MATH 4280

Lecture notes 9: Some applications

Computational Epidemiology

- Computational epidemiology is to investigate how a population interacts with a pathogen
- Modeling idea: divide the population into groups according to their disease status
- Population susceptible to acquiring a disease: group S
- Population currently infected: group I
- Population that are recovered: group R
- This is called SIR model

- An example of SIR model

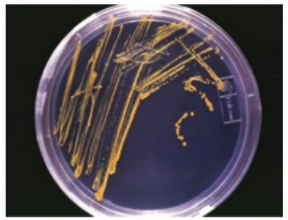
$$
\begin{aligned}
\frac{dS_i}{dt} &= -\sum_{j=1}^{n} \beta_j S_i I_j, \\
\frac{dI_i}{dt} &= \sum_{j=1}^{n} \beta_j S_i I_j - \gamma I_i, \\
\frac{dR_i}{dt} &= \gamma I_i,
\end{aligned}
$$

- Here β is an infectious parameter and γ the rate of recovery
- Note that, the number of states can be large
- For example, the index i can represent age group
- DMD can be used to recover important dynamical characteristics

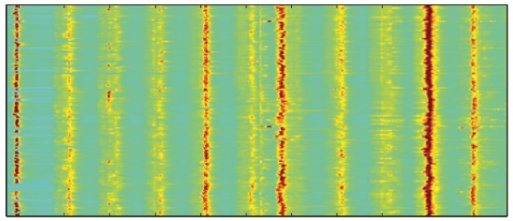
Data collection

- Snapshots can be obtained from experiment or historical records

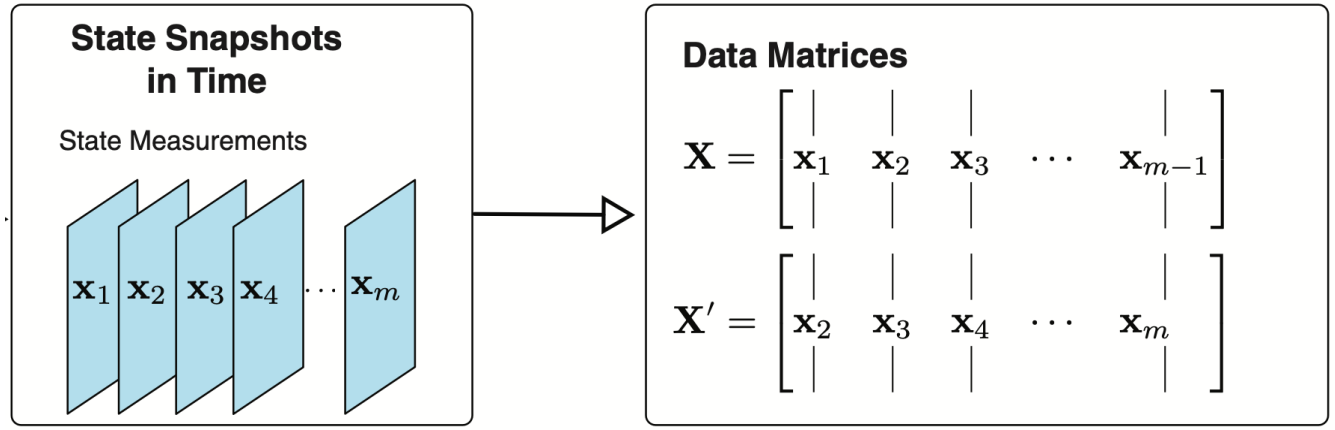
Laboratory



Historical



- Construct data matrices



Review of DMD

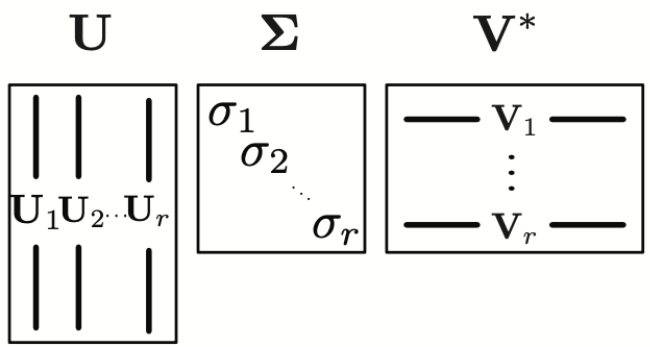
- Step 1: Compute the SVD of $\mathbf{X}$: $\mathbf{X} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$
- Step 2: Compute the projected linear operator : $\tilde{\mathbf{A}} = \mathbf{U}\mathbf{X}'\mathbf{V}\boldsymbol{\Sigma}^{-1}$
- Step 3: Compute the spectrum : $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$
- Step 4: Compute the DMD modes : $\boldsymbol{\Phi} = \mathbf{X}'\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{W}$

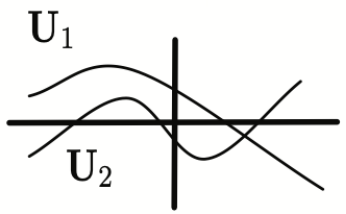
Dynamic Mode Decomposition on infectious disease data

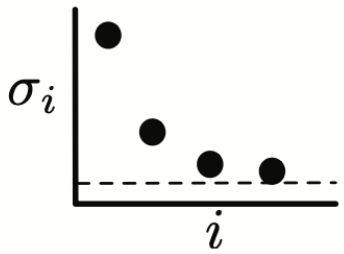
Find the dynamic properties of $\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger}$

