του Laplacian Interpolation λειτουργεί αποδίδοντας σε κάθε στοιχείο του πίνακα την μέση τιμή των γειτόνων του και συνεχίζει έως ότου η μεταβολή της τιμής του στοιχείου από την προηγούμενη τιμή του να γίνει πάρα πολύ μικρή (κάτω από κάποιο όριο). Στην παρούσα περίπτωση τα στοιχεία του πίνακα μετάβασης που έχουν ήδη μη-μηδενικές τιμές παραμένουν σταθερά, ενώ ένα for loop διατρέχει τα υπόλοιπα στοιχεία και τους αποδίδει τιμές ίσες με τον μέσο όρο των γειτόνων τους. Ο “γείτονας” εδω ορίζεται αναφορικά με τον αρχικό διακριτοποιημένο πολυδιαστατικό χώρο, έτσι πχ, για Nmax=1 και Nbins=3, η κατάσταση (1,1) θα έχει γείτονες τις καταστάσεις (2,1), (0,1), (1,2) και (1,0), οπότε η γειτνίαση αναφέρεται στους άμεσους γείτονες και όχι στους διαγώνιους, όχι δηλαδή στην κατάσταση (2,2) ή (0,0). Η μέθοδος υπολογίζει τους σωστούς γείτονες, με βάση τον αρχικό πολυδιάστατο χώρο των διακριτοποιημένων καταστάσεων, κατόπιν υπολογίζει τα αντίστοιχα id και τέλος τον μέσο όρο τους. Το for loop διατρέχει όλες τις τιμές του πίνακα (εκτός εκείνων που έχει δηλωθεί ότι θα παραμείνουν ανέπαφες από τη διαδικασία) και στο τέλος υπολογίζει τη μέγιστη αλλαγή που επήλθε στα στοιχεία του πίνακα. Αν αυτή υπερβαίνει κάποια τιμή κατωφλίου, η διαδικασία επαναλαμβάνεται, αλλιώς τερματίζει. Ειδική περίπτωση αποτελούν οι συνοριακές καταστάσεις, πχ στο προηγούμενο παράδειγμα η κατάσταση (0,0) η οποία έχει μόνους γείτονες τις (1,0) και (0,1) καθώς οι (-1,0) και (0,-1) βρίσκονται εκτός του χώρου (δεν υπάρχουν αρνητικοί αριθμοί αφού οι επιτρεπτές τιμές κυμαίνονται από 0 ως Nbins-1). Για αυτές θεωρείται ότι όλοι οι γείτονες που βρίσκονται εκτός του grid έχουν τιμή σταθερά μηδέν (δίνονται δηλαδή ανοιχτές συνοριακές συνθήκες στο πρόβλημα)!

Τέλος, σε κάθε γραμμή του πίνακα υπολογίζεται το άθροισμα και τα στοιχεία τους διαιρούνται με το άθροισμα αυτό, ώστε ο πίνακας των counts να μετατραπεί σε πίνακας πιθανοτήτων. Έτσι για κάθε κατάσταση υπάρχει έτοιμη η κατανομή της πιθανότητας μετάβασης στα bins του μέγιστης-τάξης-διαφορικού της χρονοσειράς.

