Language Technologies Institute

Carnegie Mellon University

Multimodal Affective Computing

Lecture 9: Discriminative Predictive Modeling

Louis-Philippe Morency
Jeffrey Girard

Originally developed with help from
Stefan Scherer and Tadas Baltrušaitis

Outline

- Dynamic Bayesian Network
  - Hidden Markov Models
  - Factorial and coupled HMMs
- Markov Random Fields
  - Unary, binary and clique potentials
  - Factor graph representation
- Multimodal Machine Learning
  - Core Challenges: Representation, Alignment, Fusion, Translation and Co-Learning
- Discriminative Graphical Models
  - Logistic classifier
  - Conditional random fields
  - L1 and L2 regularization
- Evaluation methods and error measures

Upcoming Deadlines and Course Schedule

- **Thursday, April 4th 4:30pm-6pm**
  - Midterm presentations
- **Sunday, April 7th at 11:59pm**
  - Midterm report deadline
- **Thursday, May 2nd 4:30pm-6pm**
  - Final presentations
- **Tuesday, May 7th at 11:30pm**
  - Final report deadline

**** No reading assignments for Weeks 12 and 13 ****

Midterm Reports – Sunday April 7th 11:59pm ET

Main report sections:

- Background and Motivation (1/3 to 1/2 page)
- Literature Review (1/2 to 1 page)
- Data Description and New Annotations (about 1 page)
- Problem Conceptualization (about 1 page)
- Statistical Analysis (1 to 2 pages)
- Next Steps about (1/2 page)
- Appendix: Team Collaboration (about 1/2 page)

Maximum report length: 7 pages

Midterm Presentations – Thursday April 4th

Main presentation sections:

- Motivation, research problem and dataset
- New annotations
- Problem conceptualization
- Statistical analysis

Presentation instructions:

- Maximum length: 8 minutes
- All teammates should participate
- Followed by questions and feedback forms

Upcoming Lectures

Classes	Tuesday	Thursday
Week 10 3/19 & 3/21 *midterm homework*	Probabilistic predictive modeling • Probabilistic graphical models • Bayesian networks and Naïve Bayes classifier • Dynamic Bayesian networks and HMMs	Discussion (probabilistic) • Vaibhav • Vasu Sharma
Week 11 3/26 & 3/28	Discriminative predictive modeling • Markov random fields • Factor graph representation • Discriminative graphical models	Discussion (discriminative) • Vaibhav • Vasu Sharma
Week 12 4/02 & 4/04 *midterm report*	Multimodal deep representations • Multimodal joint representations • Coordinated representations • Temporal representations	Midterm presentations
Week 13 4/09 & 4/11	Multimodal alignment and fusion • Attention and modality alignment • Temporal and multimodal fusion	NO CLASS
Week 14 4/16 & 4/18	Multimodal Behavior Generation • Guest lecture: Prof. Nakano • Generation based on user's attitude • Robot and virtual humans	Discussion (medical) • Jiang Liu • Mahmoud Al Ismail
Week 15 4/23 & 4/25	Multimodal applications • Assessment in the clinical process • Biomarkers and behavioral indicators • Validation in the medical sciences	Discussion (educational) • Mingtong Zhang • Ankit Shah
Week 16 4/30 & 5/02 *final report*	Final presentations	Final presentations

Dynamic Bayesian Networks

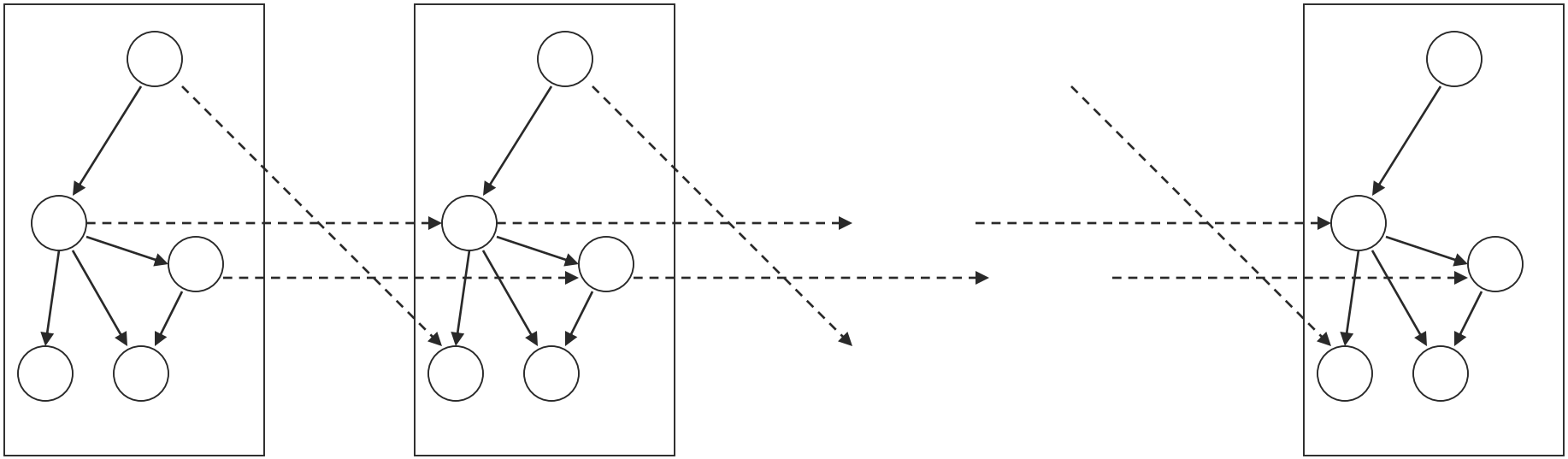
Dynamic Bayesian Network (DBN)

- Bayesian network with time-series to represent temporal dependencies.
- Dynamically changing or evolving over time.
- Directed graphical model of stochastic processes.
- Especially aiming at time series modeling.

