Announcements, Assignments, and Reminders

- Welcome to the class!
Prerequisites

- basic probability, statistics, and random processes (e.g., EE505 or a Stat 5xx class or consent of the instructor).
Prerequisites

- basic probability, statistics, and random processes (e.g., EE505 or a Stat 5xx class or consent of the instructor).
- Knowledge of and access to matlab (some of the homeworks will require matlab). See https://www.ee.washington.edu/computing/index.html if you are a UW student for access via the EE systems.
Prerequisites

- basic probability, statistics, and random processes (e.g., EE505 or a Stat 5xx class or consent of the instructor).
- Knowledge of and access to matlab (some of the homeworks will require matlab). See https://www.ee.washington.edu/computing/index.html if you are a UW student for access via the EE systems.
- Knowledge of some programming language (C, C++, Java, etc.)
Prerequisites

- basic probability, statistics, and random processes (e.g., EE505 or a Stat 5xx class or consent of the instructor).
- Knowledge of and access to matlab (some of the homeworks will require matlab). See https://www.ee.washington.edu/computing/index.html if you are a UW student for access via the EE systems.
- Knowledge of some programming language (C, C++, Java, etc.)
- It would be useful to know static graphical models, but this is not absolutely necessary as the inference methods and applications are so different between the static and the dynamic case.
Prerequisites

- basic probability, statistics, and random processes (e.g., EE505 or a Stat 5xx class or consent of the instructor).
- Knowledge of and access to matlab (some of the homeworks will require matlab). See https://www.ee.washington.edu/computing/index.html if you are a UW student for access via the EE systems.
- Knowledge of some programming language (C, C++, Java, etc.)
- It would be useful to know static graphical models, but this is not absolutely necessary as the inference methods and applications are so different between the static and the dynamic case.
- The course is open to students in all UW departments.
Class web links and infrastructure

- Check in with our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/) for up to date announcements, homeworks, etc.
Class web links and infrastructure

- Check in with our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/) for up to date announcements, homeworks, etc.

- All homeworks will be due via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379)
Class web links and infrastructure

- Check in with our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/) for up to date announcements, homeworks, etc.
- All homeworks will be due via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379)
- All questions should be posted to our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/31332)
Check in with our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/) for up to date announcements, homeworks, etc.

All homeworks will be due via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379)

All questions should be posted to our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/31332)

You can contact me anonymously if you wish via anonymous email (https://catalyst.uw.edu/umail/form/bilmes/4144)
Homeworks

- There will be 3-6 homeworks this quarter, due about 1-1.5 weeks after they are assigned.
Homeworks

- There will be 3-6 homeworks this quarter, due about 1-1.5 weeks after they are assigned.
- Some will involve programming assignments.
Homeworks

- There will be 3-6 homeworks this quarter, due about 1-1.5 weeks after they are assigned.
- Some will involve programming assignments.
- All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no paper assignments accepted.
Final Project

- There will be a final project for the course,
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.

What will be due (and what is graded):
1. 15-20 minute conference style presentation
2. Presentation slides (electronically)
3. A 4-page writeup summarizing the presentation.
4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).

All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no dead-tree assignments accepted.

Final presentations will be held Monday, March 18, 2013, 2:00-4:00 pm, in PCAR 492.

Otherwise, no regular midterm and final exam.
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- **What will be due (and what is graded):**

  1. 15-20 minute conference style presentation
  2. Presentation slides (electronically)
  3. A 4-page writeup summarizing the presentation.
  4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).

All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no dead-tree assignments accepted.

Final presentations will be held Monday, March 18, 2013, 2:00-4:20 pm, in PCAR 492.

Otherwise, no regular midterm and final exam.
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- What will be due (and what is graded):
  1. 15-20 minute conference style presentation
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- What will be due (and what is graded):
  1. 15-20 minute conference style presentation
  2. presentation slides (electronically)
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- What will be due (and what is graded):
  1. 15-20 minute conference style presentation
  2. presentation slides (electronically)
  3. A 4-page writeup summarizing the presentation.
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- What will be due (and what is graded):
  1. 15-20 minute conference style presentation
  2. presentation slides (electronically)
  3. A 4-page writeup summarizing the presentation.
  4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).
Final Project

- There will be a final project for the course,
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.

What will be due (and what is graded):

1. 15-20 minute conference style presentation
2. Presentation slides (electronically)
3. A 4-page writeup summarizing the presentation.
4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).

- All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no dead-tree assignments accepted.
There will be a final project for the course,

The projects must have something to do with DGMs.

Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.

What will be due (and what is graded):

1. 15-20 minute conference style presentation
2. Presentation slides (electronically)
3. A 4-page writeup summarizing the presentation.
4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).

All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no dead-tree assignments accepted.

Final presentations will be held Monday, March 18, 2013, 230-420 pm, in PCAR 492.
Final Project

- There will be a final project for the course.
- The projects must have something to do with DGMs.
- Ideally, it will be original research related to DGMs, but it may also be an in-depth paper review.
- What will be due (and what is graded):
  1. 15-20 minute conference style presentation
  2. Presentation slides (electronically)
  3. A 4-page writeup summarizing the presentation.
  4. Preliminary deadlines leading up to the final project (e.g., project summaries and progress reports).
- All will be due electronically via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/25379), no dead-tree assignments accepted.
- Final presentations will be held Monday, March 18, 2013, 230-420 pm, in PCAR 492.
- Otherwise, no regular midterm and final exam.
On Final Project

- Project should ideally be on some aspect of the material we have learnt, some aspect of dynamic graphical models. Possible good projects include:
  - an implementation (i.e., a fast implementation of some DGMs algorithm) and reporting and experience that you gain in doing this. Application to real data.
  - A paper summary, of papers that we are not going to cover in this class.
  - A new idea of your own, new algorithms and/or theoretical results. (e.g., approximation error for a sequential model).
  - Application of a DGM to a data domain (e.g., application of dynamic Bayesian networks to speech/language/biology/surgery or some other sequential data domain).

- The ideal project should be research-oriented, it is not acceptable to propose whatever machine learning task you are currently working on (e.g., “An application of SVMs to protein folding” is not acceptable).
Grading

Grades will be based on a combination of:

- final project (45%)
Grading

Grades will be based on a combination of:

- final project (45%)
- class attendance (5%)
Grading

Grades will be based on a combination of:

- final project (45%)
- class attendance (5%)
- class participation (10%)
Grading

Grades will be based on a combination of:

- final project (45%)
- class attendance (5%)
- class participation (10%)
- homework assignments (40%)
Dynamic Graphical Models (DGMs)

The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. Hidden Markov models,
3. Conditional Markov models,
4. Dynamic Bayesian networks,
5. Sequential conditional random fields,
6. Deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. Switching Kalman filters,
9. Linear and non-linear dynamical systems,
10. Other time-series methods, time permitting.

We'll make a distinction between models that are inherently generative vs. discriminative, and how either generative or discriminative training objectives can be used to train them. Hence, four possible combinations.

We'll discuss various flavors of discriminative sequential model (e.g., local vs. global discriminative training)
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

Dynamic Graphical Models (DGMs)

The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. hidden Markov models,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. Hidden Markov models,
3. Conditional Markov models,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. Hidden Markov models,
3. Conditional Markov models,
4. Dynamic Bayesian networks,
Dynamic Graphical Models (DGMs)

The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. hidden Markov models,
3. conditional Markov models,
4. dynamic Bayesian networks,
5. sequential conditional random fields,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. hidden Markov models,
3. conditional Markov models,
4. dynamic Bayesian networks,
5. sequential conditional random fields,
6. deep belief networks and deep models in sequential processing,
Dynamic Graphical Models (DGMs)

- The plan for this course is to serve as a thorough overview of dynamic graphical models including:
  2. hidden Markov models,
  3. conditional Markov models,
  4. dynamic Bayesian networks,
  5. sequential conditional random fields,
  6. deep belief networks and deep models in sequential processing,
  7. Kalman filters,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. hidden Markov models,
3. conditional Markov models,
4. dynamic Bayesian networks,
5. sequential conditional random fields,
6. deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. switching Kalman filters,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:
2. hidden Markov models,
3. conditional Markov models,
4. dynamic Bayesian networks,
5. sequential conditional random fields,
6. deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. switching Kalman filters,
9. linear and non-linear dynamical systems,
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. hidden Markov models,
3. conditional Markov models,
4. dynamic Bayesian networks,
5. sequential conditional random fields,
6. deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. switching Kalman filters,
9. linear and non-linear dynamical systems,
10. other time-series methods, time permitting.
Dynamic Graphical Models (DGMs)

The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. Hidden Markov models,
3. Conditional Markov models,
4. Dynamic Bayesian networks,
5. Sequential conditional random fields,
6. Deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. Switching Kalman filters,
9. Linear and non-linear dynamical systems,
10. Other time-series methods, time permitting.

