Networks - Week 5 - Centrality in Continuous Time & Dynamics on Networks

Antonio León Villares

November 2023

Contents

1 Katz Centrality in Continuous Time **2**
1.1 Matrix-Valued Functions . . . 2
1.1.1 Definition: Jordan Normal Form . . . 2
1.1.2 Definition: Matrix Valued Function . . . 3
1.1.3 Definition: Rational Function . . . 5
1.1.4 Definition: Principal Logarithm of a Matrix . . . 5
1.1.5 Definition: Principal pth Root . . . 6
1.1.6 Definition: Principal Power . . . 6
1.1.7 Exercises . . . 7
1.2 Katz Centrality with Age Discounting . . . 7
1.2.1 Definition: Katz Centrality Matrix with Age Discounting . . . 8
1.3 Katz Centrality for Continuous Time . . . 9
1.3.1 Proposition: ODE for Katz Centrality Matrix . . . 9

2 Dynamics on Networks **11**
2.1 Notation . . . 11
2.2 Consensus Dynamics . . . 12
2.2.1 Definition: Average Consensus Dynamics . . . 12
2.2.2 Behaviour of Average Consensus Dynamics . . . 12
2.3 Time-Scale Separation in Dynamical Systems . . . 13
2.3.1 Definition: Time-Scale Separation in ODEs . . . 14
2.3.2 Exploiting Time-Scale Separation for System Decoupling . . . 14
2.4 Time-Scale Separation in Consensus Dynamics . . . 15
2.4.1 Proposition: Solution to Average Consensus Dynamics in Terms of Eigenvectors and Eigenvalues . . . 15
2.4.2 Analysing Long and Short Term Consensus . . . 15

Based on the notes by Renaud Lambiotte and Pete Grindrod, Chapters 6 & 9

1 Katz Centrality in Continuous Time

1.1 Matrix-Valued Functions

When defining Katz Centrality for continuous networks, we need to be able to make sense of matrix valued functions, for arbitrary functions. We've already seen some examples of these, like A^2 or $\exp(A)$. We further generalise these to even consider multi-valued functions, such as fractional powers, roots or logarithms.

1.1.1 Definition: Jordan Normal Form

*Every $n \times n$ matrix A with **complex** entries can be written in **Jordan Normal/Canonical Form**:*

$$A = ZJZ^{-1}$$

where:

- *Z is some **invertible matrix***
- *J is a **block diagonal matrix**, composed of **Jordan Blocks** J_i, which is **unique** up to rearrangement of the J_i*

---

*Let λ_i be an **eigenvalue** of A, with **algebraic multiplicity** n_i. A **Jordan Block** J_i is a $n_i \times n_i$ **square matrix**:*

$$J_i = \lambda_i I_{n_i} + N_{n_i} = \begin{pmatrix} \lambda_i & 1 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & \dots & 0 \\ 0 & 0 & \lambda_i & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_i \end{pmatrix}$$

where N_{n_i} is a matrix containing 1s in the off-diagonal.

- **What identity do the algebraic multiplicites satisfy?**

  –

$$n = \sum_i n_i$$

- **What is the Jordan Normal Form of a diagonal/diagonalisable matrix?**

- a **diagonal matrix**
- this corresponds to a **Jordan Matrix** J composed of 1×1 **Jordan Blocks**
- each **Jordan Block** corresponds to a **single, unique eigenvalue**
- this corresponds to the fact that an $n \times n$ matrix is **diagonalisable** if and only if it has n **distinct eigenvalues**

1.1.2 Definition: Matrix Valued Function

This is a way of defining matrix valued functions - there are many others.