SVD and Truncation

$$\mathbf{X} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$$



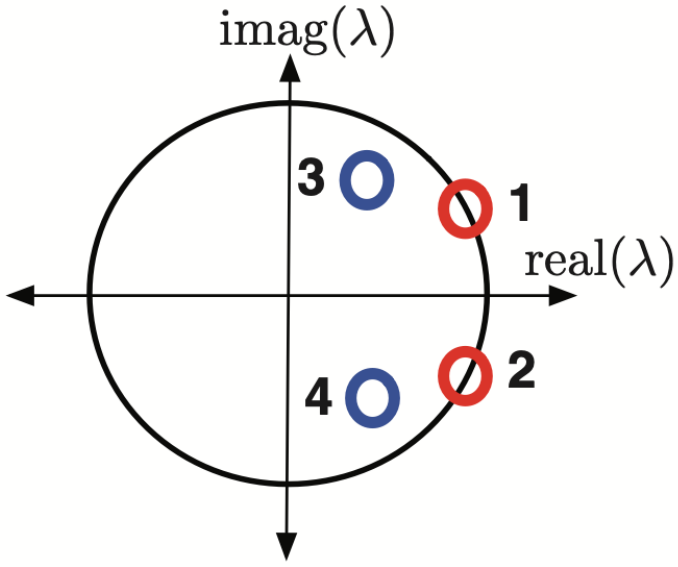




Eigenvalue Spectrum

$$\tilde{\mathbf{A}} = \mathbf{U}\mathbf{X}'\mathbf{V}\boldsymbol{\Sigma}^{-1}$$

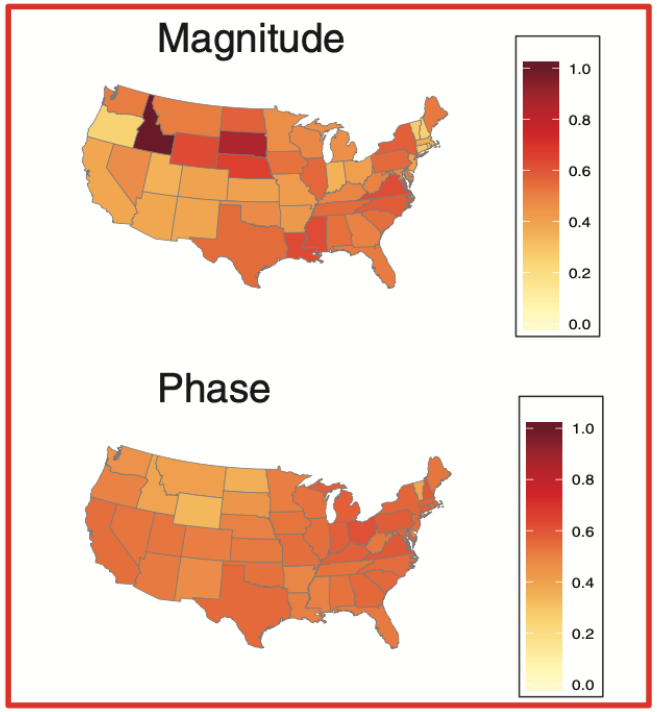
$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$$



Discrete-time eigenvalues λ

Dynamic Modes

$$\boldsymbol{\Phi} = \mathbf{X}'\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{W}$$

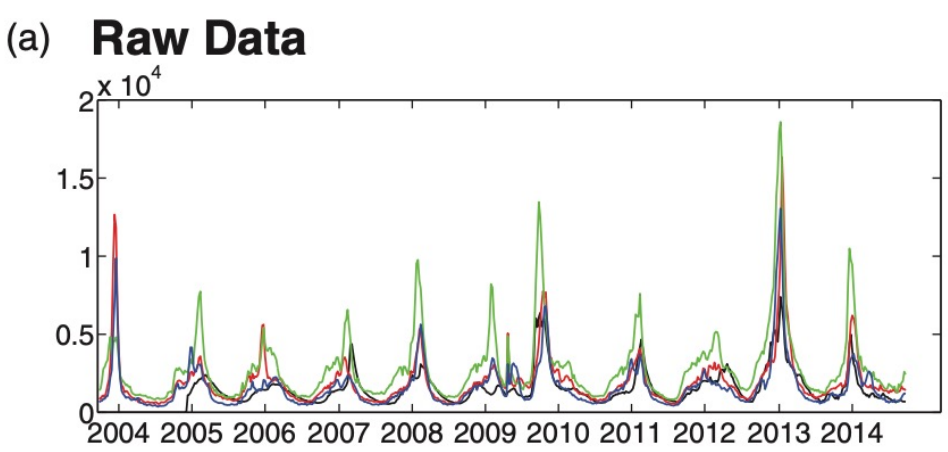


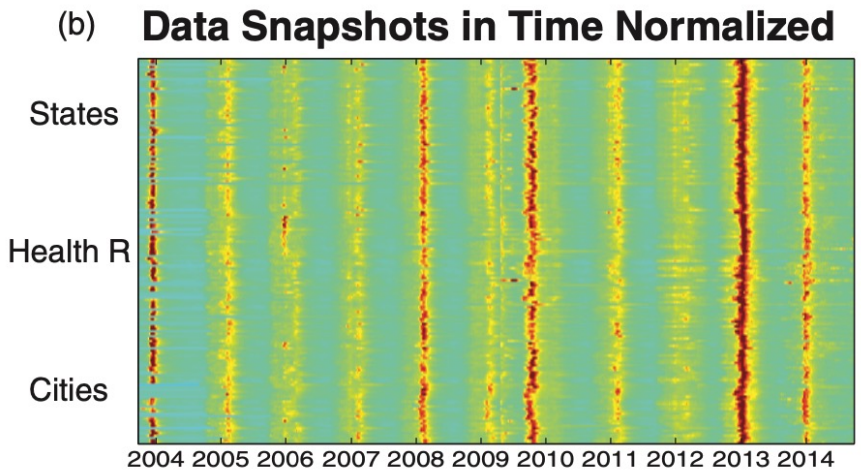
DMD results

- The eigenvalues gives dynamic characteristics such as growth/decay or oscillatory behaviour of each DMD mode
- Two pairs of eigenvalues are shown (in last page)
- The DMD mode is stable if the corresponding eigenvalue is inside the unit circle, otherwise, the DMD mode is unstable
- The absolute value of each DMD mode gives a measure of the spatial participation for that mode (see last page)