We’ll make a distinction between models that are inherently generative vs. discriminative, and how either generative or discriminative training objectives can be used to train them. Hence, four possible combinations.
The plan for this course is to serve as a thorough overview of dynamic graphical models including:

2. Hidden Markov models,
3. Conditional Markov models,
4. Dynamic Bayesian networks,
5. Sequential conditional random fields,
6. Deep belief networks and deep models in sequential processing,
7. Kalman filters,
8. Switching Kalman filters,
9. Linear and non-linear dynamical systems,
10. Other time-series methods, time permitting.

We’ll make a distinction between models that are inherently generative vs. discriminative, and how either generative or discriminative training objectives can be used to train them. Hence, four possible combinations.

We’ll discuss various flavors of discriminative sequential model (e.g.,
For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
Dynamic Graphical Models (DGMs)

For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).

This will include:

1. forward/backwards algorithms,
Dynamic Graphical Models (DGMs)

For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).

This will include:

1. forward/backwards algorithms,
2. temporal junction trees and dynamic triangulation,
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
  1. forward/backwards algorithms,
  2. temporal junction trees and dynamic triangulation,
  3. conditioning and factorization approaches,
  4. variational approaches,
  5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
  6. beam pruning strategies,
  7. multi-pass course-to-fine strategies.
  8. time space tradeoffs including the island algorithm, the archipelagos algorithm, and other time-space trade-off strategies.
  9. The challenges and approaches to parallelism with dynamic models on large parallel machines.
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
  1. forward/backwards algorithms,
  2. temporal junction trees and dynamic triangulation,
  3. conditioning and factorization approaches,
  4. variational approaches,
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
  1. forward/backwards algorithms,
  2. temporal junction trees and dynamic triangulation,
  3. conditioning and factorization approaches,
  4. variational approaches,
  5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
Dynamic Graphical Models (DGMs)

For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).

This will include:

1. forward/backwards algorithms,
2. temporal junction trees and dynamic triangulation,
3. conditioning and factorization approaches,
4. variational approaches,
5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
6. beam pruning strategies,
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
  1. forward/backwards algorithms,
  2. temporal junction trees and dynamic triangulation,
  3. conditioning and factorization approaches,
  4. variational approaches,
  5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
  6. beam pruning strategies,
  7. multi-pass course-to-fine strategies.
Dynamic Graphical Models (DGMs)

- For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).
- This will include:
  1. forward/backwards algorithms,
  2. temporal junction trees and dynamic triangulation,
  3. conditioning and factorization approaches,
  4. variational approaches,
  5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
  6. beam pruning strategies,
  7. multi-pass course-to-fine strategies.
  8. time space tradeoffs including the island algorithm, the archipelagos algorithm, and other time-space trade-off strategies.
Dynamic Graphical Models (DGMs)

For above models, we will discuss exact and approximate inference methods (i.e., efficient means to compute probabilities) that are particularly well suited to dynamic models (as distinct from static graphical models).

This will include:

1. forward/backwards algorithms,
2. temporal junction trees and dynamic triangulation,
3. conditioning and factorization approaches,
4. variational approaches,
5. sampling and particle filter approaches (including importance sampling and Rao-Blackwellization),
6. beam pruning strategies,
7. multi-pass course-to-fine strategies.
8. time space tradeoffs including the island algorithm, the archipelagos algorithm, and other time-space trade-off strategies
9. The challenges and approaches to parallelism with dynamic models on large parallel machines.
We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

1. Generative training (e.g., Maximum likelihood, and the EM algorithm)
2. Discriminative training (MMIE, MDI, and MCE, etc.)
3. Model-specific approximations (often from the speech recognition community)
4. Structured max margin approaches (often from the machine learning community).

Interplay between models and training objectives (e.g., the label-bias problem in CMMs vs. its training objective).
We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

- Including:
We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

- generative training (e.g., Maximum likelihood, and the EM algorithm)
- discriminative training (MMIE, MDI, and MCE, etc.)
- model-specific approximations (often from the speech recognition community)
- structured max margin approaches (often from the machine learning community).

Interplay between models and training objectives (e.g., the label-bias problem in CMMs vs. its training objective).
Dynamic Graphical Models (DGMs) - learning algorithms

- We will also cover learning algorithms (how to learn either the parameters of DGMs, including:
- Including:
  1. generative training (e.g., Maximum likelihood, and the EM algorithm)
  2. discriminative training (MMIE, MDI, and MCE, etc.)
Dynamic Graphical Models (DGMs) - learning algorithms

- We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

  - Including:
    1. generative training (e.g., Maximum likelihood, and the EM algorithm)
    2. discriminative training (MMIE, MDI, and MCE, etc.)
    3. model-specific approximations (often from the speech recognition community)
We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

- generative training (e.g., Maximum likelihood, and the EM algorithm)
- discriminative training (MMIE, MDI, and MCE, etc.)
- model-specific approximations (often from the speech recognition community)
- structured max margin approaches (often from the machine learning community).
We will also cover learning algorithms (how to learn either the parameters of DGMs, including:

- Generative training (e.g., Maximum likelihood, and the EM algorithm)
- Discriminative training (MMIE, MDI, and MCE, etc.)
- Model-specific approximations (often from the speech recognition community)
- Structured max margin approaches (often from the machine learning community).

Interplay between models and training objectives (e.g., the label-bias problem in CMMs vs. its training objective).
Dynamic Graphical Models (DGMs) - Applications

Applications will come mainly from:

- speech recognition,
Applications will come mainly from:

- speech recognition,
- natural language processing,
Dynamic Graphical Models (DGMs) - Applications

Applications will come mainly from:

- speech recognition,
- natural language processing,
- bio-informatics
Applications will come mainly from:

- speech recognition,
- natural language processing,
- bio-informatics
- activity recognition and other recent problems (in home device prediction)
Applications will come mainly from:

- speech recognition,
- natural language processing,
- bio-informatics
- activity recognition and other recent problems (in home device prediction)
- econometrics
Applications will come mainly from:

- speech recognition,
- natural language processing,
- bio-informatics
- activity recognition and other recent problems (in home device prediction)
- econometrics
- and robotics (i.e., localization and mapping).
Relevant Books and Readings

Relevant Books and Readings

- Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman, 2009: in particular, chapter 6 (template models) and 15 (inference in temporal models)


Sequential Monte Carlo Methods in Practice, edited by Doucet, de Freitas, Gordon, and Smith, 2001
Relevant Books and Readings

- Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman, 2009: in particular, chapter 6 (template models) and 15 (inference in temporal models)

Relevant Books and Readings

- Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman, 2009: in particular, chapter 6 (template models) and 15 (inference in temporal models)
Relevant Books and Readings

- Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman, 2009: in particular, chapter 6 (template models) and 15 (inference in temporal models)
Relevant Books and Readings

- Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman, 2009: in particular, chapter 6 (template models) and 15 (inference in temporal models).
- Sequential Monte Carlo Methods in Practice, edited by Doucet, de Freitas, Gordon, and Smith, 2001
Readings are in a sub-directory “reading_drafts” directly below our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/).
Relevant Readings: Readings/handouts

- Readings are in a sub-directory “reading_drafts” directly below our web page (http://j.ee.washington.edu/~bilmes/classes/ee596a_winter_2013/).
- uid is this class name (lower case) and pwd are the quarter/year of the class.
- Reading rough drafts are in “doc.pdf”, read chapters 1 - 6.
- In particular, read chapter 1 and 2 (overview, and about families)
- Read chapters 3 (undirected) and chapters 4 (directed) graphical models.
- read chapter 5 (on evidence)
- read chapter 6 (inference on trees)
- Then, read Wainwright/Jordan book chaps 3/4/5
  
  http://dx.doi.org/10.1561/2200000001
Dynamic graphical model class Readings

- Reading rough drafts are in “doc.pdf”.
- Read chapter 8.1 (template models)
- Read chapter 8.2 (stochastic processes, discrete-time Markov chains)
- Read chapter 8.3 (Markov chains)
Read 8.1 - 8.3 in “doc.pdf”
This is where review material will go each lecture.
Graphical model: a graph $G = (V, E)$ & set of properties $\mathcal{M}$ that define family $\mathcal{F}(G, \mathcal{M})$ of distributions that abide by all of the properties of the graph.
Graphical Models

- Graphical model: a graph $G = (V, E)$ & set of properties $\mathcal{M}$ that define family $\mathcal{F}(G, \mathcal{M})$ of distributes that abide by all of the properties of the graph.

- A given probability distribution $p$ might or might not be a member of $\mathcal{F}(G, \mathcal{M})$. If so, we say $p \in \mathcal{F}(G, \mathcal{M})$. 

Many types of graphs (e.g., DAG, undirected, bipartite) and sets of rules, leading to different families of graphical model (Bayesian networks, Markov random fields, chain graphs, etc.)