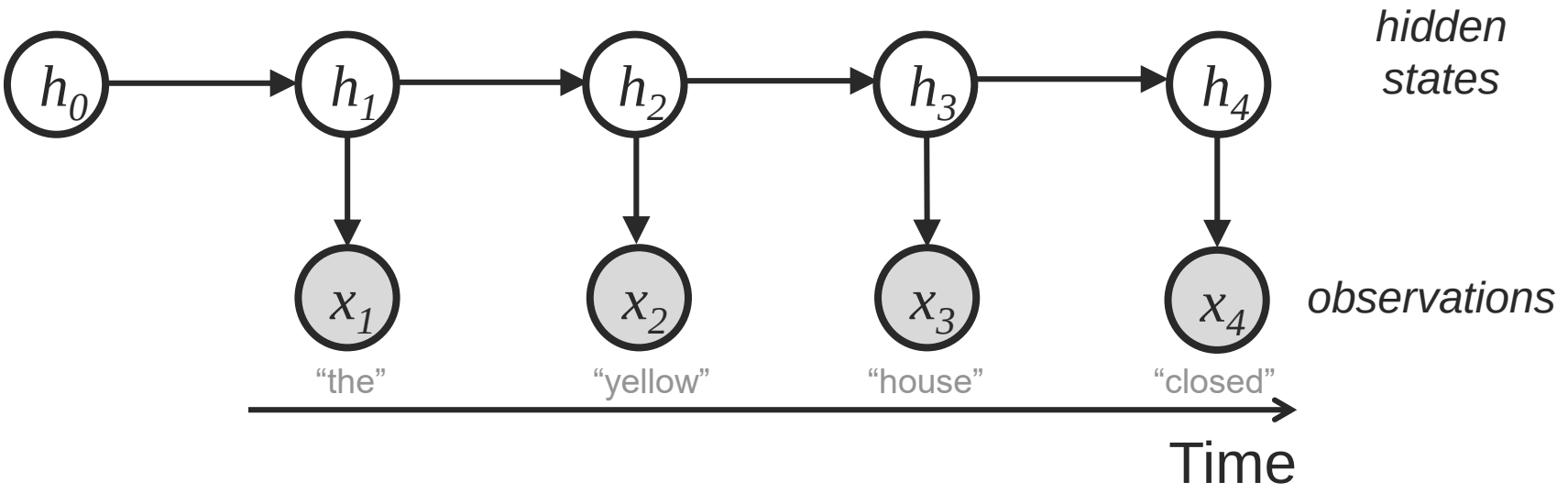
- Satisfying the Markovian condition:

  The state of a system at time t depends only on its immediate past state at time t-1.

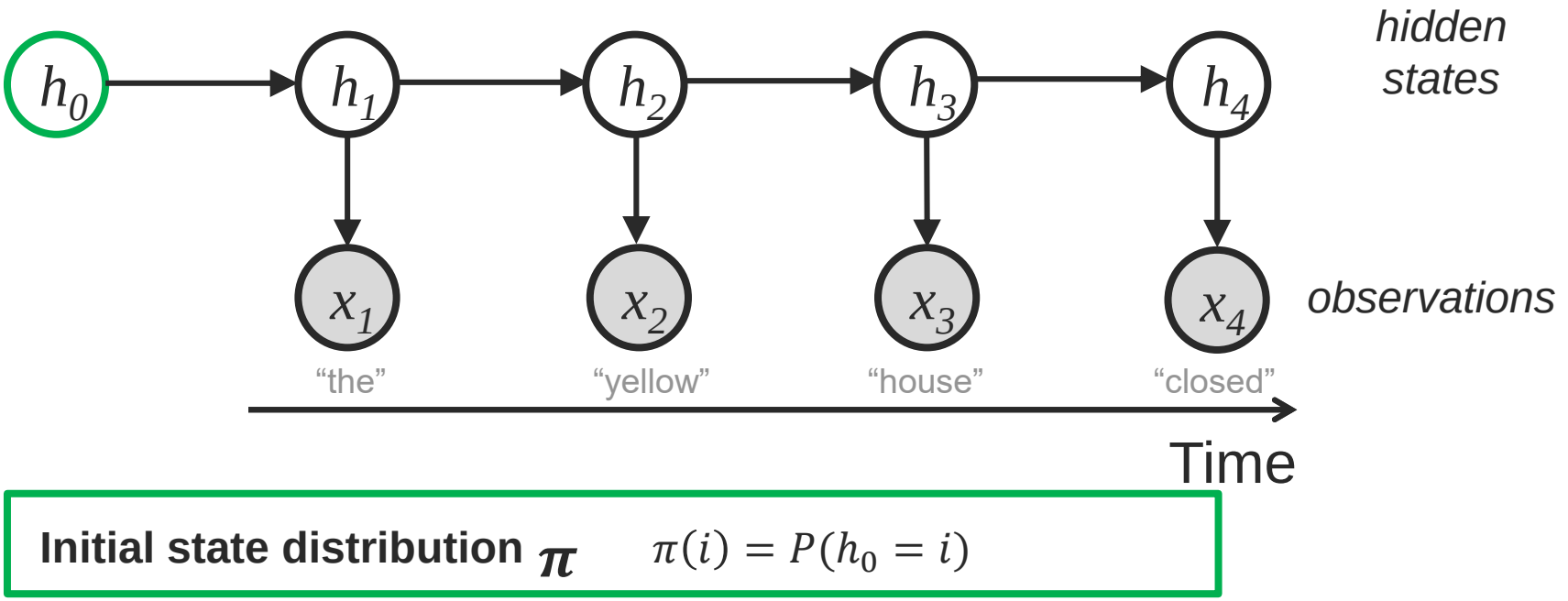
Dynamic Bayesian Network (DBN)