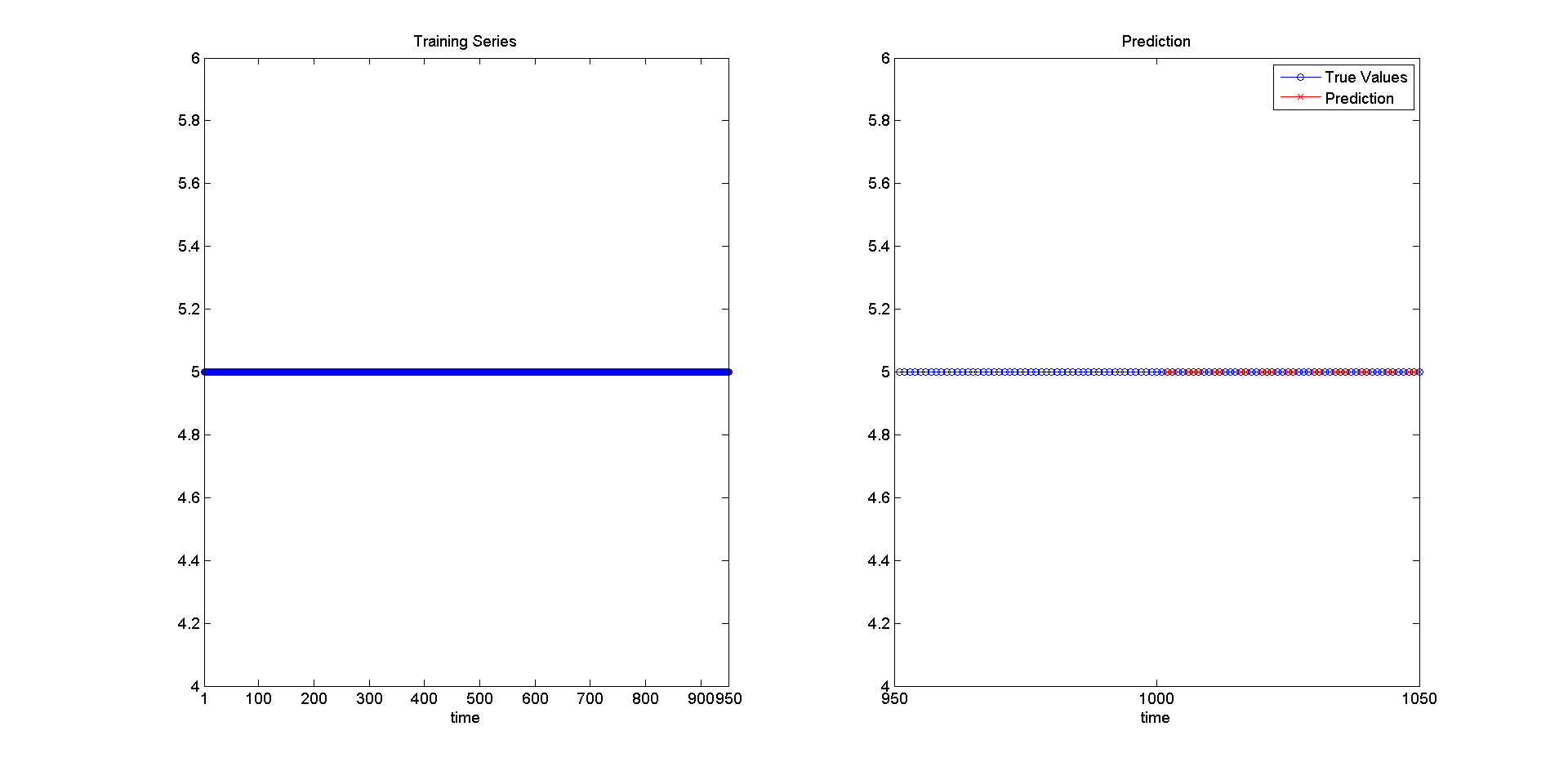
## Πρόβλεψη

Έχοντας έτοιμο τον πίνακα με τις πιθανότητες των μεταβάσεων είναι πλέον εύκολο να προβλεφθούν οι επόμενες τιμές της χρονοσειράς. Συγκεκριμένα, το σύστημα θα πάρει την τελευταία τιμή της χρονοσειράς και τις τιμές των διαφορικών της, θα την αντιστοιχήσει σε κάποιο αριθμό id και μετά θα διαβάσει την αντίστοιχη γραμμή του πίνακα πιθανοτήτων μετάβασης. Από εκεί, ο αλγόριθμος υπολογίζει έναν τυχαίο αριθμό και με βάση αυτόν και την κατανομή πιθανοτήτων που αντιστοιχούν στην κατάσταση στην οποία βρίσκεται, επιλέγει το bin της επόμενης τιμής του διαφορικού μέγιστης τάξης. Έχοντας ήδη υπολογίζει τις μέσες τιμές που αντιστοιχούν σε κάθε τέτοιο bin, διαβάζει την αντίστοιχη τιμή και την προσθέτει στην τιμή του προηγούμενης τάξης διαφορικού, για να πάρει την μελλοντική τιμή του διαφορικού αυτού. Συνεχίζει αναδρομικά προσθέτοντας τις τιμές αυτές στα διαφορικά χαμηλότερης τάξης, έως ότου έχει προβλέψει την επόμενη τιμή του διαφορικού τάξης 0, δηλαδή της ίδιας της χρονοσειράς. Η διαδικασία μετά συνεχίζεται με βάση αυτήν την τιμή κ.ο.κ. Έτσι ο αλγόριθμος προβλέπει με πιθανοκρατικό τρόπο διαφορετικά σενάρια για το πώς δύναται να είναι η εξέλιξη της σειράς.