Let:

- *A be an $n \times n$ matrices, whose **eigenvalues** $\{\lambda_i\}_{i \in [1,r]}$ have **algebraic multiplicity** $\{n_i\}_{i \in [1,r]}$*
- *f be a **complex function** defined on each of the λ_i, and such that $\forall i \in [1, r]$ each of the derivatives of f (up to the n_ith derivative) are defined on λ_i*

*Then, if A has **Jordan Canonical Form**:*

$$A = ZJZ^{-1}, \qquad J = \operatorname{diag}(J_1, \ldots, J_r)$$

we define the matrix $f(A)$ via:

$$f(A) = Z \operatorname{diag}(f(J_1), \ldots, f(J_r))$$

where:

$$f(J_i) = \begin{pmatrix} f(\lambda_i) & f^{(1)}(\lambda_1) & \frac{f^{(2)}(\lambda_i)}{2!} & \ldots & \frac{f^{(n_i-1)}(\lambda_i)}{(n_i-1)!} \\ 0 & f(\lambda_i) & f^{(1)}(\lambda_1) & \ldots & \frac{f^{(n_i-2)}(\lambda_i)}{(n_i-2)!} \\ 0 & 0 & f(\lambda_i) & \ldots & \frac{f^{(n_i-3)}(\lambda_i)}{(n_i-3)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & f(\lambda_i) \end{pmatrix}$$

- **What justifies the above definition of a matrix valued function?**
  - recall, we can write a **Jordan Block** as:

$$J_i = \lambda_i I_{n_i} + N_{n_i}$$

  - it is "standard" to define a function on a diagonal matrix by evaluating the function at each diagonal element, so:

$$f(\lambda_i I_{n_i}) = f(\lambda_i) I_{n_1} = \operatorname{diag}(f(\lambda_i), \ldots, f(\lambda_i))$$

- moreover, taking powers of N_{n_i} "moves" the off-diagonal of 1s up 1 diagonal. For example:

$$N_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \implies N_3^2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- in particular, this implies that:
  1. $$N_{n_i}^{n_i} = \mathbf{0}$$
  2. if X is any square matrix, then:
  $$XN_{n_i}^k$$
  is a matrix where the diagonal elements of X have been shifted into the kth diagonal
- finally, the **Taylor Expansion** of a function f evaluated at some z and centered at λ_i is given by:

$$f(z) = \sum_{j=0}^{\infty} \frac{f^{(j)}(\lambda_i)}{j!}(z - \lambda_i)^j$$

- thus, if we apply a **Taylor Expansion** for f evaluated at $\lambda_i I_{n_i} + N_{n_i}$ and centered at $\lambda_i I_{n_i}$ we get that:

$$\begin{aligned} f(J_i) &= \sum_{j=0}^{n_i-1} \frac{f^{(j)}\left(\lambda_i I_{n_i}\right)}{n_i!}\left(J_i - \lambda_i I_{n_i}\right) \\ &= f(\lambda_i)I_{n_i} + f^{(1)}(\lambda_i)N_{n_i} + \ldots + \frac{f^{n_i-1}(\lambda_i)}{(n_i-1)!} \end{aligned}$$

  which is precisely how $f(J_i)$ was defined above

- **Do we need to compute the Jordan Normal Form to compute functions of matrices for every function?**
  - for some functions, the definition is straightforward
  - for **powers**, we just use matrix powers
  - for **polynomials**, we just "evaluate" the polynomial:

$$f(t) = t^3 + t \implies f(A) = A^3 + A$$

  - for **exponentials**, we have the **Taylor Series** expansion:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

  However, note that:

$$e^{A_1+A_2} = e^{A_1}e^{A_2} \iff A_1A_2 = A_2A_1$$

- **How can we relax the requirements for matrix valued functions for adjacency matrices of undirected graphs?**
  - for $f(A)$ to be defined on an **abitrary** matrix A, f and its derivatives must be defined on the **spectrum** of A
  - however, if A is an **adjacency matrix** it is by definition **real** and **symmetric**, so in particular, A is **diagonalisable** and has **real eigenvalues**
  - in particular, the **Jordan Canonical Form** of A will have J be **diagonal**
  - hence, for **adjacency matrices**, $f(A)$ is defined **if and only if** f is defined on each of the **eigenvalues** of A