Hidden Markov Models

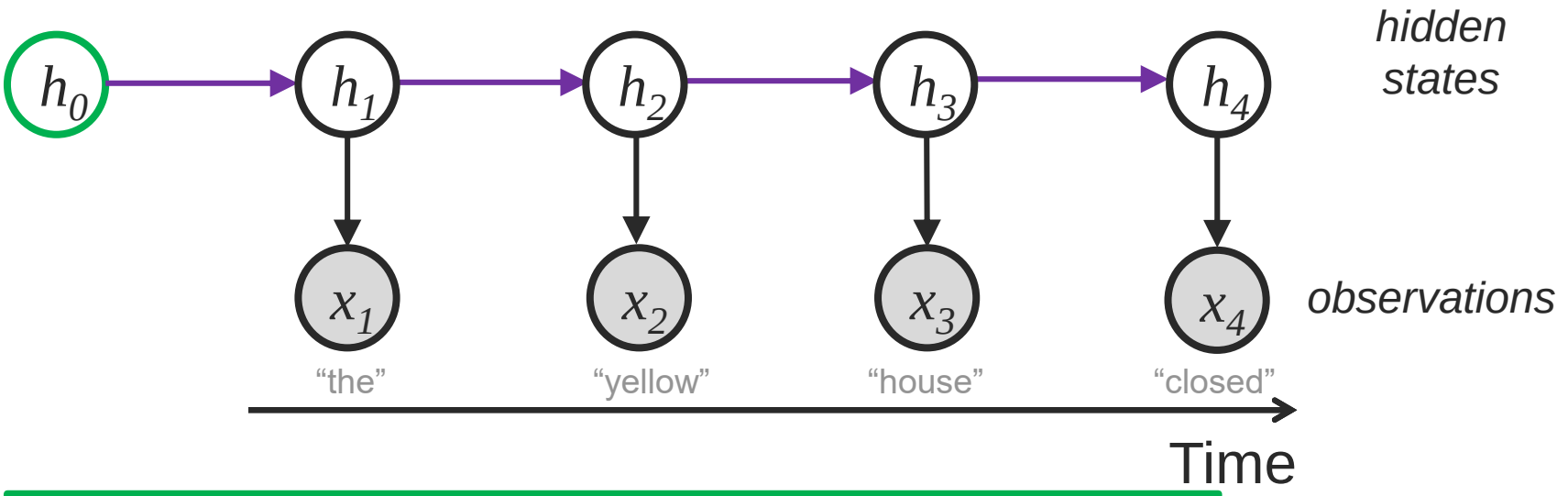


Hidden Markov Models



Initial state distribution $\boldsymbol{\pi}$ $\pi(i) = P(h_0 = i)$

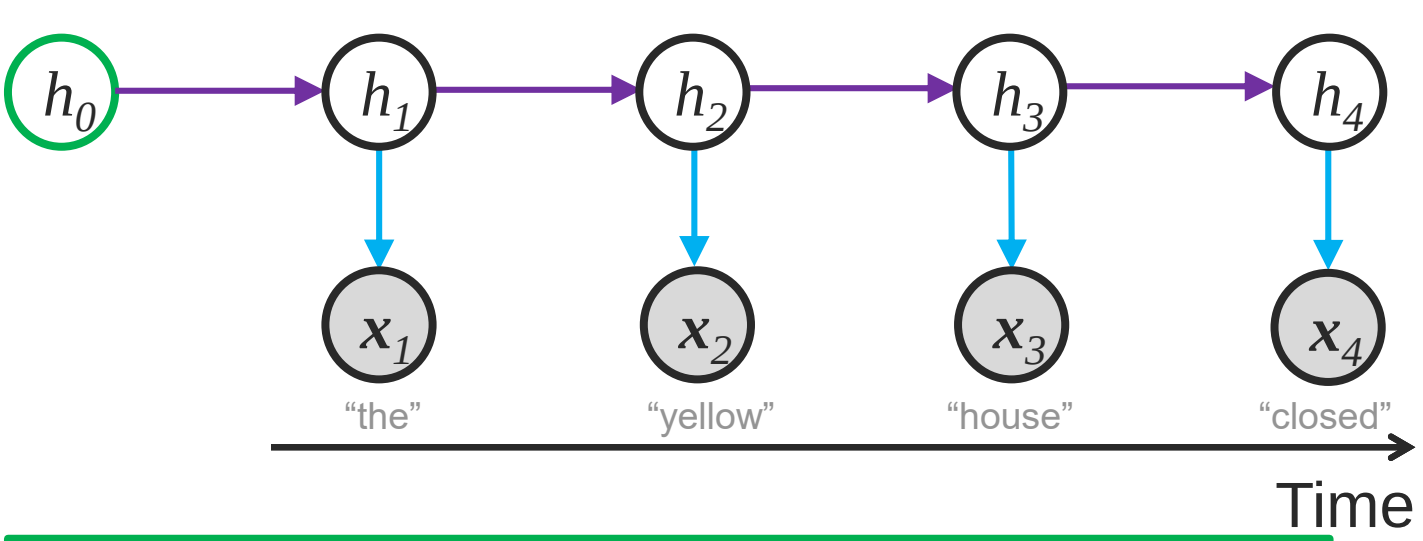
Hidden Markov Models



Initial state distribution $\boldsymbol{\pi}$ $\quad \pi(i) = P(h_0 = i)$

Transition probabilities $\boldsymbol{A}$ $\quad \mathrm{a}(i,j) = P(h_t = i | h_{t-1} = j)$