Google flu trend

- Data snapshots

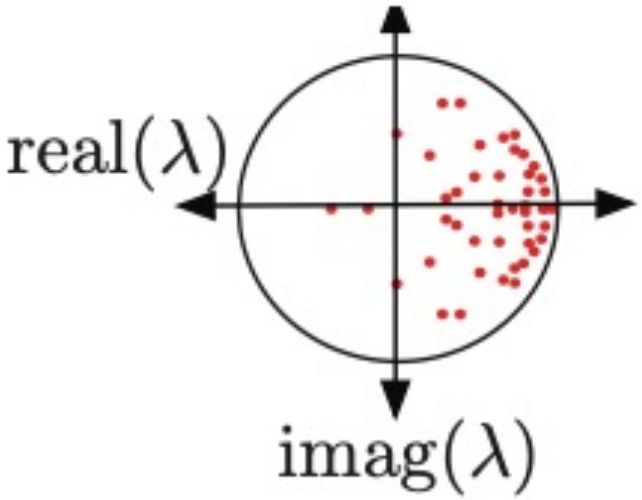




- (a) Raw data for 4 different cities
- (b) Normalization: results in zero mean and variance=1

DMD results

(d) **Discrete Eigenvalue Spectrum**



(e) **Dynamic Mode**

Algorithmic trading

- Algorithmic trading finds automated investment strategies
- It takes advantage of the speed and precision of computed-executed trading strategies that are based on mathematical structure of data
- Assume stock market is a complex dynamical system
- The DMD method can be applied to reconstruct the model
- One can update the model as the market changes in time

Data matrix

- We extract daily closing prices of some stocks
- So, we have the following data matrix

$$\mathbf{X} = \begin{array}{c} \begin{bmatrix} \cdots & \vdots & \cdots \\ \cdots & \text{‘JPM’} & \cdots \\ \cdots & \text{‘BOA’} & \cdots \\ \cdots & \text{‘AMZN’} & \cdots \\ \cdots & \text{‘CVX’} & \cdots \\ \cdots & \vdots & \cdots \end{bmatrix} \\ \longrightarrow \\ \text{daily stock prices} \end{array} \quad \downarrow \text{companies}$$

$n =$ number of companies in a given portfolio,

$m =$ number of data snapshots taken.

$\mathbf{X} \in \mathbb{R}^{n \times m}$

DMD method

- The trading algorithm is parameterized by two parameters

$$m_p = \text{number of past days of market snapshot data taken,}$$
$$m_f = \text{number of days in the future predicted.}$$

- We use the notation

$$\mathbf{x}_{\mathrm{DMD}}(m_p, m_f)$$

  to indicate the past m_p days are used to predict the future m_f days
- Aim: determine the best combination of m_p and m_f
- Trading hot spots: region of $(m_p\ , m_f)$ that gives optimal prediction

Training

- The training algorithm is an optimization procedure over the space $(m_p\ , m_f)$
- Find the prediction $\mathbf{x}_{\mathrm{DMD}}(m_p, m_f)$ having the highest rate of success
- We let $m_p = 1,2,\ldots,25$ and $m_f = 1,2,\ldots,10$
- Use historical data in a two-year period
- Trading hot spot is defined using the success rate S:

$$\text{(i)}\ \ S_{m_p,m_f} > 0.53, \quad \text{(ii)}\ \ \frac{1}{9}\sum_{j=-1}^{1}\sum_{k=-1}^{1} S_{m_p+k,m_f+j} > 0.53$$