1.1.3 Definition: Rational Function

Let

$$f(z) = (1 - \alpha z)^{-1}, \qquad \alpha \in \mathbb{R}^+$$

*Then, f is **well-defined** as a **matrix-valued function** on **adjacency matrices of undirected graphs** if:*

$$\frac{1}{\alpha} > \rho(A) > 0$$

*where $\rho(A)$ is the **spectral radius** of A.*

- **What motivates the above definition?**
  - $f(z)$ is undefined whenever $z = \frac{1}{\alpha}$
  - by the **Perron-Frobenius Theorem**, all **eigenvalues** of A are contained within a **disc** of radius $\rho(A)$
  - hence, if

    $$\frac{1}{\alpha} > \rho(A)$$

    we are guaranteed that f is well-defined on the spectrum
  - notice, this requirement is equivalent to assuming that:

    $$\alpha < \frac{1}{\rho(\alpha)}$$

    which is precisely the requirement for the **Katz Centrality** to be defined (so that the **geometric series** defining it converges)

1.1.4 Definition: Principal Logarithm of a Matrix

*Let A be a matrix. If A has no **eigenvalues** on $\mathbb{R}^-$ (0 and all the negative real numbers), then we define the **principal logarithm** of A as the **unique** matrix B such that:*

- $$B = \log(A)$$
- *for every **eigenvalue** μ_i of B we have that:*

$$\arg(\mu_i) \in (-\pi, \pi)$$

- **Why do we require that the eigenvalues be all positive for the logarithm to be defined?**
  - logarithms of **negative numbers** are well-defined as **complex numbers** (for example, the **principal logarithm** of -1 is $i\pi$, since $e^{i\pi} = -1$ and $\arg(i\pi) = \frac{\pi}{2}$)
  - however, allowing such complex values means that the **convergence** of the **exponential matrix** will be **ill-defined**
  - as such, we require that eigenvalues are strictly positive
- **What is the purpose of having a principal logarithm?**
  - the **logarithm** is a **multi-valued function**
  - the **principal logarithm** allows us to talk about a **unique** logarithm, by restricting the **spectrum** of A
  - for example, if A is the matrix of 2 dimensional **rotations** by an angle α:

    $$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

    then its logarithm is a set:

    $$B_n = (\alpha + 2\pi n) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

    This makes sense, since **rotations** can't be uniquely described (they are 2π-periodic)

1.1.5 Definition: Principal pth Root

*Let $p > 0$ be an **integer**. The **principal pth root** of a matrix A is the **unique** matrix*

$$B = A^{\frac{1}{p}}$$

such that:

- $$B^p = A$$
- *for every **eigenvalue** μ_i of B we have that:*

$$\arg(\mu_i) = \left(-\frac{\pi}{p}, \frac{\pi}{p} \right)$$

1.1.6 Definition: Principal Power

*Let $s \in \mathbb{R}$. The **principal power** of a matrix A is the **unique** matrix*

$$B = A^s = e^{s \log(A)}$$

*where $\log(A)$ is the **principal logarithm** of A.*

1.1.7 Exercises

1. **Let A be the adjacency matrix for an undirected graph. Then A is diagonalisable. How many square roots does A have (that is, matrices B for which $B^2 = A$)' What about fractional powers of A:**
$$A^\beta, \quad \beta \in [0,1]$$

2. **Let A be the adjacency matrix for an undirected graph. Then, we can write $A = U\Lambda U^T$, where U is a unitary matrix containing the eigenvectors of A as columns, a dn Λ is a real diagonal matrix of eigenvalues of A. Show taht for all polynomials P there exists a polynomial R of degree less than n such that $P(A) = UR(\Lambda)U^T$, where**
$$R(\Lambda) = \text{diag}(R(\lambda_1), \ldots, R(\lambda_n))$$

1.2 Katz Centrality with Age Discounting

In Week 3 we defined the Katz Centrality for dynamic, evolving networks:

> *Let $\{A_k\}$ be an **evolving network**. Its **Katz Centrality Matrix** is given by:*
> $$Q = \prod_{i=1}^{K} (I - \alpha A_1)^{-1}$$

We now further develop the notion of the Katz Centrality Matrix to incorporate age discounting.