Hidden Markov Models

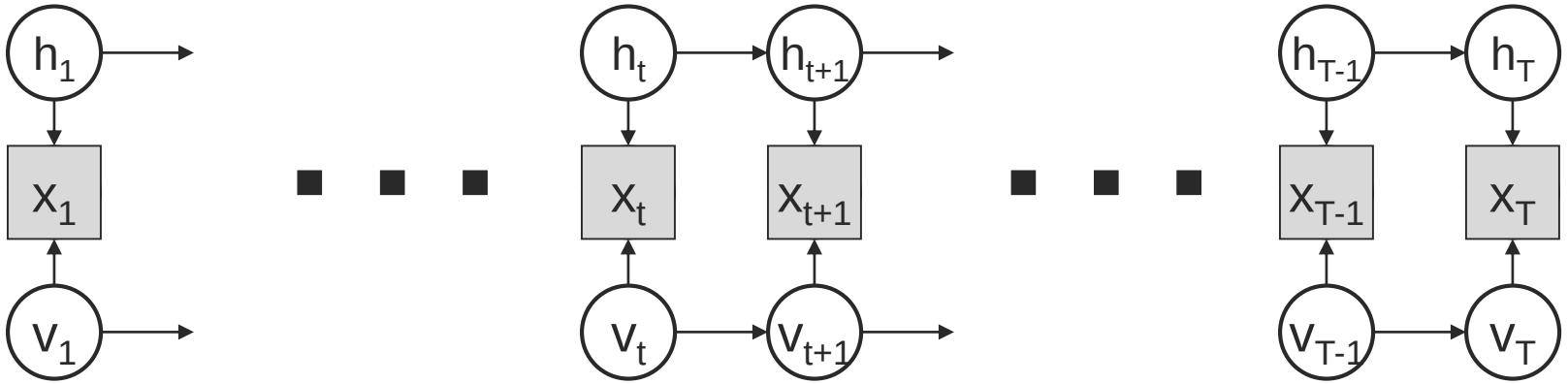


Initial state distribution $\boldsymbol{\pi}$ $\quad \pi(i) = P(h_0 = i)$

Transition probabilities $\boldsymbol{A}$ $\quad \mathrm{a}(i,j) = P(h_t = i | h_{t-1} = j)$

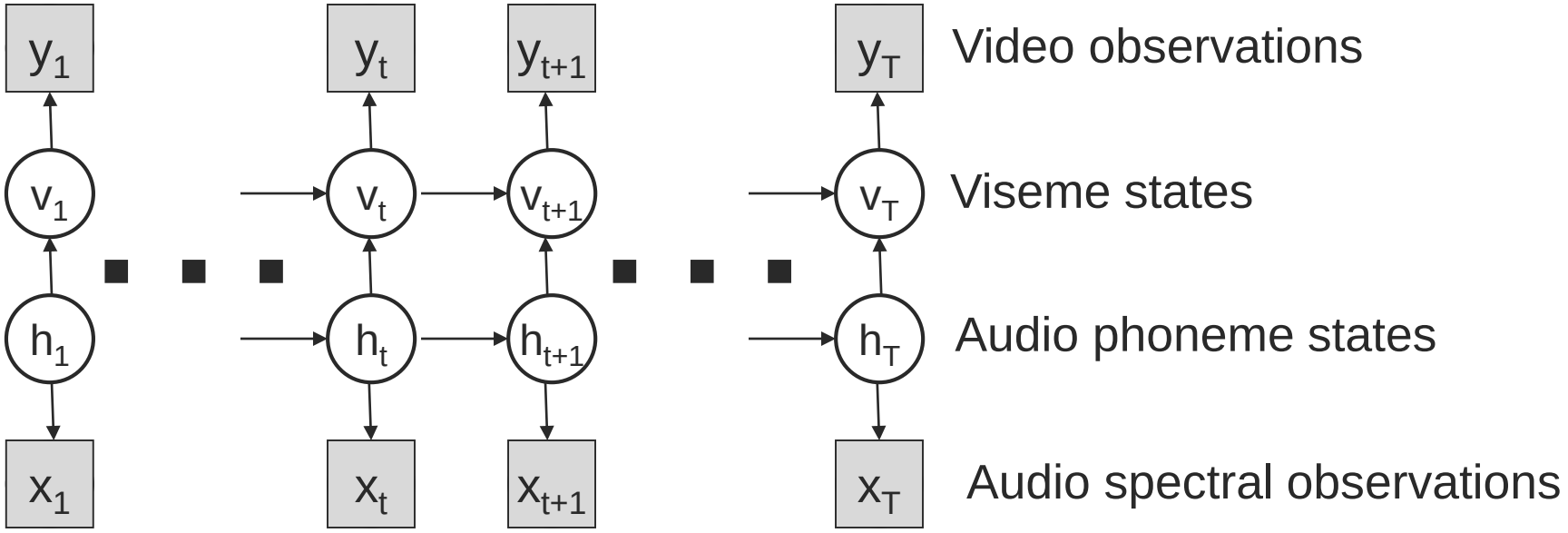
Emission Probabilities $\boldsymbol{B}$ $\quad b_t(i) = P(\boldsymbol{x}_t | h_t = i)$

Factorial HMM



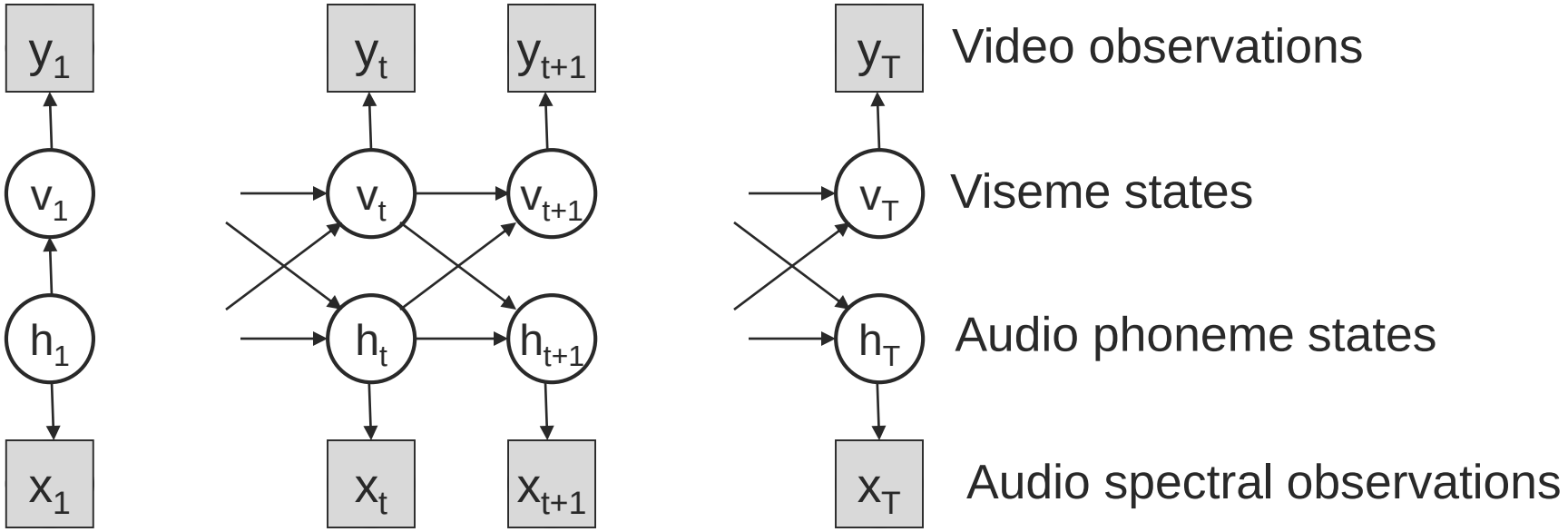
- Factorial HMM:
  - h_t and v_t represent two different types of background information, each with its own history
  - Observations x_t depend on both hidden processes
- Model parameters:
  - $p(v_{t+1}|v_t)$, $p(h_{t+1}|h_t)$, $p(x_t|h_t,v_t)$