Key aspects of a graphical model is that some graph-theoretic property helps us to deduce an inference algorithm on any member of the family — true for both exact and approximate inference. Often it has to do with clusterings and/or partitionings of the! nodes.
Graphical Models

• Graphical model: a graph $G = (V, E)$ & set of properties $\mathcal{M}$ that define family $\mathcal{F}(G, \mathcal{M})$ of distributes that abide by all of the properties of the graph.

• A given probability distribution $p$ might or might not be a member of $\mathcal{F}(G, \mathcal{M})$. If so, we say $p \in \mathcal{F}(G, \mathcal{M})$.

• Many types of graphs (e.g., DAG, undirected, bipartite) and sets of rules, leading to different families of graphical model (Bayesian network, Markov random fields, chain graphs, etc.)
Graphical Models

- Graphical model: a graph $G = (V, E)$ & set of properties $\mathcal{M}$ that define family $\mathcal{F}(G, \mathcal{M})$ of distributions that abide by all of the properties of the graph.
- A given probability distribution $p$ might or might not be a member of $\mathcal{F}(G, \mathcal{M})$. If so, we say $p \in \mathcal{F}(G, \mathcal{M})$.
- Many types of graphs (e.g., DAG, undirected, bipartite) and sets of rules, leading to different families of graphical model (Bayesian network, Markov random fields, chain graphs, etc.)
- Key aspects of a graphical model is that some graph-theoretic property helps us to deduce an inference algorithm on any member of the family — true for both exact and approximate inference.
Graphical Models

- Graphical model: a graph $G = (V, E)$ & set of properties $\mathcal{M}$ that define family $\mathcal{F}(G, \mathcal{M})$ of distributes that abide by all of the properties of the graph.

- A given probability distribution $p$ might or might not be a member of $\mathcal{F}(G, \mathcal{M})$. If so, we say $p \in \mathcal{F}(G, \mathcal{M})$.

- Many types of graphs (e.g., DAG, undirected, bipartite) and sets of rules, leading to different families of graphical model (Bayesian network, Markov random fields, chain graphs, etc.)

- Key aspects of a graphical model is that some graph-theoretic property helps us to deduce an inference algorithm on any member of the family — true for both exact and approximate inference.

- Often it has to do with clusterings and/or partitionings of the nodes.
A goal of graphical model inference - produce generic algorithm

Any $p \in \mathcal{F}(G, M)$

A particular probabilistic query

Observed Data

Correct answer
Dynamic Graphical Models

- graphs that represent the temporal evolution of the statistical properties of a temporal signal.
Dynamic Graphical Models

- graphs that represent the temporal evolution of the statistical properties of a temporal signal.

- Many different varieties, some are better at explicating certain phenomena than others.
Dynamic Graphical Models

- graphs that represent the temporal evolution of the statistical properties of a temporal signal.

- Many different varieties, some are better at explicating certain phenomena than others.

- All can be expanded unboundedly along one dimension - time or position or length (w.l.o.g., we use “time”).
Dynamic Graphical Models

- graphs that represent the temporal evolution of the statistical properties of a temporal signal.
- Many different varieties, some are better at explicating certain phenomena than others.
- All can be expanded unboundedly along one dimension - time or position or length (w.l.o.g., we use “time”).
- All start with a template to be expanded.
Dynamic Graphical Models

- graphs that represent the temporal evolution of the statistical properties of a temporal signal.
- Many different varieties, some are better at explicating certain phenomena than others.
- All can be expanded unboundedly along one dimension - time or position or length (w.l.o.g., we use “time”).
- All start with a template to be expanded.
- Notation: $t$ is the time-dimension, and (when it exists) $T$ is the length of the model in the expanded dimension.
Dynamic Graphical Models

What is similar to static graphical model case?

- A graph $G$ and set of properties $M$ define a family of probability distributions.
Dynamic Graphical Models

What is similar to static graphical model case?

- A graph $G$ and set of properties $M$ define a family of probability distributions.
- Goal is to deduce a generic algorithm that can perform inference efficiently on any member of the corresponding family — amortization.
Dynamic Graphical Models

What is similar to static graphical model case?

- A graph $G$ and set of properties $M$ define a family of probability distributions.
- Goal is to deduce a generic algorithm that can perform inference efficiently on any member of the corresponding family — amortization.
- Ideal goal is to use graph-theoretic concepts to do so.
Dynamic Graphical Models

What is similar to static graphical model case?

- A graph $G$ and set of properties $\mathcal{M}$ define a family of probability distributions.
- Goal is to deduce a generic algorithm that can perform inference efficiently on any member of the corresponding family — amortization.
- Ideal goal is to use graph-theoretic concepts to do so.
- Graph visually depicts high-level information regarding “who may directly relate to whom”, high-level domain-specific interpretable.
Dynamic Graphical Models

What is different from static graphical model case?

- Any “graph” $G$ and properties $\mathcal{M}$ are defined via an expandable but static “template” that, along with rules, allow certain graphs to be instantiated.
Dynamic Graphical Models

What is different from static graphical model case?

- Any “graph” $G$ and properties $\mathcal{M}$ are defined via an expandable but static “template” that, along with rules, allow certain graphs to be instantiated.

- An expanded template corresponds to a graph in a GM in the normal sense, and rules also say how to do expansion for given length. Expansion is in one dimension only.
Dynamic Graphical Models

What is different from static graphical model case?

- Any "graph" $G$ and properties $\mathcal{M}$ are defined via an expandable but static "template" that, along with rules, allow certain graphs to be instantiated.

- An expanded template corresponds to a graph in a GM in the normal sense, and rules also say how to do expansion for given length. Expansion is in one dimension only.

- Distributions in the same family correspond to variable (unbounded) number of random variables — RVs can expand unboundedly in time, but still a member of the family.
Dynamic Graphical Models

What is different from static graphical model case?

- Any “graph” $G$ and properties $M$ are defined via an expandable but static “template” that, along with rules, allow certain graphs to be instantiated.
- An expanded template corresponds to a graph in a GM in the normal sense, and rules also say how to do expansion for given length. Expansion is in one dimension only.
- Distributions in the same family correspond to variable (unbounded) number of random variables — RVs can expand unboundedly in time, but still a member of the family.
- Graph is shaped differently. Much wider than it is taller, often there is a “time” parameter $T$ that expands the template.
Dynamic Graphical Models

What is different from static graphical model case?

- Any “graph” $G$ and properties $\mathcal{M}$ are defined via an expandable but static “template” that, along with rules, allow certain graphs to be instantiated.

- An expanded template corresponds to a graph in a GM in the normal sense, and rules also say how to do expansion for given length. Expansion is in one dimension only.

- Distributions in the same family correspond to variable (unbounded) number of random variables — RVs can expand unboundedly in time, but still a member of the family.

- Graph is shaped differently. Much wider than it is taller, often there is a “time” parameter $T$ that expands the template. For online inference, rules say how to expand a next chunk.
Dynamic Graphical Models

What is different from static graphical model case?

- Typically some form of parameter sharing (otherwise, unbounded time would require unbounded number of parameters).
Dynamic Graphical Models

What is different from static graphical model case?

- Typically some form of parameter sharing (otherwise, unbounded time would require unbounded number of parameters).
- There is some form of “temporal Markov property” in one way or another that, loosely speaking, takes the form: The future and past are rendered conditionally independent given the present.
Dynamic Graphical Models

What is different from static graphical model case?

- Typically some form of parameter sharing (otherwise, unbounded time would require unbounded number of parameters).
- There is some form of “temporal Markov property” in one way or another that, loosely speaking, takes the form: The future and past are rendered conditionally independent given the present.
- For probabilistic inference: Still want to deduce computational properties from the static template, rather than have to expand the template to each possible length, and then (re-)deduce an inference algorithm — amortization is again the goal.
Types of Dynamic Graphical Model

Many forms of DGM:

- **Markov Chain**
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
- Dynamic Markov Random Field
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
- Dynamic Markov Random Field
- Dynamic Conditional Random Field (CRF)
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
- Dynamic Markov Random Field
- Dynamic Conditional Random Field (CRF)
- Switching Linear Dynamical System
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
- Dynamic Markov Random Field
- Dynamic Conditional Random Field (CRF)
- Switching Linear Dynamical System
- Non-linear dynamical systems
Types of Dynamic Graphical Model

Many forms of DGM:

- Markov Chain
- Martingales
- Hidden Markov Model & Kalman Filter
- Dynamic Bayesian Network
- (Partially Observable) Markov Decision Process (MDP/POMDP)
- Markov chain smoothing with deep unaries (e.g., deep models in speech recognition).
- (Linear) Dynamical System
- Dynamic Markov Random Field
- Dynamic Conditional Random Field (CRF)
- Switching Linear Dynamical System
- Non-linear dynamical systems
- **Sequentially Structured Kernels/RHKS/SVMs & Structured Prediction, string kernels, Fisher kernels, etc.**
Many forms of DGM:

- In each case, a graph template can describe many important properties of members of these families. We will become fairly well versed in almost all of the above.
Many forms of DGM:

- In each case, a graph template can describe many important properties of members of these families. We will become fairly well versed in almost all of the above.