- **How can the Katz Centrality Matrix be updated at a new time step?**
  - say we have a sequence of K **adjacency matrices** $A_1, \ldots, A_K$
  - thinking of each matrix as being "sampled" at some interval of length δt, we can define $T = K\delta t$, such that:
  $$Q(T) = \prod_{i=1}^{K} (I - \alpha A_i)^{-1}$$
  - then, we can define an **update** to the matrix, given that we observe a new adjacency matrix A_{K+1} during the interval $(K\delta t, (K+1)\delta t]$ via:
  $$Q(T + \delta t) = Q(T)(I - \alpha A_{K+1})^{-1}$$
- **What is age discounting, and why should it be used for the Katz Centrality Matrix?**
  - **age discounting** is a way of **reducing** the influence of **older events** A_i on the **Katz Centrality Matrix**
  - this means that Q will pay more attention/be more representative of more recent **adjacency matrices**, and thus, will give a better idea of which **nodes** have become more central towards the end of the **network evolution process**
  - this means that Q will be more "realistic" in portraying the dynamics of information flow in a network

---

1.2.1 Definition: Katz Centrality Matrix with Age Discounting

*Let $b > 0$ be a **discounting rate**. Then, updating the **Katz Centrality Matrix with Age Discounting** is given by:*

$$Q(T + \delta t) = (I + e^{-b\delta t}(Q(T) - I))(I - \alpha A_{K+1})^{-1}$$

where we use the fact that:

$$Q = I + (Q - I)$$

*and $Q - I$ is a term containing **non-trivial paths** (walks with **at least** one edge from some past time step) which is what we want to discount.*

- **How does the discounting affect the influence of certain paths?**
  - recall, the **Katz Centrality Matrix** had the following properties:

1. *Entry ij of Q counts **all** possible walks of **all** possibe combinations of **edges** taken from successive time steps that can be taken across time in the evolving network.*
2. *Q is only defined when:*

$$\forall k \in [1, K], \quad \alpha < \frac{1}{\rho(A_k)}$$

3. *Q is generally non-symmetric*

  - in particular, for a particular **path** of length m, starting r time steps ago, their influence is discounted by a factor of:

$$\alpha^m \left(e^{-b\delta t}\right)^r = \alpha^m e^{-rb\delta t}$$

- **What is the effect of setting the discounting rate to 0?**
  - in such a case, we default to the **standard** update rule
  - all walk history is counted equally, which won't be useful for quickly evolving networks
- **What is the effect of setting b to be large?**
  - as $b \to \infty$,

$$Q(T + \delta t) \to (I - \alpha A_{K+1})^{-1}$$

  - our understanding of the **dynamics** of the **evolving network** will depend solely on the current network status (there is no memory)

- in this sense, we have a **Markov**-like assumption, whereby future forecasts depend only on the current state

- **Why is tuning b important?**
  - depending on what **future features** we want to forecast with Q, it has been shown that b can be **optimised** for these forecasting tasks

1.3 Katz Centrality for Continuous Time

Thus far, we've focused on network evolution over discrete time steps δt. However, this misses out on a lot of "detail" regarding the dynamics (i.e certain edges might occur more frequently if we consider smaller steps). To deal with this, we consider the continuous Katz Centrality, which considers what happens when $\delta t \to 0$.

1.3.1 Proposition: ODE for Katz Centrality Matrix

We can define an ODE which encodes the dynamics of the continuous Katz Centrality Matrix. For this, understanding matrix valued functions is imperative.

> *Let $Q(t)$ be the **continuous Katz Centrality Matrix**. Then, $Q(t)$ satisfies:*
>
> $$Q'(t) = -Q(t)\log(I - \alpha A(t)) + b(I - Q(t))$$
>
> *where:*
>
> - *$A(t)$ is the **adjacency matrix** for a **continuous** dynamically evolving network*
> - *b is the **discounting rate**, used for the **age discounting** update step*

Proof. Suppose $A(t)$ defines the adjacency matrix of an evolving network over continuous time (so that $t \geq 0$).