The Boltzmann Zipper



- Both streams have a "memory" (h_t and v_t)
- Model parameters:
  - $p(h_{t+1}|h_t)$, $p(x_t|h_t)$
  - $p(v_{t+1}|v_t,h_{t+1})$, $p(y_t|h_t)$

The Coupled HMM



- Advantage over Boltzmann Zipper: More flexible, because neither vision nor sound is "privileged" over the other.
  - $p(h_{t+1}|v_t,h_t)$, $p(x_t|h_t)$
  - $p(v_{t+1}|v_t,h_t)$, $p(y_t|h_t)$

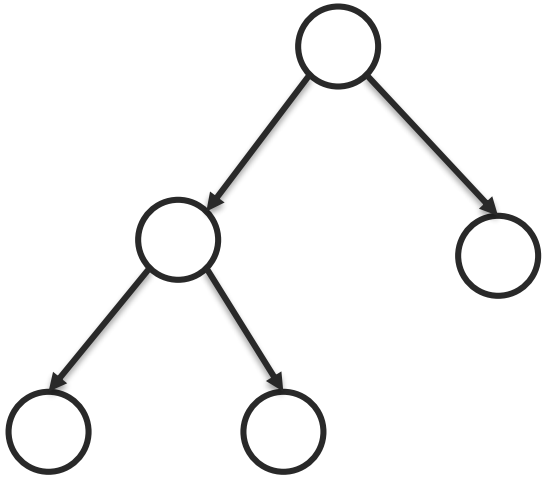
Learning (Dynamic) Bayesian Networks

- Multiple techniques exist to learn the model parameters based on data
  - Maximum likelihood estimator
  - Bayesian estimator, which allows to include prior information
- Python libraries:
  - http://pgmpy.org/
  - http://www.bayespy.org
  - https://pomegranate.readthedocs.io/en/latest/

Markov Random Field

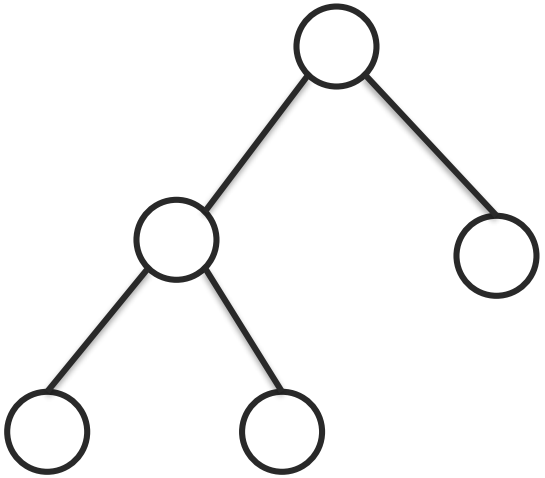
Two Main Types of Graphical Models

Bayesian networks (last week)



- Directed acyclic graph
- Conditional dependencies

Markov Models



- Undirected graphical model
- Cyclic dependencies

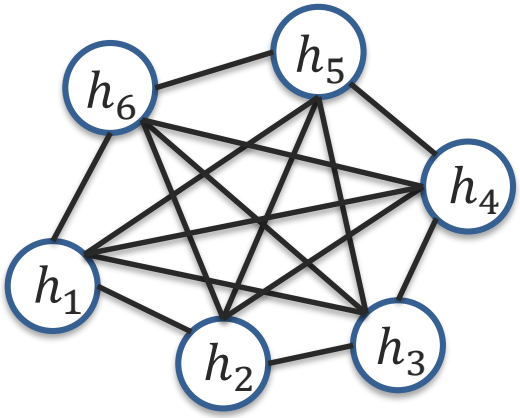
Markov Random Fields

$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)}$$

Potential of this variable assignment $\boldsymbol{h}$

Potential of all possible variable assignments $\boldsymbol{h}'$

- Set of random variables H having a Markov property described by undirected graph



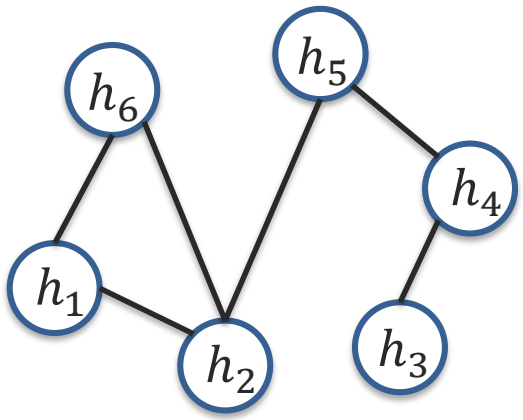
$$\Phi(\boldsymbol{h}; \theta) = \prod_k \phi_k(\boldsymbol{h}; \theta_k)$$