Examples of trading hot spots

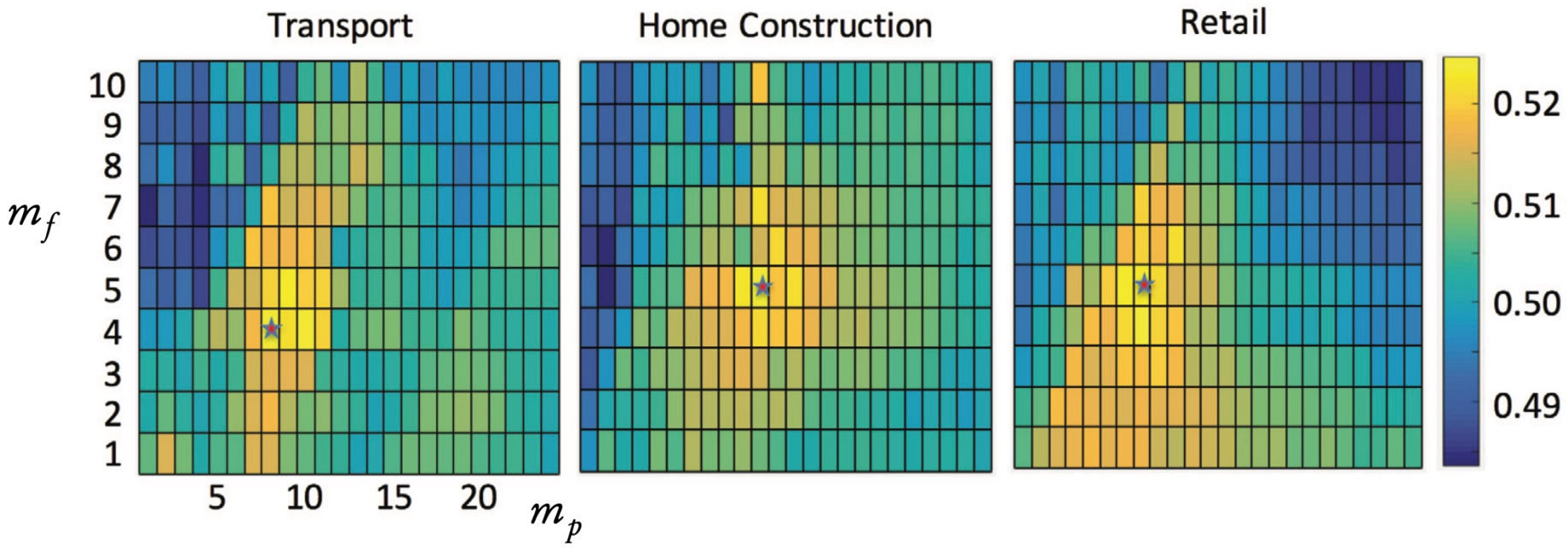


Figure 13.1. *Sectors that have a hot spot where the probability of success at the optimal trading position for* (m_p, m_f) *is above* 53% *along with its surrounding eight cells. Thus, the definition* (13.2) *is satisfied in the transport, home construction, and retail sectors. Note that the color in each cell denotes the probability of success.*

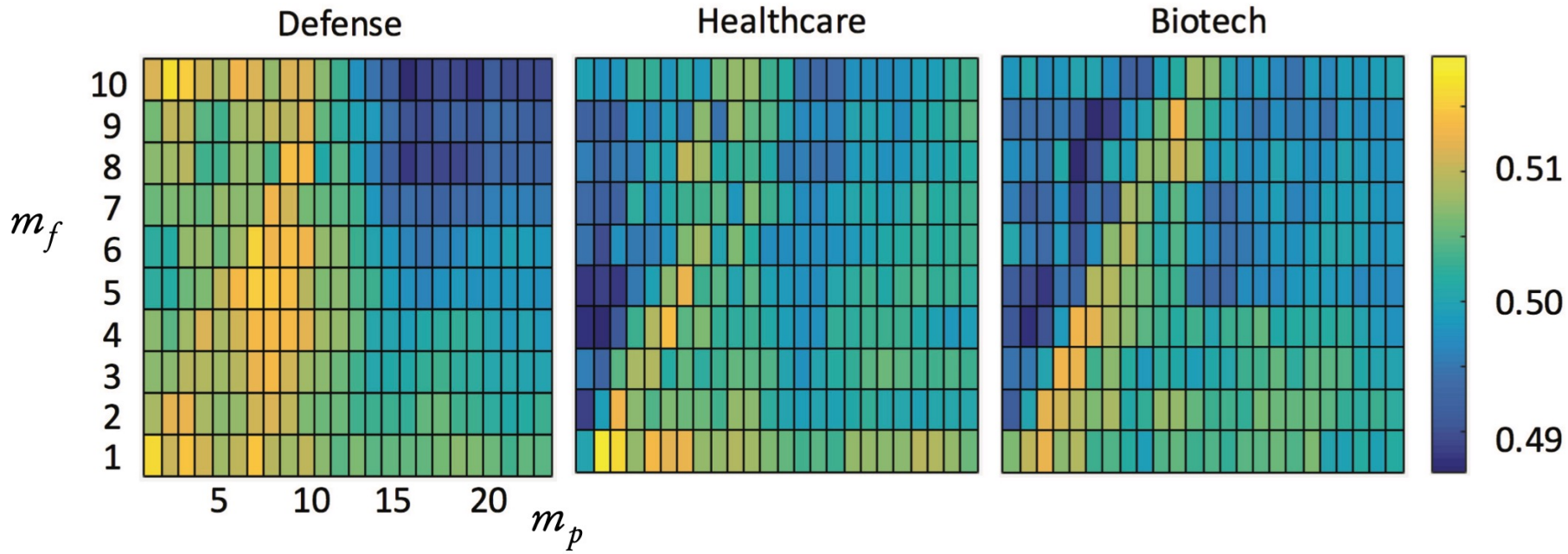
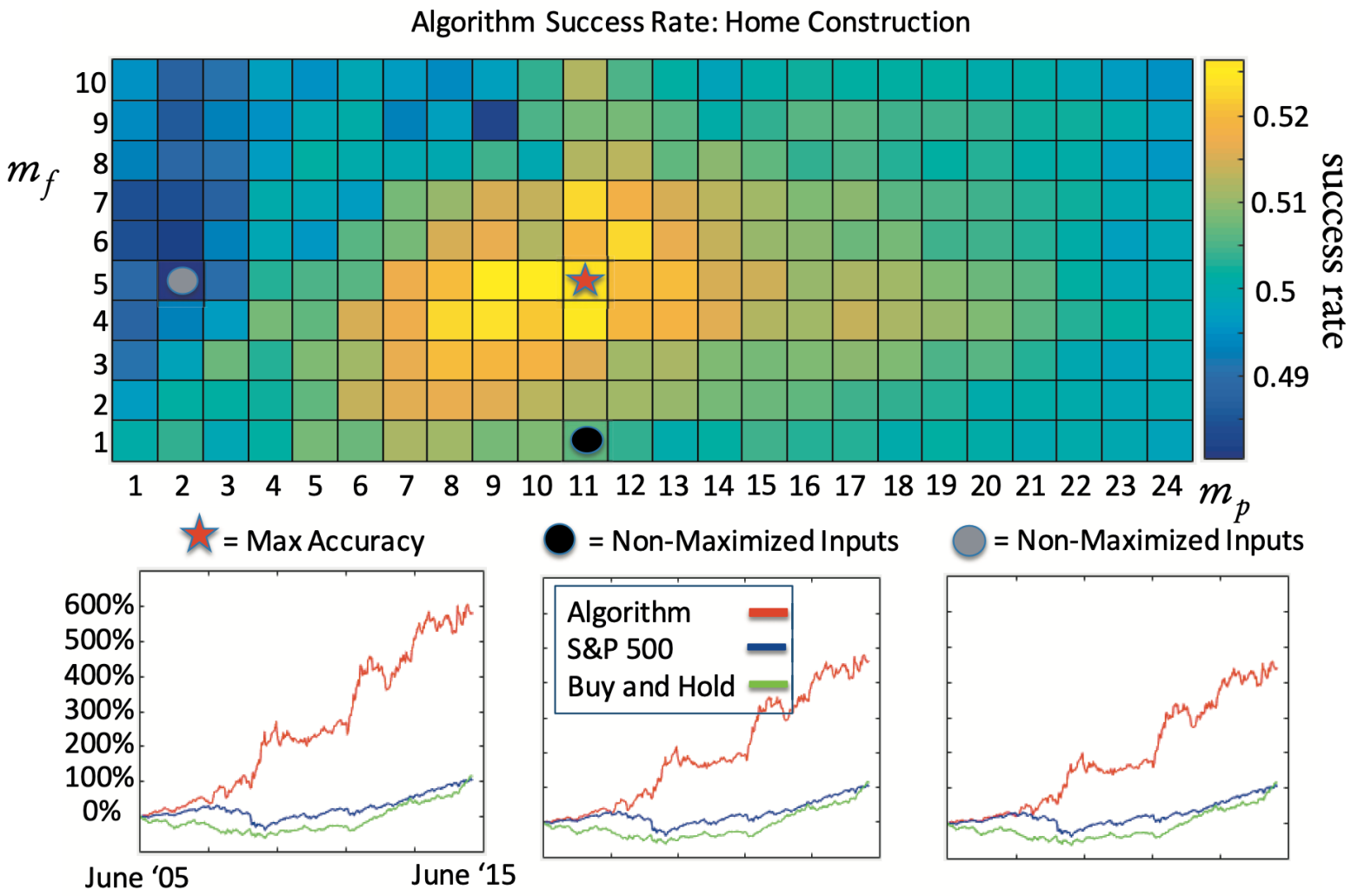