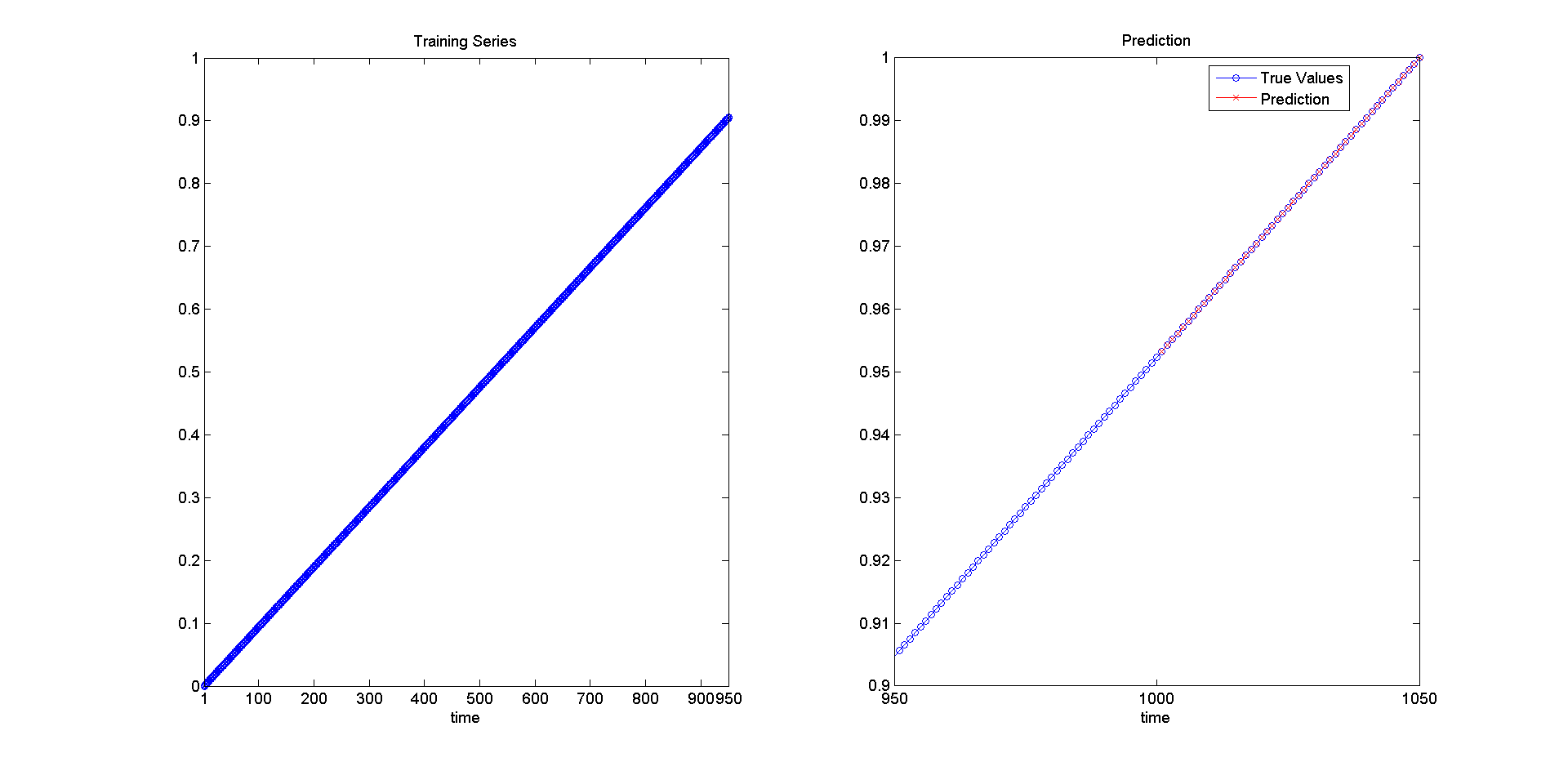
Απαιτείται μία μόνο σημείωση, για τις ειδικές περιπτώσεις στις οποίες η τιμή της χρονοσειράς ή η τιμή κάποιας από τις παραγώγους τις, δεν εμπίπτει στα όρια με βάση τα οποία έχει χτιστεί ο πίνακας μετάβασης˙ είναι δηλαδή μεγαλύτερη (ή μικρότερη) από το max (ή min) που χρησιμοποιήθηκε στην σχέση (2). Σε αυτές τις περιπτώσεις, αντιστοιχίζεται αυτή η τιμή στο μέγιστο (ή στο μηδενικό) bin, έτσι ώστε η πρόβλεψη να γίνει με την καλύτερη δυνατή προσέγγιση της κατάστασης που προσφέρει η υπάρχουσα δομή