Potential functions $\phi_k(\boldsymbol{h}; \theta) > 0$

Markov Random Fields – Graphical Model

$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)}$$

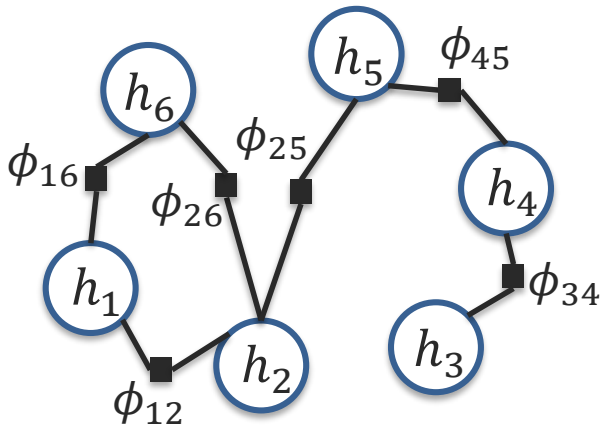
$$\begin{aligned}\Phi(\boldsymbol{h}; \theta) = \; & \phi_{12}(h_1, h_2; \theta_{12}) \times \\ & \phi_{16}(h_1, h_6; \theta_{16}) \times \\ & \phi_{26}(h_2, h_6; \theta_{26}) \times \\ & \phi_{25}(h_2, h_5; \theta_{25}) \times \\ & \phi_{45}(h_4, h_5; \theta_{45}) \times \\ & \phi_{34}(h_3, h_4; \theta_{34})\end{aligned}$$



Markov Random Fields: Factor Graphs

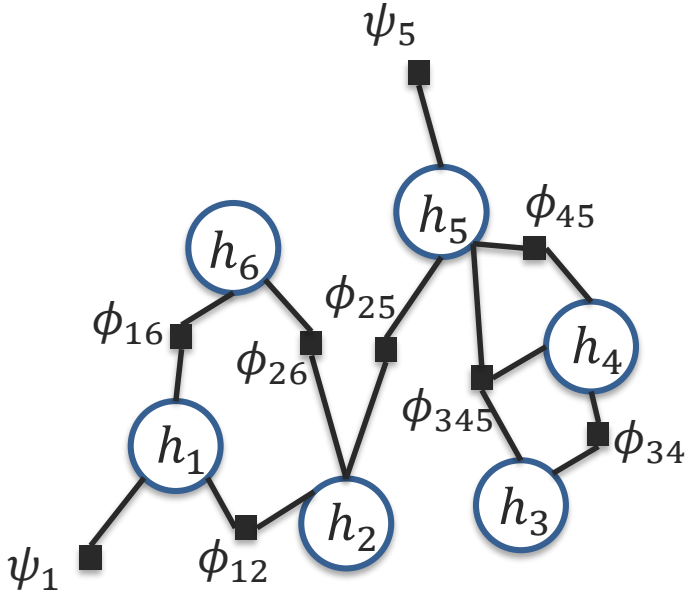
$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)}$$

$$\begin{aligned}\Phi(\boldsymbol{h}; \theta) = {} & \phi_{12}(h_1, h_2; \theta_{12}) \times \\ & \phi_{16}(h_1, h_6; \theta_{16}) \times \\ & \phi_{26}(h_2, h_6; \theta_{26}) \times \\ & \phi_{25}(h_2, h_5; \theta_{25}) \times \\ & \phi_{45}(h_4, h_5; \theta_{45}) \times \\ & \phi_{34}(h_3, h_4; \theta_{34})\end{aligned}$$



Markov Random Fields (Factor Graphs)

$$p(H = \boldsymbol{h}, \boldsymbol{x}; \theta) = \frac{\Phi(\boldsymbol{h}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}', \boldsymbol{x}; \theta)}$$



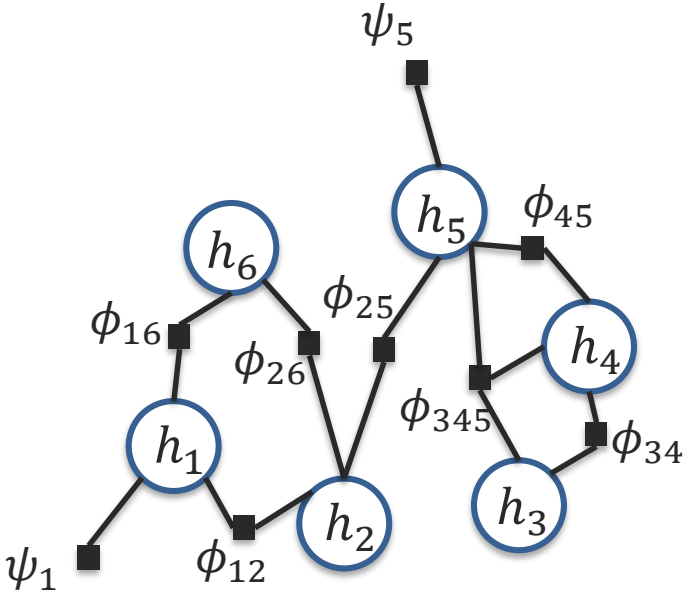
$$\Phi(\boldsymbol{h}; \theta) = \phi_{12}(h_1, h_2; \theta_{12}) \times$$
$$\phi_{16}(h_1, h_6; \theta_{16}) \times$$
$$\phi_{26}(h_2, h_6; \theta_{26}) \times$$
$$\phi_{25}(h_2, h_5; \theta_{25}) \times$$
$$\phi_{45}(h_4, h_5; \theta_{45}) \times$$
$$\phi_{34}(h_3, h_4; \theta_{34}) \times$$
$$\psi_1(h_1, \boldsymbol{x}; \theta_1) \times \psi_5(h_5, \boldsymbol{x}; \theta_5)$$

pairwise potentials

Unary potentials

Markov Random Fields (Factor Graphs)

$$p(H = \boldsymbol{h}, \boldsymbol{x}; \theta) = \frac{\Phi(\boldsymbol{h}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}', \boldsymbol{x}; \theta)}$$



$$\Phi(\boldsymbol{h}; \theta) = \phi_{12}(h_1, h_2; \theta_{12}) \times$$
$$\phi_{16}(h_1, h_6; \theta_{16}) \times$$
$$\phi_{26}(h_2, h_6; \theta_{26}) \times$$
$$\phi_{25}(h_2, h_5; \theta_{25}) \times$$
$$\phi_{45}(h_4, h_5; \theta_{45}) \times$$
$$\phi_{34}(h_3, h_4; \theta_{34}) \times$$

pairwise potentials

$$\psi_1(h_1; \theta_1) \times \psi_5(h_5; \theta_5)$$
$$\times \phi_{345}(h_3, h_4, h_5; \theta_{345})$$