If we fix $\delta t > 0$, then in the discrete setting, each A_k can be thought of as:

$$A_k = \bigcup_{t \in ((k-1)\delta t, k\delta t]} A(t)$$

In particular, if we then take $\delta t \to 0$, we will get that:

$$A_k \to A(k\delta t)$$

which is what we had over the discrete case. Then, to this approximation of the discrete by using the continuous, we can apply the age discounting update formula. In particular, for $T = K\delta t$), we have that:

$$Q(T + \delta t) = (I + e^{-b\delta t}(Q(T) - I))(I - \alpha A_{K+1})^{-1}$$

We can't just take $\delta t \to 0$: we could technically split δt into smaller subintervals, which would yield more terms $(I - \alpha A_\gamma)^{-1}$ on the interval $(T, T + \delta]$. This would make convergence of the formula as $\delta t \to 0$ impossible.

Instead, we need a "trick" to avoid the generation of these new terms as $\delta t \to 0$. To this end, we modify the update expression, and use a fractional power:

$$Q(T + \delta t) = (I + e^{-b\delta t}(Q(T) - I))(I - \alpha A_{K+1})^{-\delta t}$$

By definition of fractional powers, we then have:

$$Q(T + \delta t) = (I + e^{-b\delta t}(Q(T) - I)) \exp\left(-\delta t \log(I - \alpha A_{K+1})\right)$$

For this to even make sense, the principal logarithm of $I - \alpha A_{K+1}$ must be defined. However, notice that, by requirement of convergence of the Katz Centrality Matrix, the spectral radius of $A(t)$ will always satisfy:

$$\rho(A) < \frac{1}{\alpha}$$

In particular, if λ is any eigenvalue of A, nad using the fact that 1 is the only eigenvalue of the identity, the eigenvalues of $I - \alpha A$ will be of the form:

$$1 - \alpha\lambda > 1 - \alpha\rho(A) > 1 - \frac{\alpha}{\alpha} > 0$$

so in particular, $1 - \alpha A$ always has positive eigenvalues, and thus, its principal logarithm is well-defined. Hence, so are fractional powers.

Now, if we let $\delta t \to 0$ and apply the definition of the derivative, we obtain:

$$Q'(t) = -Q(t) \log(I - \alpha A(t)) + b(I - Q(t))$$

as required.

□

- **What is the benefit of adopting an ODE approach?**
  - in practice, we don't aim to solve the ODE **analytically**
  - instead, we can use **numerical methods**, which use **adaptive timesteps**
  - this allows us to analyse dramatic changes in network behaviour (for example, as $A(t)$ start evolving more dynamically, we might use a small timestep to better gauge these changes)
  - by using the solver, we are using time discretisation under the hood
- **How do the broadcast and receiver centralities change under the new definition of the Katz Centrality matrix?**
  - recall:

> *The **broadcast centralities** measure the outward influence of each vertex within the network, as a weighted sum of walks from each vertex to anywhere.*
>
> ---
>
> *These are given by the **row sums** in the **Katz Centrality Matrix**.*

*The **receive centralities** measure the inwards influence of each vertex within the network, as a weighted sum of walks from anywhere to each vertex.*

---

*These are given by the **column sums** in the **Katz Centrality Matrix**.*

- if we define $\underline{s}$ to be the column vector of ones, we can define:

$$\underline{b}(t) = Q(t)\underline{s} \qquad \underline{r}(t) = Q^T\underline{s}$$

- the **receiver centrality** satisfies an ODE:

$$r'(t9 = b(\underline{s}(t) - \underline{r}(t)) - (\log(I - \alpha A(t)))^T\underline{r}(t) \qquad \underline{r}(0) = \underline{s}$$

- however, the **broadcast centrality** doesn't have this: intuitively, this is because $\underline{b}$ deals with **outgoing** information from a vertex, which doesn't include where the information is currently, and thus, outward information flow can't be updated using just $\underline{b}$
- on the other hand, $\underline{r}$ contains **incoming** information, which can be updated by knowing $\underline{r}$

2 Dynamics on Networks

In Week 3, we aimed to identify modular structures. This was because we are often interested in analysing a network from the point of view of how different modules interact (i.e group of friends).