- In each case, some form of “the past is independent of the future given the present”, for various definitions of past, present, and future.
Many forms of DGM:

- In each case, a graph template can describe many important properties of members of these families. We will become fairly well versed in almost all of the above.

- In each case, some form of “the past is independent of the future given the present”, for various definitions of past, present, and future.

- In the next few slides, we see a few examples.
Markov Chains

- One of the simplest forms of time-series models
Markov Chains

- One of the simplest forms of time-series models
- Can exist in various orders. Example, 1st order Markov chain

\[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]

Here, the “present” is fairly local, only the current state.
Markov Chains

- One of the simplest forms of time-series models
- Can exist in various orders. Example, 1st order Markov chain

\[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]

Here, the “present” is fairly local, only the current state.

- 2nd order Markov chain

\[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]

The “present” is less local, two successive states.
Markov Chains

- One of the simplest forms of time-series models
- Can exist in various orders. Example, 1st order Markov chain
  \[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]
  Here, the “present” is fairly local, only the current state.
- 2nd order Markov chain
  \[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]
  The “present” is less local, two successive states.
- 3rd order Markov chain
  \[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \]
  The “present” is still less local, three successive states.
A Markov chain with other variables hanging off of it.
A Markov chain with other variables hanging off of it.

Typically, the Markov chain is hidden, and what is observed are the $X$ variables.
A Markov chain with other variables hanging off of it.

Typically, the Markov chain is hidden, and what is observed are the $X$ variables.

Markov Chain is typically 1st order, but extensions exist.
A Markov chain with other variables hanging off of it.

Typically, the Markov chain is hidden, and what is observed are the $X$ variables.

Markov Chain is typically 1st order, but extensions exist.

Kalman filter is identical, except it is joint Gaussian (inference in joint Gaussian is incredibly easy).
A Markov chain with other variables hanging off of it.

Typically, the Markov chain is hidden, and what is observed are the $X$ variables.

Markov Chain is typically 1st order, but extensions exist.

Kalman filter is identical, except it is joint Gaussian (inference in joint Gaussian is incredibly easy).

Multiple possible training algorithms (e.g., some discriminative and some generative) are available, each with computational implications.
A Markov chain with other variables hanging off of it.

Typically, the Markov chain is hidden, and what is observed are the $X$ variables.

Markov Chain is typically 1st order, but extensions exist.

Kalman filter is identical, except it is joint Gaussian (inference in joint Gaussian is incredibly easy).

Multiple possible training algorithms (e.g., some discriminative and some generative) are available, each with computational implications.

We’ll talk extensively about HMMs (see doc.pdf) writeup.
Switches direction of arrow in BN representation of an HMM, not possible to precisely represent with a MRF.
Switches direction of arrow in BN representation of an HMM, not possible to precisely represent with a MRF.

Original goal: attempt to be more “discriminative”, but this was a poor motivation.
Switches direction of arrow in BN representation of an HMM, not possible to precisely represent with a MRF.

Original goal: attempt to be more “discriminative”, but this was a poor motivation.

Can sometimes suffer from what is known as “label bias” or “observation bias”, but we’ll see that this is much less often a problem than commonly thought.
Switches direction of arrow in BN representation of an HMM, not possible to precisely represent with a MRF.

Original goal: attempt to be more “discriminative”, but this was a poor motivation.

Can sometimes suffer from what is known as “label bias” or “observation bias”, but we’ll see that this is much less often a problem than commonly thought.

Often used in NLP tagging tasks (e.g., take a sentence and produce the part-of-speech tags, like noun, verb, etc.).
Dynamic Bayesian Network - example

- The hidden state is inherently factored rather than monolithic.
The hidden state is inherently factored rather than monolithic.

- Inference within a given frame might require some smarts, more akin to static graphical models.
The hidden state is inherently factored rather than monolithic.

Inference within a given frame might require some smarts, more akin to static graphical models.

From a data-domain and/or scientific modeling perspective, much more flexible and powerful than an HMM.
The hidden state is inherently factored rather than monolithic.

Inference within a given frame might require some smarts, more akin to static graphical models.

From a data-domain and/or scientific modeling perspective, much more flexible and powerful than an HMM.

Sometimes, this can lead to significant computational advantages over the HMM as well (we will discuss when this is the case precisely).
The hidden state is inherently factored rather than monolithic.

Inference within a given frame might require some smarts, more akin to static graphical models.

From a data-domain and/or scientific modeling perspective, much more flexible and powerful than an HMM.

Sometimes, this can lead to significant computational advantages over the HMM as well (we will discuss when this is the case precisely).

Can also lead to much better parameter estimation (bias/variance), as factorization can act as a kind of regularization.
The hidden state is inherently factored rather than monolithic.

Inference within a given frame might require some smarts, more akin to static graphical models.

From a data-domain and/or scientific modeling perspective, much more flexible and powerful than an HMM.

Sometimes, this can lead to significant computational advantages over the HMM as well (we will discuss when this is the case precisely).

Can also lead to much better parameter estimation (bias/variance), as factorization can act as a kind of regularization.

Multiple training algorithms, each with computational implications.
Dynamic Conditional Random Field

\[ p(y|x) = p(y_1:T|x_1:T) = \frac{1}{Z(x)} \prod_{t=1}^{T} g(x_t, y_t) \prod_{t=2}^{T} h(y_t, y_{t-1}) \] (1.1)

\[ = \frac{1}{Z(x)} \prod_{t} h(y_t, y_{t-1}) g(x_t, y_t) \] (1.2)

- This is a conditional model only, not a joint model of \( x, y \).
Dynamic Conditional Random Field

\[
p(y|x) = p(y_{1:T}|x_{1:T}) = \frac{1}{Z(x)} \prod_{t=1}^{T} g(x_t, y_t) \prod_{t=2}^{T} h(y_t, y_{t-1}) \tag{1.1}
\]

\[
= \frac{1}{Z(x)} \prod_{t} h(y_t, y_{t-1}) g(x_t, y_t) \tag{1.2}
\]

- This is a conditional model only, not a joint model of \(x, y\).
- We need only that the hidden variables \(y\) factorize, the input \(\bar{x}\) is always observed, so does not contribute to state space.
\[ p(y|x) = p(y_{1:T}|x_{1:T}) = \frac{1}{Z(x)} \prod_{t=1}^{T} g(x_t, y_t) \prod_{t=2}^{T} h(y_t, y_{t-1}) \]  

\[ = \frac{1}{Z(x)} \prod_{t} h(y_t, y_{t-1}) g(x_t, y_t) \]  

- This is a conditional model only, not a joint model of \( x, y \).
- We need only that the hidden variables \( y \) factorize, the input \( \bar{x} \) is always observed, so does not contribute to state space.
- Often convex, so relatively easy to optimize (many simple iterative approaches, e.g., perceptron updates or gradient steps).
Dynamic Conditional Random Field

\[ p(y|x) = p(y_{1:T}|x_{1:T}) = \frac{1}{Z(x)} \prod_{t=1}^{T} g(x_t, y_t) \prod_{t=2}^{T} h(y_t, y_{t-1}) \]  

This is a conditional model only, not a joint model of \( x, y \).

We need only that the hidden variables \( y \) factorize, the input \( \bar{x} \) is always observed, so does not contribute to state space.

Often convex, so relatively easy to optimize (many simple iterative approaches, e.g., perceptron updates or gradient steps).

Multiple (imperfect) ways of drawing this in GM notation.
Dynamic Hidden Conditional Random Field

\[
p(y|\bar{x}) = \sum_{h_{1:T}} p(y_{1:T}, h_{1:T}|\bar{x}_{1:T})
\]

\[
= \frac{1}{Z(x)} \sum_{h_{1:T}} \prod_{t} g_{xh}(\bar{x}, h_t) g_{hh}(h_t, h_{t-1}) g_{hy}(h_t, y_t) h(y_t, y_{t-1})
\]

(1.3)

(1.4)

- A CRF with hidden variables
\[ p(y|\bar{x}) = \sum_{h_{1:T}} p(y_{1:T}, h_{1:T}|\bar{x}_{1:T}) \] 

\[ = \frac{1}{Z(x)} \sum_{h_{1:T}} \prod_{t} g_{xh}(\bar{x}, h_t) g_{hh}(h_t, h_{t-1}) g_{hy}(h_t, y_t) h(y_t, y_{t-1}) \]

- A CRF with hidden variables
- No longer convex
Dynamic Hidden Conditional Random Field

$$p(y|\bar{x}) = \sum_{h_{1:T}} p(y_{1:T}, h_{1:T}|\bar{x}_{1:T})$$  \hspace{1cm} (1.3)

$$= \frac{1}{Z(x)} \sum_{h_{1:T}} \prod_{t} g_{xh}(\bar{x}, h_{t})g_{hh}(h_{t}, h_{t-1})g_{hy}(h_{t}, y_{t})h(y_{t}, y_{t-1})$$  \hspace{1cm} (1.4)

- A CRF with hidden variables
- No longer convex
- Potentially much more powerful but much more computationally complex
Switching Linear Dynamical System

- A Kalman filter with an additional hidden Markov chain.
A Kalman filter with an additional hidden Markov chain.