Unary potentials

Markov Random Fields – Clique Factorization

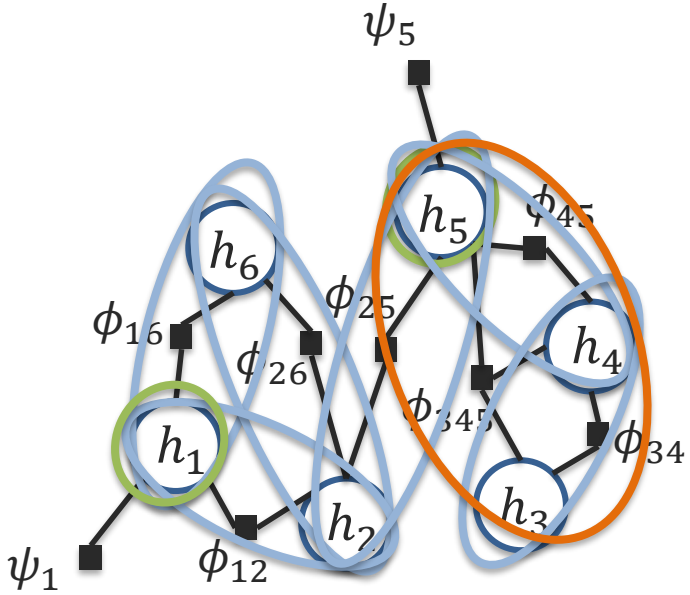
$$p(H = \boldsymbol{h}, \boldsymbol{x}; \theta) = \frac{\Phi(\boldsymbol{h}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}', \boldsymbol{x}; \theta)}$$

Clique factorization

$$\Phi(\boldsymbol{h}; \theta) = \phi_{12}(h_1, h_2; \theta_{12}) \times$$
$$\phi_{16}(h_1, h_6; \theta_{16}) \times$$
$$\phi_{26}(h_2, h_6; \theta_{26}) \times$$
$$\phi_{25}(h_2, h_5; \theta_{25}) \times$$
$$\phi_{45}(h_4, h_5; \theta_{45}) \times$$
$$\phi_{34}(h_3, h_4; \theta_{34}) \times$$
$$\psi_1(h_1; \theta_1) \times \psi_5(h_5; \theta_5)$$
$$\times \phi_{345}(h_3, h_4, h_5; \theta_{345})$$

pairwise potentials

Unary potentials



Chain Markov Random Fields (Factor Graphs)

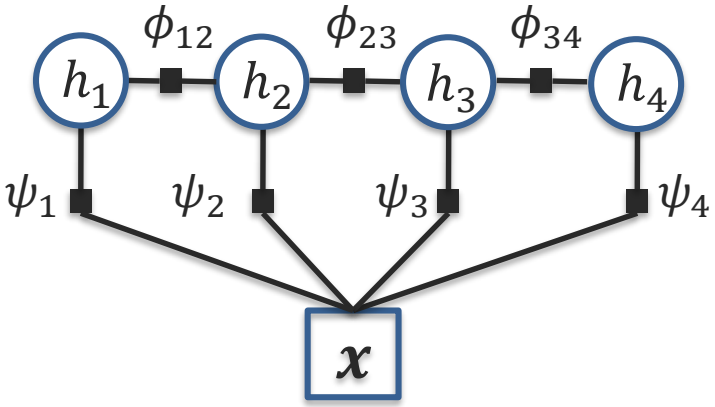
$$p(H = \boldsymbol{h}, \boldsymbol{x}; \theta) = \frac{\Phi(\boldsymbol{h}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}', \boldsymbol{x}; \theta)}$$

$$\Phi(\boldsymbol{h}; \theta) = \phi_{12}(h_1, h_2; \theta_{12}) \times$$
$$\phi_{23}(h_2, h_3; \theta_{23}) \times$$
$$\phi_{34}(h_3, h_4; \theta_{34}) \times$$