Figure 13.2. *Sectors that do not have a hot spot. Although individual cells may be above 53% in predictive accuracy, these sectors fail to have the surrounding eight cells above the threshold in* (13.2). *These sectors include defense, healthcare, and biotech. Note that the color in each cell denotes the probability of success.*

Performance



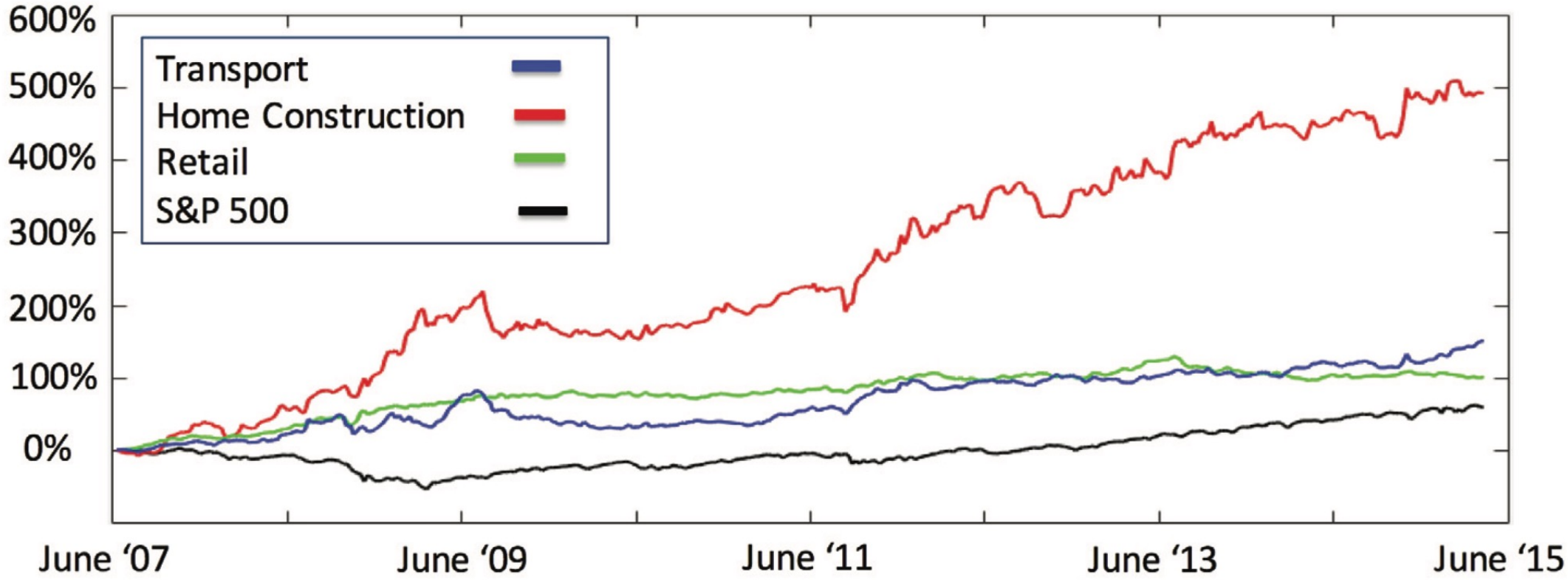


Figure 13.5. *Trading success of Algorithm I of the DMD method for three sectors exhibiting hot spots (transport, home construction, and retail). The optimal values of* (m_p, m_f) *are determined during a training period of two years (July* 2005 *to July* 2007*). Once determined, the hot spot trading window is used to execute trades from July* 2007 *onwards. When a hot spot is present, the algorithm can significantly increase financial gains over investing and holding in the S&P* 500.