We have that $p(x, h|y)$ is a Kalman filter, and $y$ is a discrete Markov chain.
A Kalman filter with an additional hidden Markov chain.

We have that \( p(x, h|y) \) is a Kalman filter, and \( y \) is a discrete Markov chain.

Can be considerably more complex than either an HMM or a Kalman filter since we now have switching hidden random continuous random variables, sampling approaches make it relatively easy.
These are only just a handful of models.
These are only just a handful of models.

We’ll see many more, and study the properties of each, as the quarter progresses.
Templates

- A static piece of a graph possibly with other annotations
Templates

- A static piece of a graph possibly with other annotations
- rules for expansion say how a template may be instantiated to form a particular (expanded) instance.
Templates

- A static piece of a graph possibly with other annotations
- Rules for expansion say how a template may be instantiated to form a particular (expanded) instance.
- Family of distributions corresponding to the template is the union of all probability distributions that obey the rules for any one of these expansions.
A static piece of a graph possibly with other annotations
rules for expansion say how a template may be instantiated to form a particular (expanded) instance.
family of distributions corresponding to the template is the union of all probability distributions that obey the rules for any one of these expansions.
a template may be expanded or instantiated, and use the terms interchangeably.
Templates

- A static piece of a graph possibly with other annotations
- rules for expansion say how a template may be instantiated to form a particular (expanded) instance.
- family of distributions corresponding to the template is the union of all probability distributions that obey the rules for any one of these expansions.
- a template may be expanded or instantiated, and use the terms interchangeably.
- A graphical model that is derived from a template is called an instantiated model.
Templates

- A static piece of a graph possibly with other annotations
- rules for expansion say how a template may be instantiated to form a particular (expanded) instance.
- family of distributions corresponding to the template is the union of all probability distributions that obey the rules for any one of these expansions.
- a template may be expanded or instantiated, and use the terms interchangeably.
- A graphical model that is derived from a template is called an instantiated model.
- Hierarchy of template models.
In a static graphical model, the graph $G$ is static and directly characterizes the family $\mathcal{F}(G, \mathcal{M})$. 
In a static graphical model, the graph $G$ is static and directly characterizes the family $\mathcal{F}(G, \mathcal{M})$.

Properties of $G$ directly determine which distributions $p$ are or are not members of the family.
In a static graphical model, the graph $G$ is static and directly characterizes the family $\mathcal{F}(G, \mathcal{M})$.

Properties of $G$ directly determine which distributions $p$ are or are not members of the family.

$G$ itself can determine the computational cost of performing inference (or at least an upper bound on this),
In a static graphical model, the graph $G$ is static and directly characterizes the family $\mathcal{F}(G, M)$.

Properties of $G$ directly determine which distributions $p$ are or are not members of the family.

$G$ itself can determine the computational cost of performing inference (or at least an upper bound on this),

operations on $G$ alone can produce fast exact or approximate inference.
Plate models

- a convenience to represent that a portion of a graph can be expanded any number of times, say $M$. 
Plate models

- a convenience to represent that a portion of a graph can be expanded any number of times, say $M$.
- The plate = graph stub surrounded by a rectangle, and an integer that indicates the number of times that the stub should be repeated.
Plate models

- A convenience to represent that a portion of a graph can be expanded any number of times, say $M$.
- The plate = graph stub surrounded by a rectangle, and an integer that indicates the number of times that the stub should be repeated.
- With no edges crossing the plate, graph is simply expanded $M$ times corresponding to $M$ set of random variable groups — groups are mutually independent of each other, dependencies exist within each group identical to the plate template.
Plate models

- With edges crossing plate, coupling may occur between stub instantiations.
Plate models

- With edges crossing plate, coupling may occur between stub instantiations.

- This may also occur hierarchically to express multiple groups with a common parameter in a Bayesian setting:

$$\theta$$

$$y_i$$

$$x_j$$

$$M$$

$$N$$
Plate models - self loops

- The stub itself may posses self-loops, something normally disallowed in a Bayesian network.
Plate models - self loops

- The stub itself may possess self-loops, something normally disallowed in a Bayesian network.
- This corresponds to edges between the same variable in successive instantiations of the stub. Compare with persistent edges in a dynamic graphical model.
The stub itself may posses self-loops, something normally disallowed in a Bayesian network.

This corresponds to edges between the same variable in successive instantiations of the stub. Compare with persistent edges in a dynamic graphical model.

HMM as a plate model:
DGMs are also template models but with a template more general than a plate model.
Dynamic Graphical Model

- DGMs are also template models but with a template more general than a plate model.
- Key limiting factor: template and expansion rules are typically sufficient to infer computational properties of any model after expansion, regardless of the amount of expansion.
Dynamic Graphical Model

- DGMs are also template models but with a template more general than a plate model.
- Key limiting factor: template and expansion rules are typically sufficient to infer computational properties of any model after expansion, regardless of the amount of expansion.
- Like static models, it should be possible to perform offline computation to produce an inference algorithm for any probability model in the family, regardless of the expansion.
DGMs are also template models but with a template more general than a plate model.

Key limiting factor: template and expansion rules are typically sufficient to infer computational properties of any model after expansion, regardless of the amount of expansion.

like static models, it should be possible to perform offline computation to produce an inference algorithm for any probability model in the family, regardless of the expansion.

We study DGMs in this class.
General template models

- probabilistic relational models
General template models

- probabilistic relational models
- multi-dynamic Bayesian networks
General template models

- probabilistic relational models
- multi-dynamic Bayesian networks
- Markov logic
General template models

- probabilistic relational models
- multi-dynamic Bayesian networks
- Markov logic

- not as much “offline” processing that can be performed based only on the template, instead heuristics must be used and one can’t be sure of the quality/cost of inference in an instantiated model.
General template models

- probabilistic relational models
- multi-dynamic Bayesian networks
- Markov logic
- not as much “offline” processing that can be performed based only on the template, instead heuristics must be used and one can’t be sure of the quality/cost of inference in an instantiated model.
- “lifted” inference schemes are popular
Template models - summary

- static graphical models
Template models - summary

- static graphical models
- plate models
Template models - summary

- static graphical models
- plate models
- dynamic graphical models
## Template models - summary

- static graphical models
- plate models
- dynamic graphical models
- general template models
Definition 1.6.1 (Independent and Identically Distributed (i.i.d.))

The stochastic process is said to be i.i.d. if the following condition holds:

\[ p(X_t = x_t, X_{t+1} = x_{t+1}, \ldots, X_{t+h} = x_{t+h}) = \prod_{i=0}^{h} p(X = x_{t+i}) \]  

for all \( t \), for all \( h \geq 0 \), for all \( x_{t:t+h} \), and for some distribution \( p(\cdot) \) that is independent of the index \( t \).

i.i.d. processes satisfy exchangeability. For any permutation \( \sigma \), we have

\[ p(X_t = x_t, X_{t+1} = x_{t+1}, \ldots, X_{t+h} = x_{t+h}) = p(X_{\sigma(t)} = x_t, X_{\sigma(t+1)} = x_{t+1}, \ldots, X_{\sigma(t+h)} = x_{t+h}) \]
Definition 1.6.2 (Stationary Stochastic Process)

The stochastic process \( \{X_t : t \geq 1\} \) is said to be (strongly) stationary if the two collections of random variables

\[
\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}
\]

and

\[
\{X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_n+h}\}
\]

have the same joint probability distributions for all \( n \) and \( h \).

In the discrete case, stationarity is equivalent to the condition

\[
P(X_{t_1} = x_1, X_{t_2} = x_2, \ldots, X_{t_n} = x_n) = P(X_{t_1+h} = x_1, X_{t_2+h} = x_2, \ldots, X_{t_n+h} = x_n)
\]

for all \( t_1, t_2, \ldots, t_n \), for all \( n > 0 \), for all \( h > 0 \), and for all \( x_i \). Every i.i.d. processes is stationary.
Definition 1.6.3 (Markov chain)

A collection of discrete-valued random variables \( \{Q_t : t \geq 1\} \) forms an \( n^{th} \)-order Markov chain if

\[
P(Q_t = q_t | Q_{t-1} = q_{t-1}, Q_{t-2} = q_{t-2}, \ldots, Q_1 = q_1) = \tag{1.11}
\]

\[
P(Q_t = q_t | Q_{t-1} = q_{t-1}, Q_{t-2} = q_{t-2}, \ldots, Q_{t-n} = q_{t-n}) \tag{1.12}
\]

for all \( t \geq 1 \), and all \( q_1, q_2, \ldots, q_t \).
does not imply that a variable is independent of future variable — on the contrary, variable might be quite dependent on future variables.
On Markov Chains

- does not imply that a variable is independent of future variable — on the contrary, variable might be quite dependent on future variables.