2.1 Notation

- we use n for the **number of vertices** and m for the **number of edges**
- in **directeed graph** $A \in \mathbb{R}^{n\times n}$, the **weight** between vertices i, j is given by A_{ij}
- the **weighted out-degree** of nodes is the vector:

$$\underline{d} = A\hat{1}$$

  where $\hat{1}$ is the vector of ones
- the **combinatorial Laplacian** is denoted with L:

Let A be a $n \times n$ matrix which is:

- ***real***
- ***non-negative***
- ***normal***

*If $\underline{s} = \underline{1} \in \mathbb{R}^n$, then $A\underline{s} = (d_1, \ldots, d_n)^T$ contains the **row sums** of A.*

If we define:

$$D = \text{diag}(d_1, \ldots, d_n)$$

*the combinatorial **Laplacian** of A is the **symmetric** matrix:*

$$L = D - A$$

L is:

- **symmetric**
- **positive semi-definite**
- if A is connected, L has a **simple zero eigenvalue**, all its eigenvalues are positive, and the **eigenvector** corresponding to the smallest non-zero eigenvalue (known as the **Fiedler eignevalue**, λ_F) is the **Fiedler eigenvector**

2.2 Consensus Dynamics

2.2.1 Definition: Average Consensus Dynamics

*Let A be a **connected network** of n nodes. Endow each node with a **scalar state variable** which is **time dependent**:*

$$x_i(t) \in \mathbb{R}$$

and define:

$$\underline{x}(t) = (x_1(t), \ldots, x_n(t))^T$$

*The **average consensus dynamics** on A is defined by the ODE:*

$$\dot{\underline{x}} = -L\underline{x}$$

2.2.2 Behaviour of Average Consensus Dynamics

- **What does the ODE above amount to componentwise?**
  - we have that:

$$\dot{\underline{x}} = -L\underline{x} = A\underline{x} - D\underline{x}$$

so that:

$$\dot{x}_i = \sum_{j \in [1,n]} A_{ij} x_j - d_i x_i$$

But by definition:

$$d_i = \sum_{j \in [1,n]} A_{ij}$$

– hence, **componentwise** we have that:

$$\dot{x}_i = \sum_{j \in [1,n]} A_{ij}(x_j - x_i)$$

- **Why is this setting called average consensus dynamics?**
  - if $\forall i, j, x_i \approx x_j$, then:
  $$\forall i, \qquad \dot{x}_i \approx 0$$
  - thus, $\underline{x}$ **stabilises** into a value whenever each of its entries are **pairwise similar**
  - thus, the ODE will "push" $\underline{x}$ to a **consensus state**, where each node has the **same state variable**
  - in fact:
  $$\lim_{t \to \infty} x_i = \frac{\hat{1}^T \cdot \underline{x}_0}{n}$$
  so each x_i will tend to the **arithmetic mean** of the initial node states
  - this corresponds to the notion that, in absence of **external input**, opinions in communities tend to "average out"
- **What dominates the rate of convergence to the average state?**
  - convergence is dominated by the **Fiedler eigenvalue** λ_F:
  $$\underline{x}(t) = x^* \hat{1} + \mathcal{O}(e^{-\lambda_F t})$$
  - to see why, we can diagonalise:
  $$L = U \Lambda U^T$$
  and then:
  $$\dot{\underline{x}} = -L\underline{x} \implies \underline{x} = U e^{-\Lambda t} U^T \underline{x}_0$$
  Since λ_F is the smallest eigenvalue, then all other terms get dominated by $e^{-\Lambda_F t}$

2.3 Time-Scale Separation in Dynamical Systems

More on time-scaled separation can be found in the following book: Neuronal Dynamics.

2.3.1 Definition: Time-Scale Separation in ODEs

Time-scale separation *occurs when observing two different systems, one of which is assumed to occur much faster than the other.*

As ODEs, we may write:

$$\frac{dx}{dt} = f(x, y) \qquad \frac{dy}{dt} = \varepsilon g(x, y)$$

where $0 < \varepsilon << 1$ *is* ***very small and positive****. In this setting,* $x(t)$ *changes more rapidly than* $y(t)$*.*

Alternatively, if we define $\tau = \varepsilon t$*, then:*

$$\frac{dy}{d\tau} = g(x, y)$$

where t, τ *represent the fact that* x, y ***evolve*** *over different* ***timescales*** *(*τ *is the* ***slow timescale****,* t *is the* ***fast timescale****).*