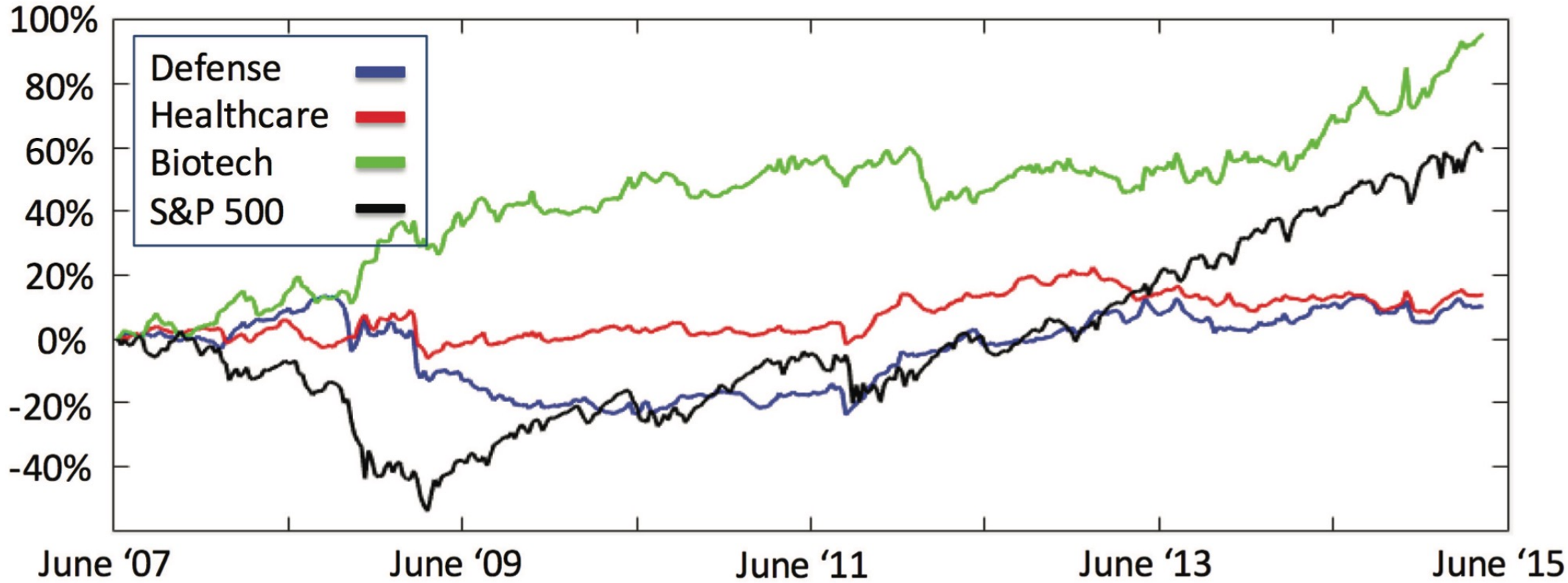


Figure 13.6. *Demonstration of trading success of Algorithm I of the DMD method for three sectors that do not exhibit a hot spot (defense, healthcare, and biotech). The optimal value of* (m_p, m_f) *is still determined during a training period of two years (July* 2005 *to July* 2007*). Once determined, the trading window is used to execute trades from July* 2007 *onwards. When there is no hot spot present, the algorithm performs no better than investing and holding in the S&P* 500.

Video processing

- Background/foreground separation is important in detecting, identifying, tracking and recognizing objects in video
- Need real time and robust algorithms
- One viewpoint: separate a matrix (video) into low rank (background) and sparse (foreground) components
- This can be done using RPCA
- Another viewpoint: the video frames are considered as snapshots of some underlying dynamical systems
- Applying the method of DMD