- only states: variable is independent of the long-past given the recent past.
On Markov Chains

- does not imply that a variable is independent of future variable — on the contrary, variable might be quite dependent on future variables.
- only states: variable is independent of the long-past given the recent past.
- Example: Bayesian networks of $n^{th}$-order Markov chains for various $n$. Left: 1st order. Middle: 2nd order. Right: 3rd order. Zero’th order is i.i.d..
On Markov Chains

- We view the event \( \{Q_t = i\} \) as if the chain is “in state \( i \) at time \( t \)” and the event \( \{Q_t = i, Q_{t+1} = j\} \) as a transition from state \( i \) to state \( j \) starting at time \( t \). This notion arises by viewing a Markov chain as a stochastic finite-state automata (FSA).
Markov chain order conversion

- An \(n^{th}\)-order Markov chain may be converted into equivalent first-order Markov chain via state equivalence:

\[
Q'_t \triangleq \{Q_t, Q_{t-1}, \ldots, Q_{t-n}\}
\]

where \(Q_t\) is an \(n^{th}\)-order Markov chain.
Markov chain order conversion

- An $n^{th}$-order Markov chain may be converted into equivalent first-order Markov chain via state equivalence:
  
  $$ Q'_t \triangleq \{ Q_t, Q_{t-1}, \ldots, Q_{t-n} \} $$

  where $Q_t$ is an $n^{th}$-order Markov chain.

- Then $Q'_t$ is a first-order Markov chain because

  $$ P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1}, Q'_{t-2} = q'_{t-2}, \ldots, Q'_1 = q'_1) $$

  $$ = P(Q_{t-n:t} = q_{t-n:t} | Q_{1:t-1} = q_{1:t-1}) $$

  $$ = P(Q_{t-n:t} = q_{t-n:t} | Q_{t-n-1:t-1} = q_{t-n-1:t-1}) $$

  $$ = P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1}) $$
Markov chain order conversion

- An $n^{th}$-order Markov chain may be converted into equivalent first-order Markov chain via state equivalence:

  $$Q'_t \triangleq \{Q_t, Q_{t-1}, \ldots, Q_{t-n}\}$$

where $Q_t$ is an $n^{th}$-order Markov chain.

- Then $Q'_t$ is a first-order Markov chain because

  $$P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1}, Q'_{t-2} = q'_{t-2}, \ldots, Q'_1 = q'_1)$$

  $$= P(Q_{t-n:t} = q_{t-n:t} | Q_{1:t-1} = q_{1:t-1})$$

  $$= P(Q_{t-n:t} = q_{t-n:t} | Q_{t-n-1:t-1} = q_{t-n-1:t-1})$$

  $$= P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1})$$

- Given a large enough state space, first-order Markov chain may represent any $n^{th}$-order Markov, but with exponential in $n$ state space.
An \(n^{th}\)-order Markov chain may be converted into equivalent first-order Markov chain via state equivalence:

\[
Q'_t \triangleq \{Q_t, Q_{t-1}, \ldots, Q_{t-n}\}
\]

where \(Q_t\) is an \(n^{th}\)-order Markov chain.

Then \(Q'_t\) is a first-order Markov chain because

\[
P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1}, Q'_{t-2} = q'_{t-2}, \ldots, Q'_1 = q'_1)
\]

\[
= P(Q_{t-n:t} = q_{t-n:t} | Q_{1:t-1} = q_{1:t-1})
\]

\[
= P(Q_{t-n:t} = q_{t-n:t} | Q_{t-n-1:t-1} = q_{t-n-1:t-1})
\]

\[
= P(Q'_t = q'_t | Q'_{t-1} = q'_{t-1})
\]

Given a large enough state space, first-order Markov chain may represent any \(n^{th}\)-order Markov, but with exponential in \(n\) state space.

Can use a single integer to represent \(Q'_t\).
Homogeneous

- The statistical evolution of a 1st-order Markov chain is determined by the state transition probabilities $a_{ij}(t) \triangleq P(Q_t = j|Q_{t-1} = i)$.
- function both of the states at successive time steps and of the current time $t$.
- sometimes, there is no dependence on $t$, and this is called time-homogeneous (or just homogeneous) because $a_{ij}(t) = a_{ij}$ for all $t$.
- The transition probabilities in homogeneous Markov chains are transition matrix $A$ where $a_{ij} \triangleq (A)_{ij}$. 
Homogeneous Markov Chain - Transition Matrix/Graph

\[ A = \begin{pmatrix}
  a_{00} & a_{01} & a_{02} & 0 & 0 & 0 & 0 & 0 \\
  0 & a_{11} & 0 & a_{13} & a_{14} & 0 & 0 & 0 \\
  0 & 0 & a_{22} & a_{23} & 0 & 0 & a_{26} & 0 \\
  0 & 0 & 0 & a_{33} & a_{34} & a_{35} & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & a_{46} & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{57} \\
  0 & a_{61} & 0 & 0 & 0 & 0 & 0 & a_{67} \\
  a_{70} & 0 & 0 & 0 & 0 & 0 & 0 & a_{77}
\end{pmatrix} \]
Homogeneous and Bayesian plate model

- A time-homogeneous 1st-order Markov chain can be viewed as a Bayesian model with plate notation.

\[ y_t | y_{t-1} \theta \]

\[ M \]

For non-random parameters, we have that \( \theta_{y_t|y_{t-1}} \) is a constant random variable.

Q: Can a 2nd order Markov chain be described by a standard plate model?
Homogeneous and Bayesian plate model

- A time-homogeneous 1st-order Markov chain can be viewed as a Bayesian model with plate notation.

- For non-random parameters, we have that $\Theta_{y_t|y_{t-1}}$ is a constant random variable.
Homogeneous and Bayesian plate model

- A time-homogeneous 1st-order Markov chain can be viewed as a Bayesian model with plate notation.

\[ \Theta_{y_t | y_{t-1}} \]

- For non-random parameters, we have that \( \Theta_{y_t | y_{t-1}} \) is a constant random variable.

- Q: Can a 2nd order Markov chain be described by a standard plate model?
State Types

- A state $i$ is said to be **transient** if, after visiting the state, it is possible for it never to be visited again, i.e.,:

  $$p(Q_n = i \text{ for some } n > t | Q_t = i) < 1.$$  

- A state $i$ is said to be **null-recurrent** if it is not transient but the expected return time is infinite, i.e.,

  $$E[\min\{n > t : Q_n = i\} | Q_t = i] = \infty$$

(1.13)

- A state is **positive-recurrent** if it is not transient and the expected return time to that state is finite

- Under finite number of states, a state can only be either transient or positive-recurrent.
State Types

- State types, example:

```
q0 q1
q2
q3
```

- $q_0$ is transient while states $q_1$, $q_2$, and $q_3$ are positive-recurrent.
State Types

- State types, example:

  ![State Diagram]

  - $q_0$ is transient while states $q_1$, $q_2$, and $q_3$ are positive-recurrent.
If $Q_t$ is a time-homogeneous stationary first-order Markov chain then:

$$P(Q_{t_1} = q_1, Q_{t_2} = q_2, \ldots, Q_{t_n} = q_n)$$

$$= P(Q_{t_1+h} = q_1, Q_{t_2+h} = q_2, \ldots, Q_{t_n+h} = q_n)$$

for all $t_i$, $h$, $n$, and $q_i$. 

\[ \text{(1.14)} \]

\[ \text{(1.15)} \]
Stationarity Again

If $Q_t$ is a time-homogeneous stationary first-order Markov chain then:

$$P(Q_{t_1} = q_1, Q_{t_2} = q_2, \ldots, Q_{t_n} = q_n)$$ (1.14)

$$= P(Q_{t_1+h} = q_1, Q_{t_2+h} = q_2, \ldots, Q_{t_n+h} = q_n)$$ (1.15)

for all $t_i$, $h$, $n$, and $q_i$.