2.3.2 Exploiting Time-Scale Separation for System Decoupling

- **How can short term behaviour of a time-scale separated system be studied?**
  - consider how the coupled system x, y evolves in the **short term**
  - in such a situation, only the **fast system** $x(t)$ will do anything interesting; in the **slow system** $y(t)$ will be **effectively constant**
  - thus, as $\varepsilon \to 0$ the system is essentially decoupled
- **How can long term behaviour of a time-separated sysmte be studied?**
  - in the **long term**, we may assume that the **fast system** $x(t)$ will have reached a **stable configuration** x^*
  - over the fast time-scale, we may assume that y is somewhat constant (at each step), so that the **stable point**:
    $$x(t) \to x^*(y)$$
    acts as a function of y
  - in the **short term**, we thus have a system:
    $$\frac{dy}{d\tau} = g(x^*(y), y)$$
  - this will lead to errors, but allows us to focus on the simpler 1-dimensional system

2.4 Time-Scale Separation in Consensus Dynamics

2.4.1 Proposition: Solution to Average Consensus Dynamics in Terms of Eigenvectors and Eigenvalues

Consider the ***average consensus dynamics*** *ODE:*

$$\dot{\underline{x}} = -L\underline{x}$$

with initial condition $\underline{x}_0$.

If $\underline{v}_i$ *are* ***eigenvectors*** *of* L *with corresponding* ***eigenvalues*** λ_i*, then the* ***solution*** *to the* ***average consensus dynamics*** *is:*

$$\underline{x}(t) = \sum_i \exp(-\lambda_i t)\underline{v}_i\underline{v}_i^T\underline{x}_0$$

Proof. Standard differential equation knowledge tells us that the solution to the ODE is given by:

$$\underline{x}(t) = \exp(-Lt)\underline{x}_0$$

The Laplacian is diagonalisable, so we can write it as:

$$L = U\Lambda U^T$$

where U has the $\underline{v}_i$ as column vectors, and Λ is a diagonal matrix containing the iegenvalues of L. Then, we have that:

$$L = \sum_i \lambda_i \underline{v}_i\underline{v}_i^T$$

Hence, we obtain the desired result:

$$\underline{x}(t) = \sum_i \exp(-\lambda_i t)\underline{v}_i\underline{v}_i^T\underline{x}_0$$

□

2.4.2 Analysing Long and Short Term Consensus

- **How does time-scale separation arise in the solution to the average consensus dynamics?**
  - the contribution of each **eigenvector** $\underline{v}_i$ to $\underline{x}$ is scaled by the term $\exp(-\lambda_i t)$
  - thus, they depend on the **eigenvalue** λ_i
  - in particular, if we define

    $$\tau_i = \frac{1}{\lambda_i}$$

    we see that as $t > \tau_i$ the contribution of $\underline{v}_i$ degrades

- if there is a **separation** in the magnitude of **eigenvalues** (such that there are k "small" **eigenvalues**, which WLOG we may call $\{0 = \lambda_1, \ldots, \lambda_k\}$, and such that $\lambda_{k+1} >> \lambda_k$) then there is a time scale separation
- this occurs whenever $t > \tau_{k+1}$, point at which the **eigenvectors** with the **larger eigenvalues** contribute negligibly to $\underline{x}(t)$
- in this respect, $\underline{x}$ (and thus its dynamics) can be reasonable described by the k smallest **eigenvectors** (they form a **dominant invariant subspace of the dynamics**)

- **What matrix construction can be used to analyse the time-scale separation in consensus dynamics?**
  - consider a network composed of k modules, which have **strong intraconnections**, but **weak interconnection**
  - such a network can be described as:

$$A = A_{structure} + A_{random}$$

    where:
    * $A_{structure}$ is a **block diagonal matrix**, composed of A_i, which are the **adjacency matrices** of individual modules
    * A_{random} can be thought of as "noise" (i.e an ER graph), meant to add **sparse connectivity** amongst the different modules
  - we can then ask questions about how the **eigenvalues** of $L = L_{structure} + L_{random}$ affect the **time-scale separation** of A