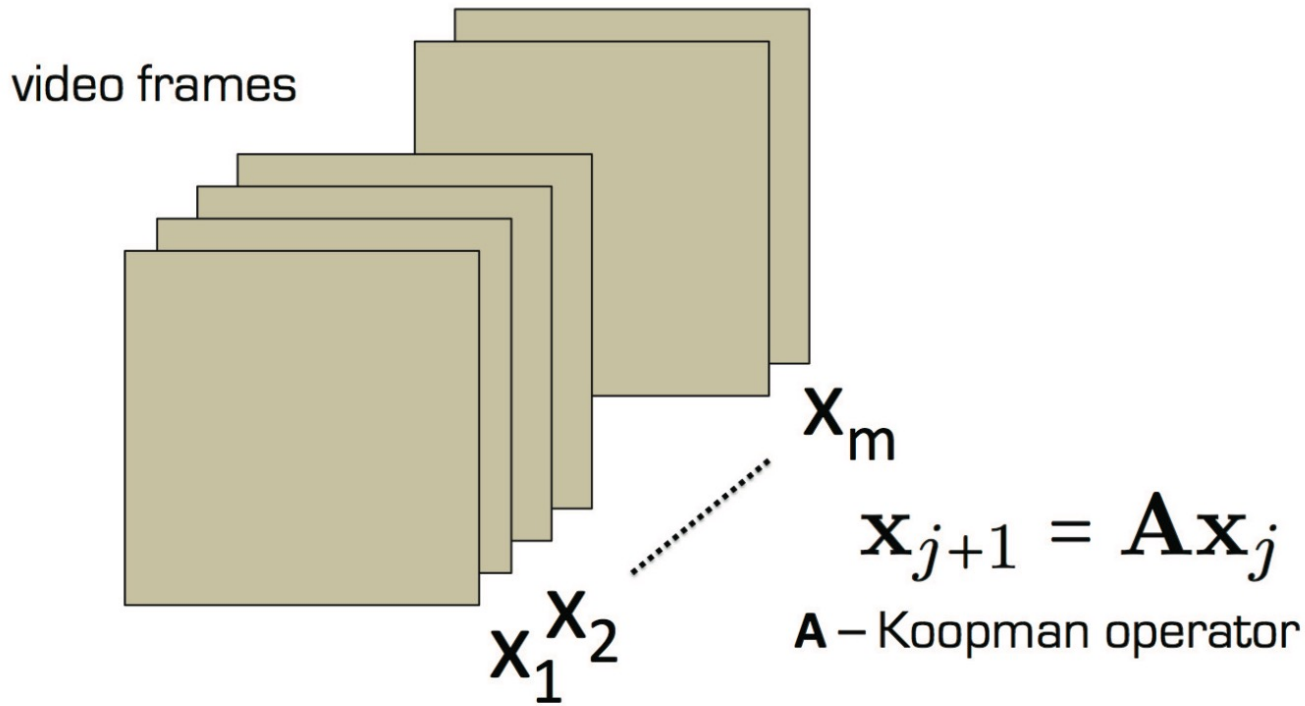


Figure 4.1. *Illustration of the DMD method where snapshots (pixels of video frames)* $\mathbf{x}_k$ *are vectorized and a linear transformation* $\mathbf{A}$ *is constructed. The DMD method constructs the best matrix* $\mathbf{A}$ *that minimizes the least-square error for all transformations* $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$ *with* $k = 1, 2, \ldots, m-1$.

RPCA

- The goal is to decompose the matrix $X = L + S$, where L has low rank and S is sparse
- This problem can be formulated as

$$\begin{aligned} \arg\min \ & \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ \text{subject to} \ & \mathbf{X} = \mathbf{L} + \mathbf{S}. \end{aligned}$$

- where the nuclear norm $\|M\|_* = trace\sqrt{M^* M}$
- The scalar regularization parameter λ is non-negative
- It is shown that a good choice is given by $\lambda = \frac{1}{\sqrt{\max(n, m)}}$

RPCA interpretation of DMD

- Recall that DMD approximates solution by

$$\mathbf{x}(t) \approx \sum_{k=1}^{r} \boldsymbol{\phi}_k \exp(\omega_k t) b_k$$

- Low rank features in video can be defined using $|\omega_j| \approx 0$
- One can set a threshold to define low rank modes by $|\omega_j| \leq \varepsilon$
- This gives

$$\mathbf{L} \approx \sum_{|\omega_k| \leq \epsilon} b_k \boldsymbol{\phi}_k \exp(\omega_k t),$$

$$\mathbf{S} \approx \sum_{|\omega_k| > \epsilon} b_k \boldsymbol{\phi}_k \exp(\omega_k t).$$

DMD for video processing

- Given a video with m frames, the $n_x \times n_y = n$ pixels in each frame are denoted by the $n \times 1$ vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$
- The video sequence is reconstructed using DMD as follows

$$\mathbf{X}_{\mathrm{DMD}} = \sum_{k=1}^{r} b_k \boldsymbol{\phi}_k e^{\omega_k \mathbf{t}} = \boldsymbol{\Phi} \mathrm{diag}(\exp(\omega t))\mathbf{b}.$$

- Note that $\boldsymbol{\phi}_k$ is an $n \times 1$ vector
- Note also that $\mathbf{x}_1 = \boldsymbol{\Phi}\mathbf{b}$
- We see that the diagonal matrix ω determines how the first frame gets altered in time

- Assume that only a single mode changes slowly in time, $||\omega_p|| \approx 0$
- Then we have

$$\mathbf{X}_{\mathrm{DMD}} = \underbrace{b_p \boldsymbol{\phi}_p e^{\omega_p \mathbf{t}}}_{\text{Background Video}} + \underbrace{\sum_{k \neq p} b_j \boldsymbol{\phi}_j e^{\omega_k \mathbf{t}}}_{\text{Foreground Video}}.$$

- One difficulty: the terms $b_k \boldsymbol{\phi}_k \exp(\omega_k \mathbf{t})$ are complex in general
- First, we define the DMD low rank mode by

$$\mathbf{X}_{\mathrm{DMD}}^{\text{Low-Rank}} = b_p \boldsymbol{\phi}_p e^{\omega_p \mathbf{t}}.$$

- Note that

$$\mathbf{X} = \mathbf{X}_{\mathrm{DMD}}^{\text{Low-Rank}} + \mathbf{X}_{\mathrm{DMD}}^{\text{Sparse}}$$

- Recall the DMD sparse mode is

$$\mathbf{X}_{\mathrm{DMD}}^{\mathrm{Sparse}} = \sum_{k \neq p} b_k \boldsymbol{\phi}_k e^{\omega_k \mathbf{t}}$$

- The sparse mode can be computed with real-valued elements by

$$\mathbf{X}_{\mathrm{DMD}}^{\mathrm{Sparse}} = \mathbf{X} - \left| \mathbf{X}_{\mathrm{DMD}}^{\mathrm{Low\text{-}Rank}} \right|$$