του πίνακα. Φυσικά, σε πραγματικές εφαρμογές απαιτείται η επανακατασκευή του πίνακα πιθανοτήτων μετάβασης, όταν η χρονοσειρά έχει αλλάξει αρκετά από την αρχική της μορφή.

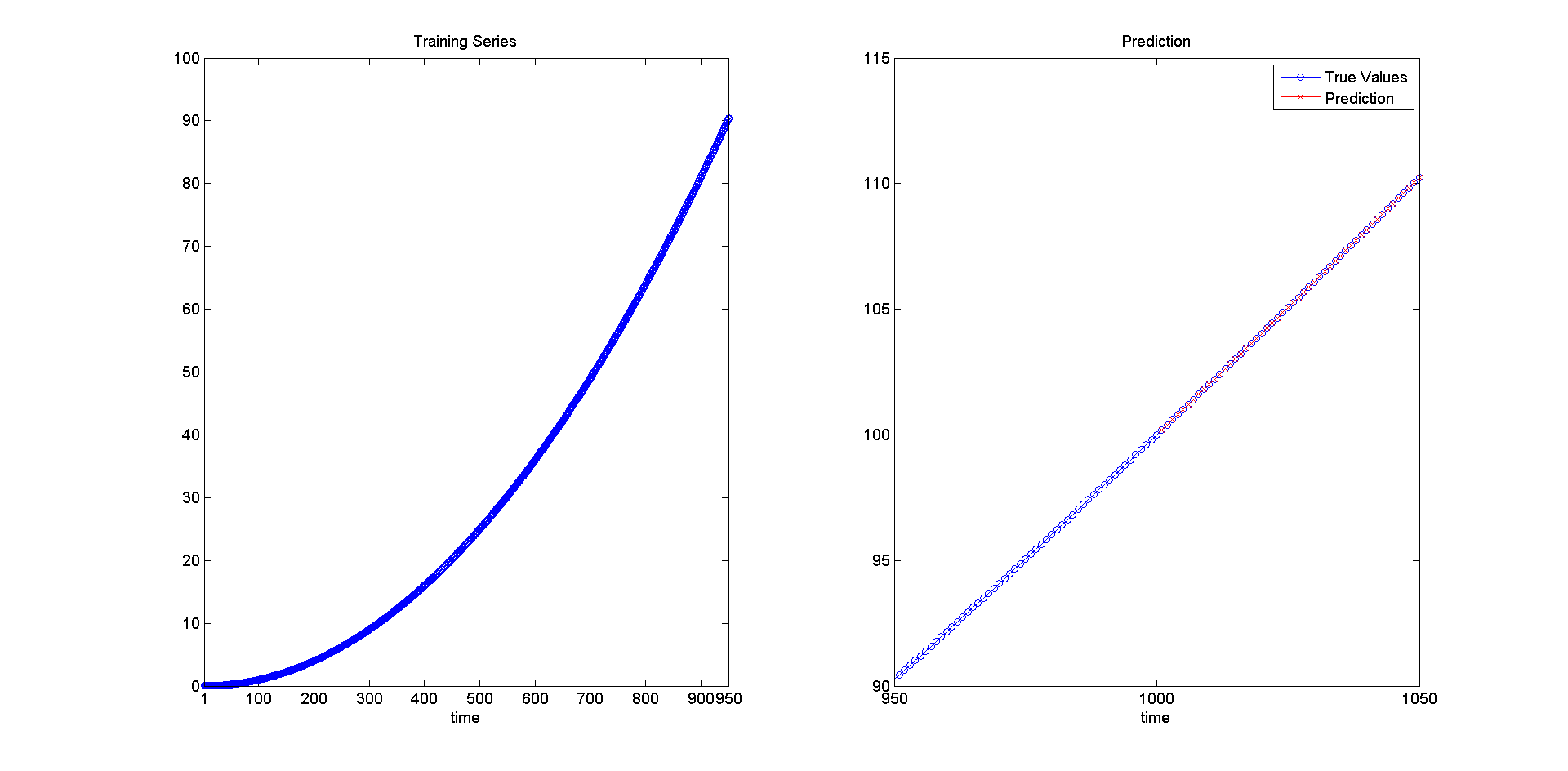
## Εφαρμογές σε θεωρητικές χρονοσειρές

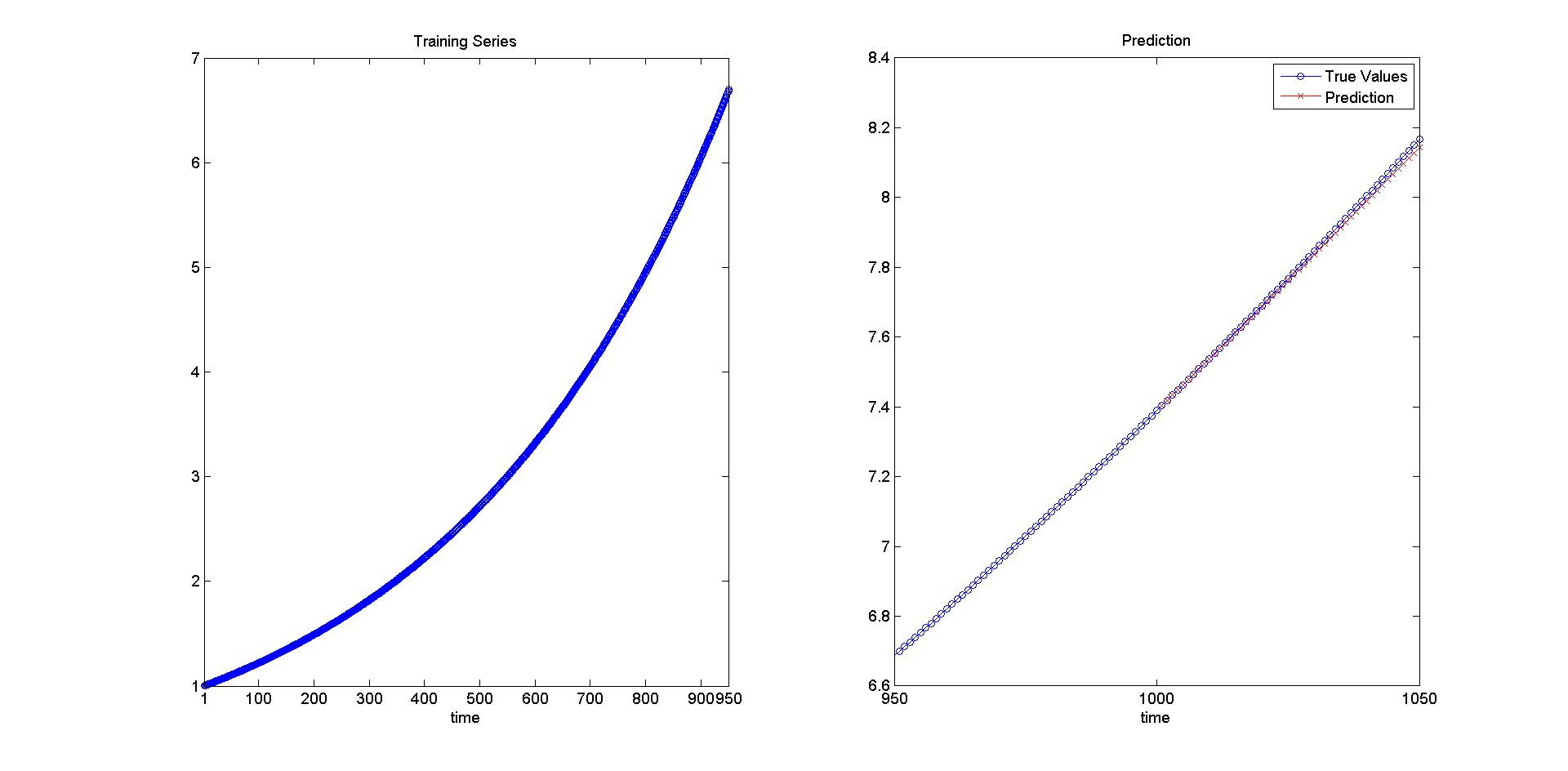
Σε όλα τα παρακάτω παραδείγματα οι παράμετροι του συστήματος έχουν τεθεί σε Nmax = 2 και Nbins = 21, ενώ η πρόβλεψη προχωρά για τα επόμενα 50 σημεία μετά το τέλος της input χρονοσειράς. Το πρόγραμμα παράγει 10 διαφορετικά (τυχαία) σενάρια