Using the first order Markov property, the above can be written as:

$$P(Q_{t_n} = q_n | Q_{t_{n-1}} = q_{n-1})$$ (1.16)

$$P(Q_{t_{n-1}} = q_{n-1} | Q_{t_{n-2}} = q_{n-2})$$ (1.17)

$$\ldots P(Q_{t_2} = q_2 | Q_{t_1} = q_1) P(Q_{t_1} = q_1)$$ (1.18)

$$= P(Q_{t_n+h} = q_n | Q_{t_{n-1}+h} = q_{n-1})$$ (1.19)

$$P(Q_{t_{n-1}+h} = q_{n-1} | Q_{t_{n-2}+h} = q_{n-2})$$ (1.20)

$$\ldots P(Q_{t_2+h} = q_2 | Q_{t_1+h} = q_1) P(Q_{t_1+h} = q_1)$$ (1.21)
Stationarity Again

Therefore, a time-homogeneous Markov chain is stationary iff

\[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \text{ for all } q \in D_Q \]
Stationarity Again

- Therefore, a time-homogeneous Markov chain is stationary iff
  \[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \text{ for all } q \in D_Q \]
- a stationary distribution has the property that \( \xi A = \xi \) implying that \( \xi \) must be a left eigenvector of the transition matrix \( A \).
Therefore, a time-homogeneous Markov chain is stationary iff
\[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \]
for all \( q \in D_Q \)

a stationary distribution has the property that \( \xi A = \xi \) implying that \( \xi \) must be a left eigenvector of the transition matrix \( A \).

Example: let \( p_1 = [0.5, 0.5] \) be the current distribution over 2-state Markov chain.
Therefore, a time-homogeneous Markov chain is stationary iff
\[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \]
for all \( q \in D_Q \).

A stationary distribution has the property that \( \xi A = \xi \) implying that
\( \xi \) must be a left eigenvector of the transition matrix \( A \).

Example: let \( p_1 = [\cdot.5, \cdot.5] \) be the current distribution over 2-state
Markov chain. Let \( A_1 = [\cdot.3, \cdot.7; \cdot.7, \cdot.3] \) be the transition matrix.
Therefore, a time-homogeneous Markov chain is stationary iff
\[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \]
for all \( q \in D_Q \).

A stationary distribution has the property that \( \xi A = \xi \) implying that \( \xi \) must be a left eigenvector of the transition matrix \( A \).

Example: let \( p_1 = [.5, .5] \) be the current distribution over 2-state Markov chain. Let \( A_1 = [.3, .7; .7, .3] \) be the transition matrix. The Markov chain is stationary since \( p_1 A_1 = p_1 \).
Therefore, a time-homogeneous Markov chain is stationary iff
\[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \text{ for all } q \in D_Q \]

a stationary distribution has the property that \( \xi A = \xi \) implying that \( \xi \) must be a left eigenvector of the transition matrix \( A \).

Example: let \( p_1 = [0.5, 0.5] \) be the current distribution over 2-state Markov chain. Let \( A_1 = [0.3, 0.7; 0.7, 0.3] \) be the transition matrix. The Markov chain is stationary since \( p_1 A_1 = p_1 \). If the current distribution is \( p_2 = [0.4, 0.6] \), however, then \( p_2 A_1 \neq p_2 \), so the chain is no longer stationary (even with same transition matrix).
Stationarity Again

- Therefore, a time-homogeneous Markov chain is stationary iff
  \[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \]
  for all \( q \in \mathbb{D}_Q \).

- A stationary distribution has the property that \( \xi A = \xi \) implying that
  \( \xi \) must be a left eigenvector of the transition matrix \( A \).

- Example: let \( p_1 = [.5, .5] \) be the current distribution over a 2-state
  Markov chain. Let \( A_1 = [.3, .7; .7, .3] \) be the transition matrix. The
  Markov chain is stationary since \( p_1 A_1 = p_1 \). If the current
  distribution is \( p_2 = [.4, .6] \), however, then \( p_2 A_1 \neq p_2 \), so the chain is
  no longer stationary (even with same transition matrix).

- Can exist more than 1 stationary distribution.
Stationarity Again

- Therefore, a time-homogeneous Markov chain is stationary iff
  \[ P(Q_{t_1} = q) = P(Q_{t_1+h} = q) = P(Q_t = q) \]
  for all \( q \in D_Q \).
- A stationary distribution has the property that \( \xi A = \xi \) implying that
  \( \xi \) must be a left eigenvector of the transition matrix \( A \).
- Example: let \( p_1 = [.5, .5] \) be the current distribution over 2-state
  Markov chain. Let \( A_1 = [.3, .7; .7, .3] \) be the transition matrix. The
  Markov chain is stationary since \( p_1 A_1 = p_1 \). If the current
  distribution is \( p_2 = [.4, .6] \), however, then \( p_2 A_1 \neq p_2 \), so the chain is
  no longer stationary (even with same transition matrix).
- Can exist more than 1 stationary distribution.
- Exercise: Must there exist a stationary distribution in a
  time-homogeneous Markov chain?
Stationarity implies Homogeneity

If stationary, then

\[ P(Q_t = i, Q_{t-1} = j) = P(Q_{t-1} = i, Q_{t-2} = j) \]  \hspace{1cm} (1.22)

and

\[ P(Q_t = i) = P(Q_{t-1} = i) \]  \hspace{1cm} (1.23)

Therefore,

\[ a_{ij}(t) = \frac{P(Q_t = i, Q_{t-1} = j)}{P(Q_{t-1} = j)} \]  \hspace{1cm} (1.24)

\[ = \frac{P(Q_{t-1} = i, Q_{t-2} = j)}{P(Q_{t-2} = j)} \]  \hspace{1cm} (1.25)

\[ = a_{ij}(t - 1) \]  \hspace{1cm} (1.26)

and by induction \( a_{ij}(t) = a_{ij}(t + \tau) \) for all \( \tau \).
Example

- Let $A_t = [.3, .7; .7, .3]$ when $t$ is even and $A_t = [.4, .6; .6, .4]$ when $t$ is odd, so chain is inhomogeneous.
Example

- Let $A_t = [0.3, 0.7; 0.7, 0.3]$ when $t$ is even and $A_t = [0.4, 0.6; 0.6, 0.4]$ when $t$ is odd, so chain is inhomogeneous.

- If the current state distribution is $p = [0.5, 0.5]$, then $pA_t = p$ for $t$ both even and odd.
Example

- Let $A_t = [.3, .7; .7, .3]$ when $t$ is even and $A_t = [.4, .6; .6, .4]$ when $t$ is odd, so chain is inhomogeneous.
- If the current state distribution is $p = [.5, .5]$, then $pA_t = p$ for $t$ both even and odd.
- Is this stationary?
Example

- Let $A_t = [.3, .7; .7, .3]$ when $t$ is even and $A_t = [.4, .6; .6, .4]$ when $t$ is odd, so chain is inhomogeneous.
- If the current state distribution is $p = [.5, .5]$, then $pA_t = p$ for $t$ both even and odd.
- Is this stationary?
- Note that this is not a stationary distribution. When $t$ is even, we have that $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.3$ but when $t$ is odd, $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.4$, so the chain does not exhibit a stationary distribution according to the definition.
Example

- Let $A_t = [.3, .7; .7, .3]$ when $t$ is even and $A_t = [.4, .6; .6, .4]$ when $t$ is odd, so chain is inhomogeneous.
- If the current state distribution is $p = [.5, .5]$, then $pA_t = p$ for $t$ both even and odd.
- Is this stationary?
- Note that this is not a stationary distribution. When $t$ is even, we have that $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.3$ but when $t$ is odd, $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.4$, so the chain does not exhibit a stationary distribution according to the definition.
- So the aforementioned criterion for stationary ($\xi A = \xi$) requires a homogeneous chain. $\xi A_t = \xi$ alone does not guarantee stationarity.
Example

- Let $A_t = [0.3, 0.7; 0.7, 0.3]$ when $t$ is even and $A_t = [0.4, 0.6; 0.6, 0.4]$ when $t$ is odd, so chain is inhomogeneous.
- If the current state distribution is $p = [0.5, 0.5]$, then $pA_t = p$ for $t$ both even and odd.
- Is this stationary?
- Note that this is not a stationary distribution. When $t$ is even, we have that $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.3$ but when $t$ is odd, $p(Q_t = 0, Q_{t+1} = 1) = 0.5 \times 0.4$, so the chain does not exhibit a stationary distribution according to the definition.
- So the aforementioned criterion for stationary ($\xi A = \xi$) requires a homogeneous chain. $\xi A_t = \xi$ alone does not guarantee stationarity.
- In general, important to realize that stationarity and homogeneity of a Markov chain (or any DGM) are distinct properties. If stationary, then homogeneous. If homogeneous, then might or might not be stationary.
Definition 1.6.4 (period)

The **period** \( d(i) \) of a state is defined as \( d(i) = \gcd \{ n : a_{ii}^n > 0 \} \), the greatest common divisor of the time intervals where return is possible. The state \( i \) is periodic if \( d(i) > 1 \) and aperiodic if \( d(i) = 1 \).
Definition 1.6.4 (period)

The **period** $d(i)$ of a state is defined as $d(i) = \gcd \{ n : a_{ii}^n > 0 \}$, the greatest common divisor of the time intervals where return is possible. The state $i$ is periodic if $d(i) > 1$ and aperiodic if $d(i) = 1$.