- The above may result in negative elements, which are put in matrix R
- Then

$$\mathbf{X}_{\mathrm{DMD}}^{\mathrm{Low\text{-}Rank}} \leftarrow \mathbf{R} + \left| \mathbf{X}_{\mathrm{DMD}}^{\mathrm{Low\text{-}Rank}} \right|,$$

$$\mathbf{X}_{\mathrm{DMD}}^{\mathrm{Sparse}} \leftarrow \mathbf{X}_{\mathrm{DMD}}^{\mathrm{Sparse}} - \mathbf{R}.$$

Figure 4.3. *The DMD background/foreground separation results are illustrated for three specific frames in the "Parked Vehicle" video. The 30-frame video segment that contains frame 500 has three vehicles driving in the same lane toward the camera. The 30-frame video segment that contains frame 1000 has a man stepping up to a crosswalk as a vehicle passes by, and a second car in the distance, barely perceptible, starts to come into view. The 30-frame video segment that contains frame 2000 has three vehicles stopped at a traffic light at the bottom of the frame, with another two vehicles parked on the right side of the road, and five moving vehicles, two going into the distance and three coming toward the vehicles waiting at the light, the last vehicle being imperceptible to the eye at this pixel resolution:* $n = 11520$.

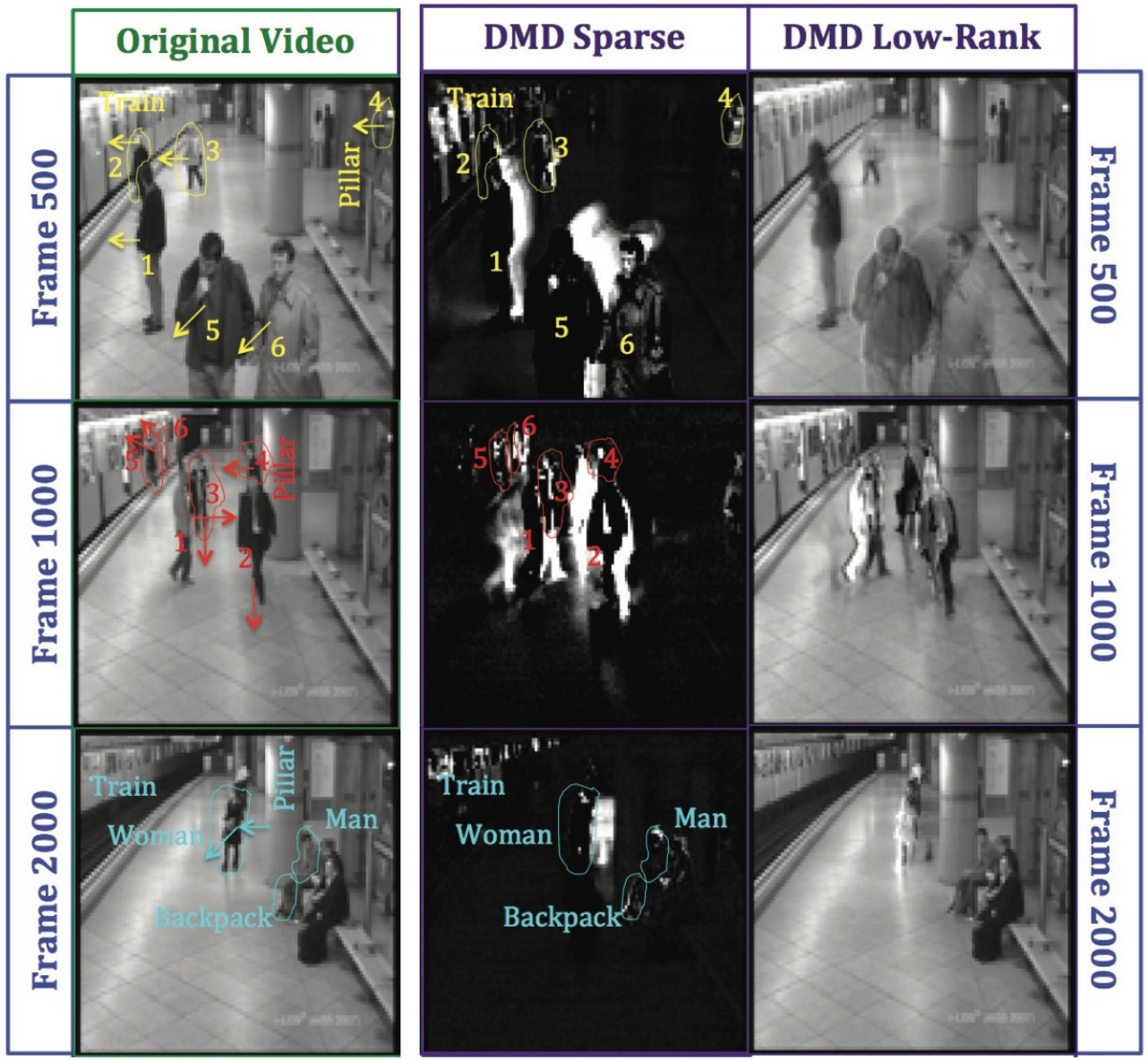


Figure 4.4. *The DMD background/foreground separation results are illustrated for three specific frames in the "Abandoned Bag" video. The 30-frame video segment that contains frame 500 has three people stepping slightly closer to an arriving train, and another person walking behind a support pillar toward the train in the upper right area of the frame. The two people closest to the camera walk toward the bottom of the frame. The 30-frame video segment that contains frame 1000 has four people walking in different directions in the middle of the frame, one of whom comes out from behind a support pillar, while, farther down the platform, two people enter the train. The 30-frame video segment that contains frame 2000 has a woman walk out from behind a support pillar, move to the left, and then turn somewhat toward the camera. The train is moving and the man sitting closest to the support pillar adjusts his backpack.*