εξέλιξης. Στα σχήματα που ακολουθούν, το πρώτο πάνελ (αριστερά) δείχνει την input χρονοσειρά με την οποία έχει εκπαιδευτεί το σύστημα, δηλαδή έχει παραχθεί ο πίνακας πιθανοτήτων μετάβασης, ενώ το δεύτερο δείχνει τα τελευταία 50 σημεία της input χρονοσειράς, καθώς και τα 50 επόμενα, αφενός σύμφωνα με την θεωρητική σχέση από την οποία έχει προκύψει η σειρά (με μπλε) και αφετέρου τα σημεία που προκύπτουν από τα διαφορετικά σενάρια πρόβλεψης. Να τονιστεί, ότι σε κάθε σενάριο πρόβλεψης, η χρονοσειρά προβλέπει το ένα σημείο και μετά, με βάση αυτό, το επόμενο κ.ο.κ. Έτσι ένα μικρό αρχικο λάθος στην εκτίμηση του επόμενου σημείου μπορεί να διαδοθεί και στα επόμενα και κατά συνέπεια η τελική προβλεπόμενη σειρά να απέχει αρκετά από τις πραγματικές τιμές, αλλά με αυτόν τον τρόπο γίνεται εμφανές το χρονικό εύρος ισχύος της εκάστοτε πρόβλεψης.

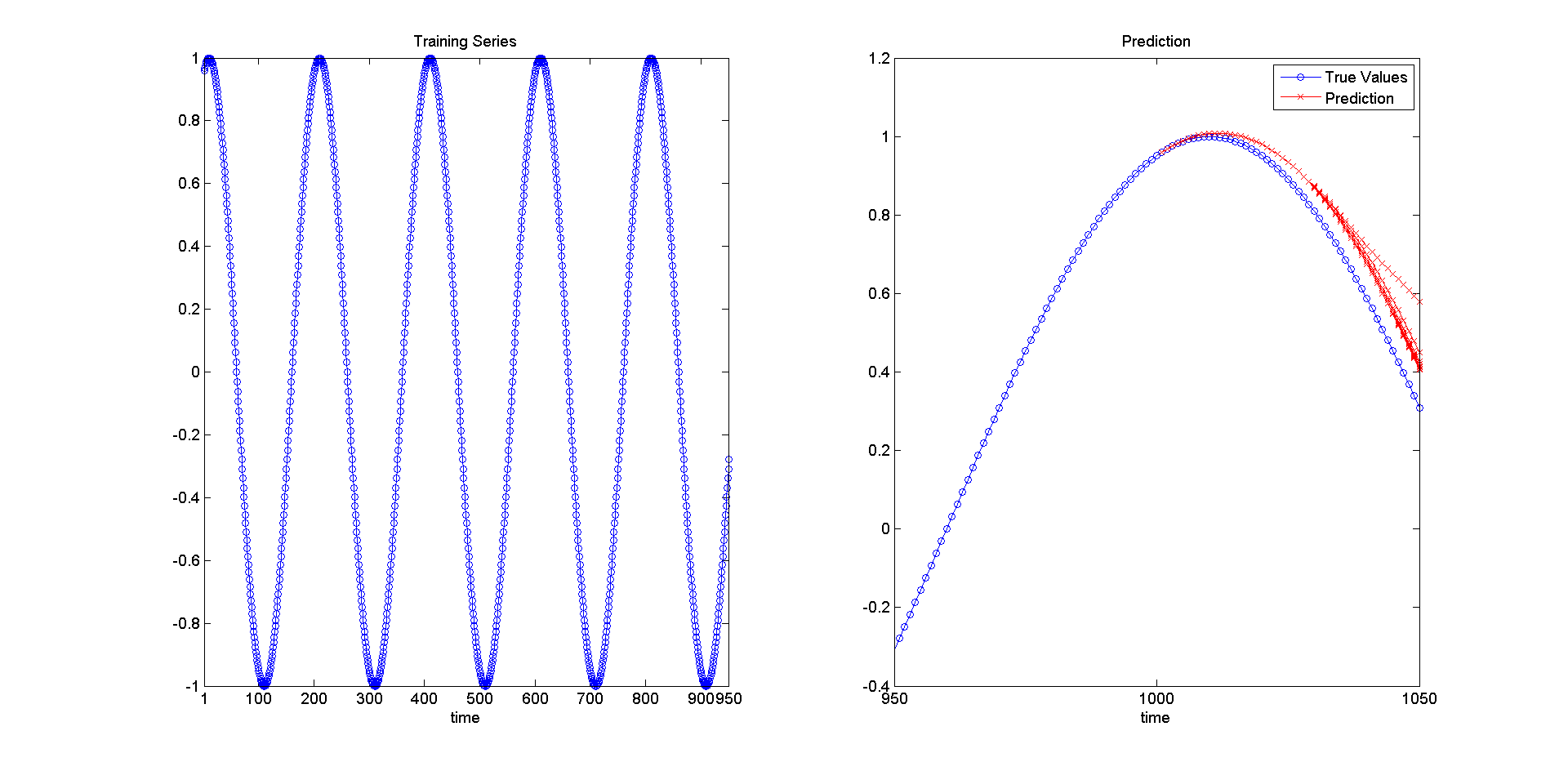
Παράδειγμα 1. Σταθερή τιμή

Παράδειγμα 2. Γραμμική περίπτωση



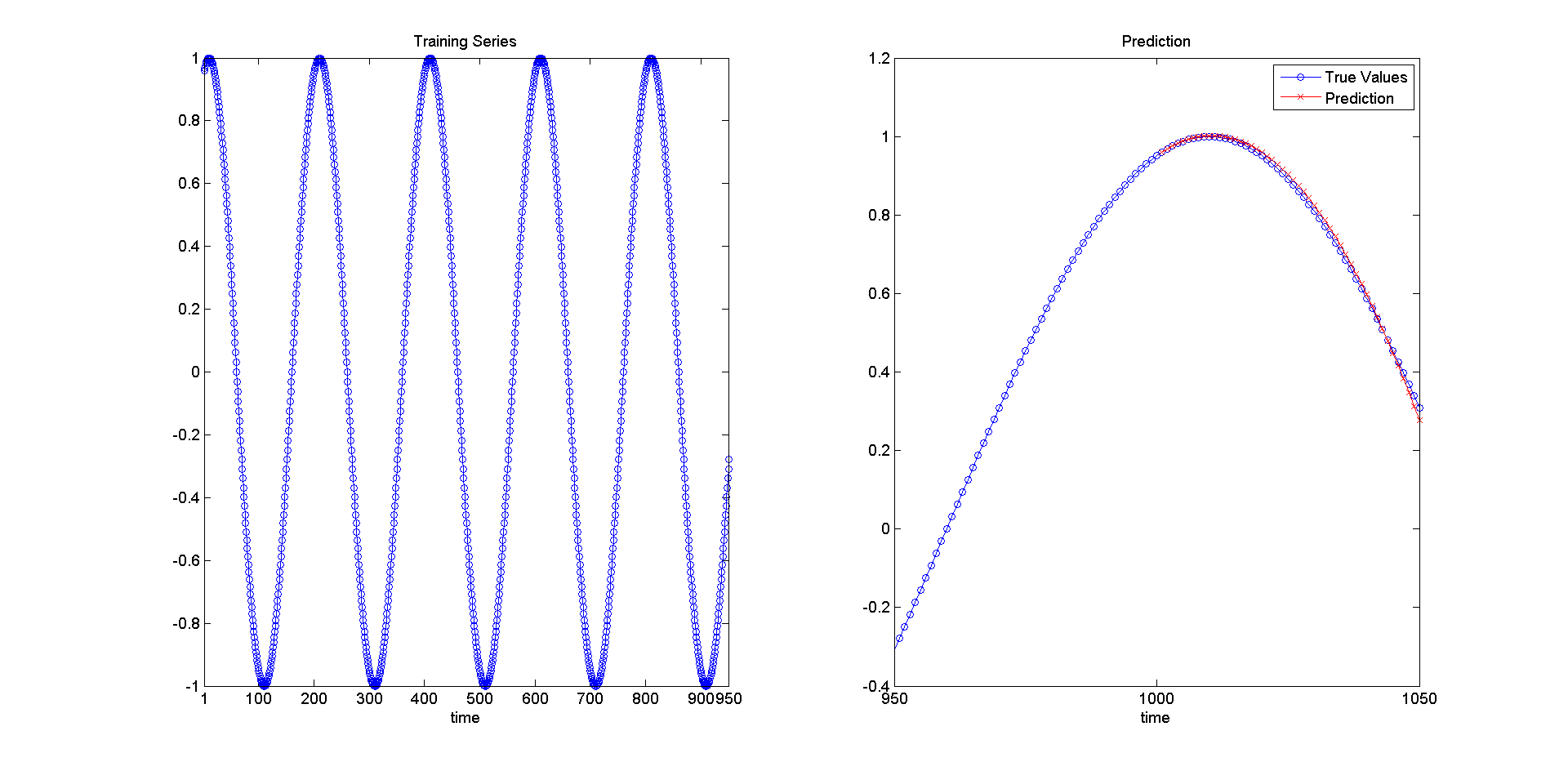
Παράδειγμα 3. Πολυώνυμο 2ης τάξης

Παράδειγμα 4. Εκθετική

Παράδειγμα 5. Περιοδική