Ex 1: If all states have self-loops, period is 1. Ex 2: concentric loops, in one case loops of length 3,5 (period is 1) another case of length 3,6 (period is 3).
Definition 1.6.4 (period)

The period $d(i)$ of a state is defined as $d(i) = \gcd \{ n : a_{ii}^n > 0 \}$, the greatest common divisor of the time intervals where return is possible. The state $i$ is periodic if $d(i) > 1$ and aperiodic if $d(i) = 1$.

Ex 1: If all states have self-loops, period is 1. Ex 2: concentric loops, in one case loops of length 3,5 (period is 1) another case of length 3,6 (period is 3).

- Period is a probabilistic concept - if period is $> 1$, then there will be certain times where state is impossible (e.g., 3,6 case above). If period = 1 (i.e., return times are co-prime), eventually, all states will have positive probability at successive time intervals (e.g., $3\mathbb{N} + 5\mathbb{N} = \mathbb{N} \cap \{ i : i > k, k \in \mathbb{N} \}$ in case above).
Stationarity and probability flow

- Probability “flow” can help to give intuition behind stationary distribution.
- For stationarity, probability outflow must equal probability inflow for every state!

Stationary, or $\xi A = \xi$, implies that for all $i$

$$\xi_i = \sum_j \xi_j a_{ji} \quad (1.27)$$

or equivalently,

$$\xi_i (1 - a_{ii}) = \sum_{j\neq i} \xi_j a_{ji} \quad (1.28)$$

which is the same as

$$\sum_{j\neq i} \xi_i a_{ij} = \sum_{j\neq i} \xi_j a_{ji} \quad (1.29)$$

Left side interpreted as probability flow out of state $i$, right side interpreted as flow into state $i$. 

Transition Matrix

Let $A$ be a transition matrix for a 1st-order time-homogeneous Markov chain.

**Theorem 1.6.5**

The transition matrix $A = [a_{ij}]_{i,j}$ is a stochastic matrix, meaning:

(a) $A$ has non-negative entries, so that $a_{ij} \geq 0 \ \forall i, j$

(b) $A$ has row-sums equal to one $\sum_j a_{ij} = 1$. 
Chapman-Kolmogorov equations
Computing transition probabilities over $k$ steps

Given time-homogeneous 1st-order chain, let

$$a_{ij}^k \triangleq p(Q_{t+k} = j|Q_t = i),$$

meaning probability of transitioning from state $i$ to state $j$ in exactly $k$ steps. Then

$$a_{ij}^2 = p(Q_{t+2} = j|Q_t = i)$$

$$= \sum_{\ell} p(Q_{t+2} = j|Q_{t+1} = \ell)p(Q_{t+1} = \ell|Q_t = i)$$

$$= \sum_{\ell} a_{i\ell}a_{\ell j}$$
Chapman-Kolmogorov equations (cont. II)
Computing transition probabilities over $k$ steps

We can generalize this immediately to give:

\[ a_{ij}^k = \sum_{\ell} a_{i\ell}^m a_{\ell j}^n \]  \hspace{1cm} (1.34)

where $m, n \geq 0$ with $k = m + n$. Can be generalized to $n^{th}$-order and/or time-inhomogeneous chains.

Example, we can define:

\[ a_{ij}(t, t + \tau) = p(Q_{t+\tau} = i | Q_t = i) \]  \hspace{1cm} (1.35)

for $\tau > 0$. 
Communication and Partitions

Theorem 1.6.6

If \( i \leftrightarrow j \) then

(a) \( i \) and \( j \) have same period

(b) \( i \) is transient iff \( j \) is transient

(c) \( i \) is null-recurrent iff \( j \) is null-recurrent

- State \( i \) communicates with state \( j \) \((i \rightarrow j)\) if \( a_{ij}^k > 0 \) for some \( k \geq 0 \). States intercommunicate if \( i \rightarrow j \) and \( j \rightarrow i \), written \( i \leftrightarrow j \).
- Set of states \( C \) is closed if \( a_{ij} = 0 \) for \( i \in C \) and \( j \notin C \).
- Set of states \( C \) is irreducible if \( i \leftrightarrow j \) for all \( i, j \in C \).
Theorem 1.6.7

Set of states can be partitioned uniquely as

\[ S = T \cup C_1 \cup C_2 \cup \ldots \]  

(1.36)

where \( T \) is the set of transient states, and \( C_i \) are sets of irreducible closed sets of recurrent states.

Intuition: When we run a Markov chain (by choosing the next state according to \( a_{ij} \)), states are either such that they eventually are impossible to reach, or we get into a “rut” where only a subset of the states are reachable, and we can never break out of that rut. It might be that no states are transient and there is only one \( C_i \) (e.g., a matrix with all strictly positive entries). The entire chain is then irreducible.
Example 1.6.8

\[
A = \begin{pmatrix}
1/2 & 1/2 & 0 & 0 & 0 & 0 \\
1/4 & 3/4 & 0 & 0 & 0 & 0 \\
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 \\
1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 \\
\end{pmatrix}
\]

(1.37)

States \(\{1, 2\}\) and \(\{5, 6\}\) are irreducible and closed and recurrent. States 3 and 4 are transient since 3 \(\rightarrow\) 4 \(\rightarrow\) 6 but we can’t get back from 6. The period of all states is 1 (since \(a_{ii} > 0\)).

HW/Challenge Problem: Derive additional Markov chain matrices such that you have variants of the above (i.e., a case with no \(T\), a case with only one \(C\), a case with \(> 1\) \(C\), and so on).
One aspect of temporal models we wish to study is how does information transfer decay over time. I.e., for a given fixed order, how bad is the past⊥future|present property. Example: protein folding, many long term interactions, how poor would it be to make such a Markov assumption? Another example: statistical machine translation (permutations, expansions, contractions).

Chapman-Kolmogorov allows us to begin to look at that. Let $\xi^t$ be the distribution at time $t$, so that $\xi^{t+1} = \xi^t A$. Then

**Lemma 1.6.9**

$$\xi^{t+\tau} = \xi^t A^\tau$$

where $A^\tau = A \times A \times \cdots \times A$ is the $\tau^{th}$ power of $\tau$.

If $\xi$ is a stationary distribution, then $\xi = \xi A^\tau$ for any $\tau > 1$. 

Note: For every component $C_i$ in the above decomposition, there is a unique stationary distribution. If the entire chain is irreducible, there is a single unique stationary distribution. In fact, we have

**Theorem 1.6.10 (Perron-Frobenius)**

*If $A$ is the transition matrix of a finite irreducible chain with period $d$, then*

(a) $\lambda_1 = 1$ is an eigenvalue of $A$

(b) the $d$ complex roots of unity, $\lambda_k = W_d^k$, $k = 1, \ldots, d$ with $W_d = \exp(2\pi \sqrt{-1}/d)$ are eigenvalues of $A$

(c) the remaining eigenvalues $\lambda_{d+1}, \ldots, \lambda_N$ satisfy $|\lambda_j| < 1$.\]
When the eigenvalues $\lambda_1, \ldots, \lambda_N$ are distinct, there exists a matrix $B$ where

$$A^k = B^{-1} \Lambda^n B = B^{-1} \begin{pmatrix} \lambda_1^n & 0 & \cdots & 0 \\ 0 & \lambda_2^n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N^n \end{pmatrix} B$$ (1.39)

rows of $B$ are left eigenvectors of $A$. Therefore, if $d = 1$, we have

$$A^n \rightarrow B^{-1} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} B \text{ as } n \rightarrow \infty$$ (1.40)

We also have the following theorem
Transition Matrices Factorization (cont. III)

Theorem 1.6.11

If \( A \) is the transition matrix of a finite irreducible chain with period \( d \), then

\[
A^k \rightarrow 1\xi
\]  \hspace{1cm} (1.41)

where \( \xi \) is the unique stationary distribution associated with the Markov chain, and the rate of convergence, moreover, is geometric.

So this means that, regardless of the initial distribution where we start, we will approach a (unique if irreducible) stationary distribution of the Markov chain, and we'll do so quite fast. That is, we will forget from where we started from.
Clearly, an $k$th-order Markov chain is a $k + 1$-tree, so if we were to perform exact inference (with the goal a clique or subclique), the cost would be $O(r^{k+1})$.

The structure (and resulting application) of the Markov chain is entirely encoded in the transition matrix $[a_{ij}]_{i,j}$.

Many applications of Markov chains

- Random walk on a integer point line, with $a_{ij} = 0$ unless $j = i \pm 1$.
- Markov Chain Monte Carlo - goal is to generate samples of some difficult distribution $p(x)$. $p(x)$ corresponds to a stationary distribution of the Markov chain.
Stochastic Finite State Automata - any discrete time series could be modeled by this. E.g., Language ($n$-gram modeling in speech/natural language processing).

Various Genomic & Proteomic Sequencing

And so on.

Many applications listed in Grimmett & Stirzaker text.

We will revisit Markov chains repeatedly in this course, since many of the properties of DGMs have properties generalized from a Markov chain.
Sources for Today’s Lecture

- “doc.pdf” sections 8.1 - 8.3