- **What are the eigenvectors for L when $L_{random} = 0$?**
  - if $L_{random} = 0$, then A consists of k separated components
  - the number of **components** in A corresponds to the number of 0 eigenvalues of L, so L has $\lambda = 0$ as the **unique eigenvalue**, with **algebraic** and **geometric** multiplicity 0
  - for each **component** j, there is an associated **eigenvector** defined componentwise via:

$$\underline{c}(j)_i = \begin{cases} 1, & i \in j \\ 0, & otherwise \end{cases}$$

- **How are the eigenvectors of L with the smallest k eigenvalues related to the eigenvectors of L when we assume that $L_{random} = 0$?**
  - assume that $L_{random} \neq 0$, and that there is a collection of k smallest eigenvalues of L
  - then, for networks of the form:

$$A = A_{structure} + A_{random}$$

    the **Davis-Kahan Theorem** states that, provided that the noise A_{random} is not too high, the spaces:
    * Y, spanned by $\{\underline{c}(j)\}_{j \in [1,k]}$, the eigenvectors of L when $L_{random} = 0$
    * Y', spanned by the eigenvectors of L (when $L_{random} \neq 0$) corresponding to the k smallest eigenvalues

    are similar:

$$Y \approx Y'$$

- **What implications does the Davis-Kahan Theorem have in terms of time-scale separation in consensus dynamics?**

– in the **long term**, when the k smallest eigenvalues define the behaviour of $\underline{x}$, L behaves as if the Laplacian of k separated components
– in particular, **global network consensus** is achieved in the **long term**
– on the other hand, in the **short term**, consensus is reached **within blocks**
– thus, **consensus dynamics** in this network arise as a stwo step process:
  1. intra-module consensus
  2. global consense

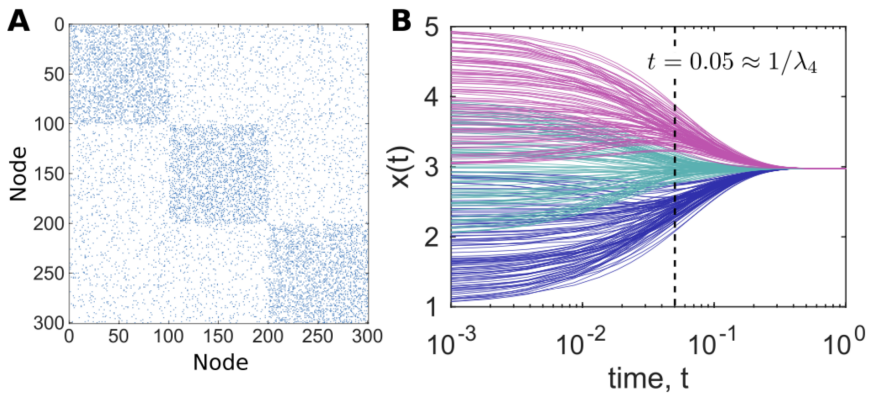

Figure 1: Consensus is first reached within each of the blocks, before collapsing into a global consensus.

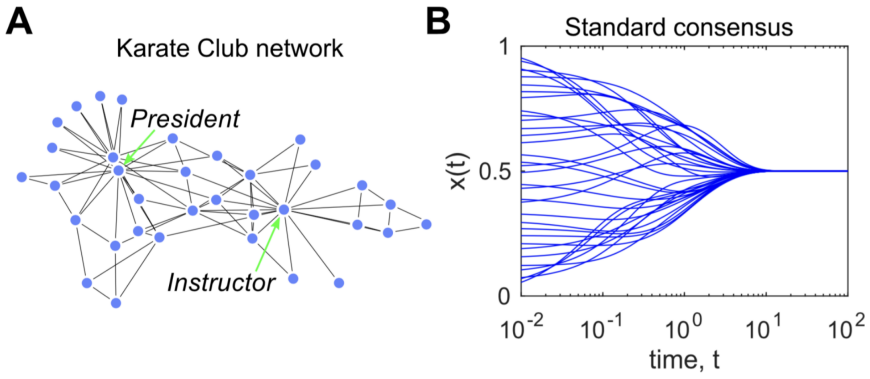

Figure 2: Consensus arising in the karate network.