Όπως είναι εμφανές, για όλα σχεδόν τα παραδείγματα πλην της περιοδικής περίπτωσης, η ταύτιση πρόβλεψης και πραγματικής σειράς είναι σχεδόν απόλυτη. Να τονιστεί εδώ ότι στις περιπτώσεις της γραμμικής, πολυωνυμικής και εκθετικής συνάρτησης, το πρόγραμμα προβλέπει τιμές που δεν έχει συναντήσει ποτέ πριν στο ιστορικό της input χρονοσειράς! Στην περίπτωση της περιοδικής, το εύρος της ακριβούς πρόβλεψης περιορίζεται σε μερικά μόνο σημεία, ενώ από εκεί και πέρα, το μικρό σφάλμα που υπάρχει στην αρχή διαδίδεται και στο μέλλον με αποτέλεσμα οι μετέπειτα προβλεφθείσες τιμές να εμφανίζουν μία ολοένα και αυξανόμενη απόκλιση από την πραγματική χρονοσειρά.

Προφανώς, οι όποιες αποκλίσεις μπορούν να αντιμετωπιστούν με αλλαγή των παραμέτρων του προγράμματος. Για παράδειγμα, αυξάνοντας την μέγιστη τάξη διαφόρισης από 2 στο 3, θέτοντας δηλαδή Nmax = 3 και διατηρώντας τις υπόλοιπες παραμέτρους ίδιες, η πρόβλεψη του προγράμματος για την περιοδική περίπτωση βελτιώνεται δραστικά, όπως απεικονίζεται και στο τελευταίο διάγραμμα του Παραδείγματος 6.

Παράδειγμα 6. Περιοδική (ανάλυση Nmax = 3)