pairwise potentials

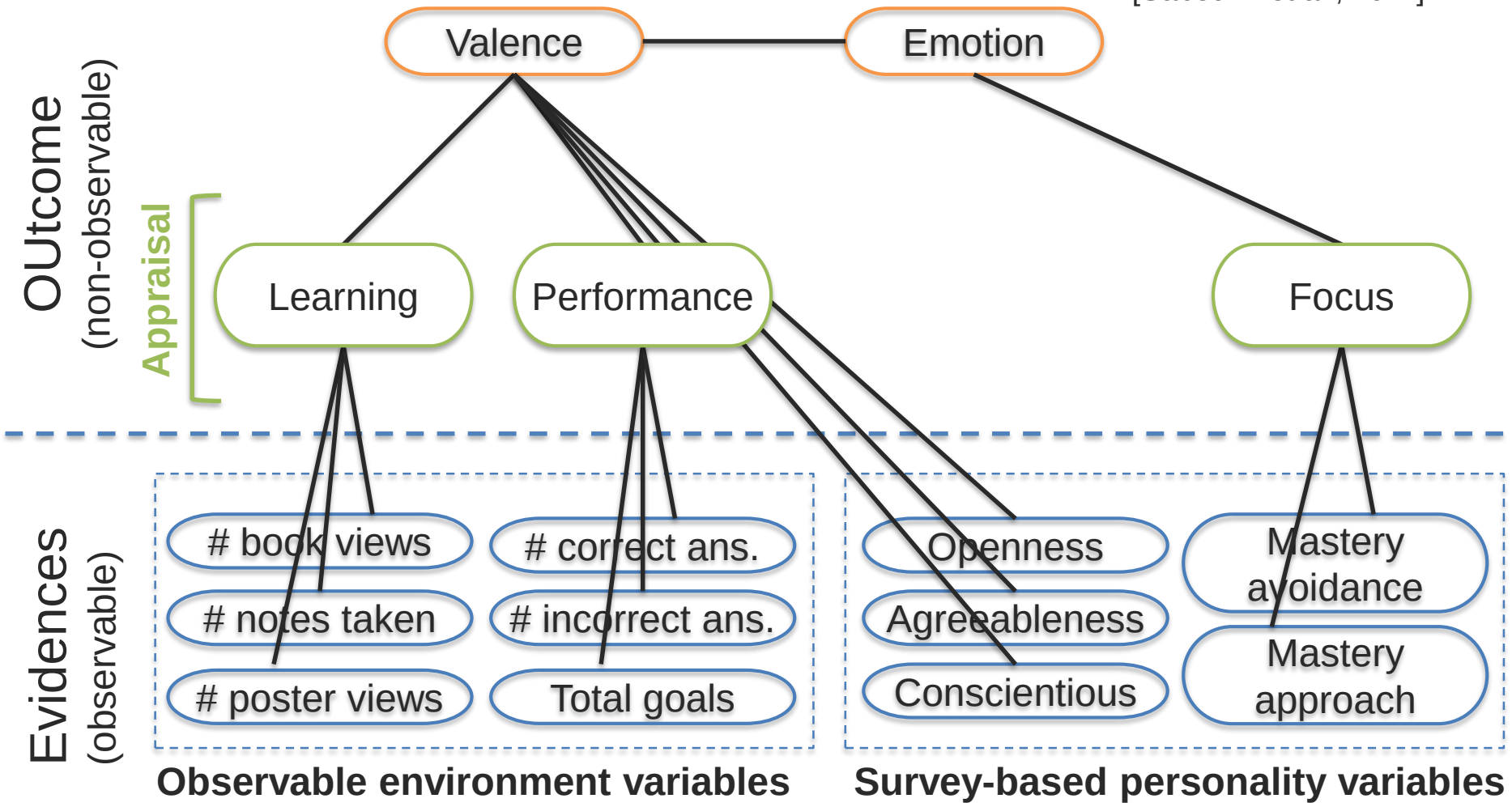
$$\psi_1(h_1; \theta_1) \times$$
$$\psi_2(h_2; \theta_2) \times$$
$$\psi_3(h_3; \theta_3) \times$$
$$\psi_4(h_4; \theta_4)$$

Unary potentials



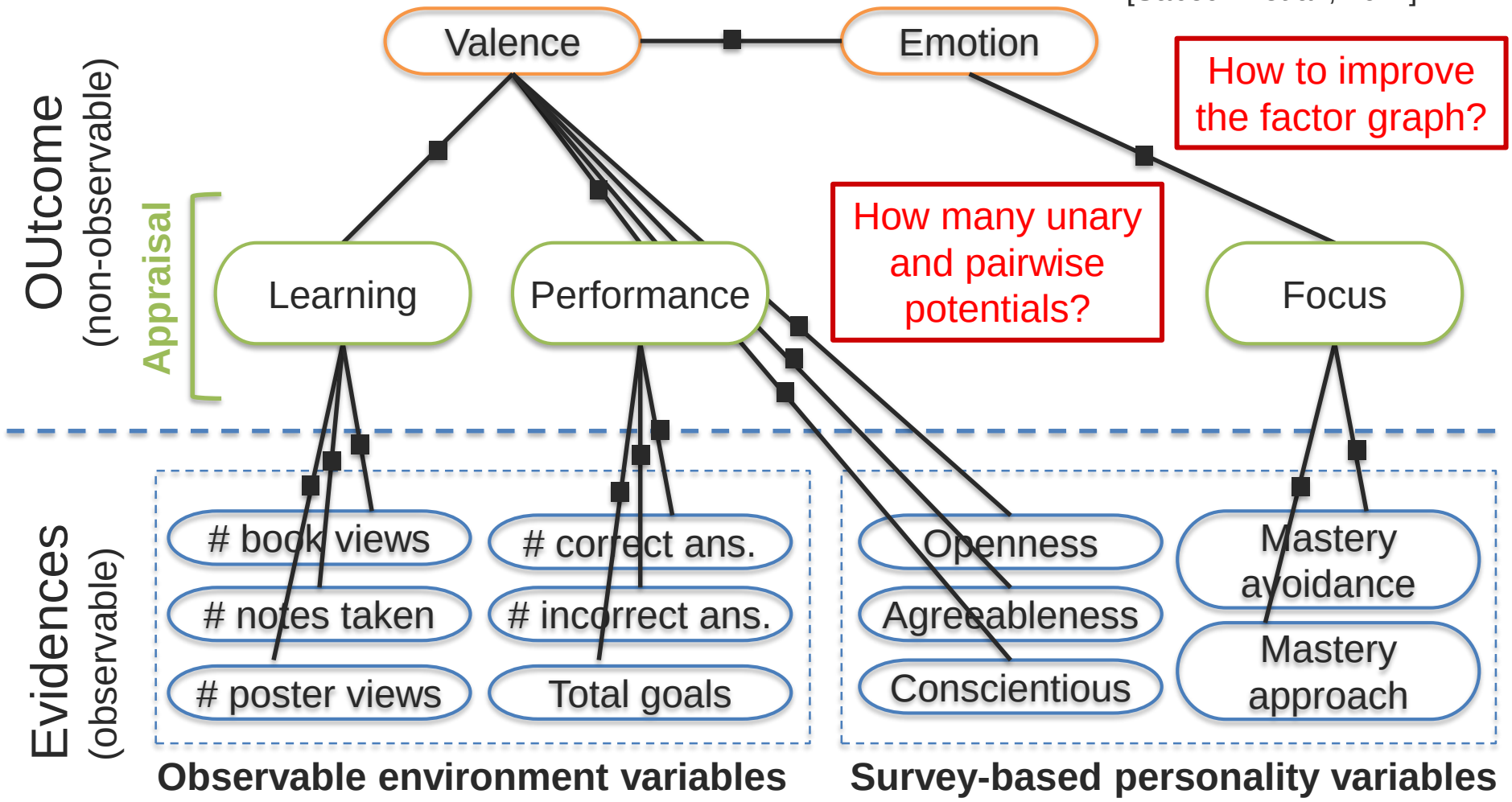
Example: Markov Random Field – Graphical Model

[Sabourin et al., 2011]



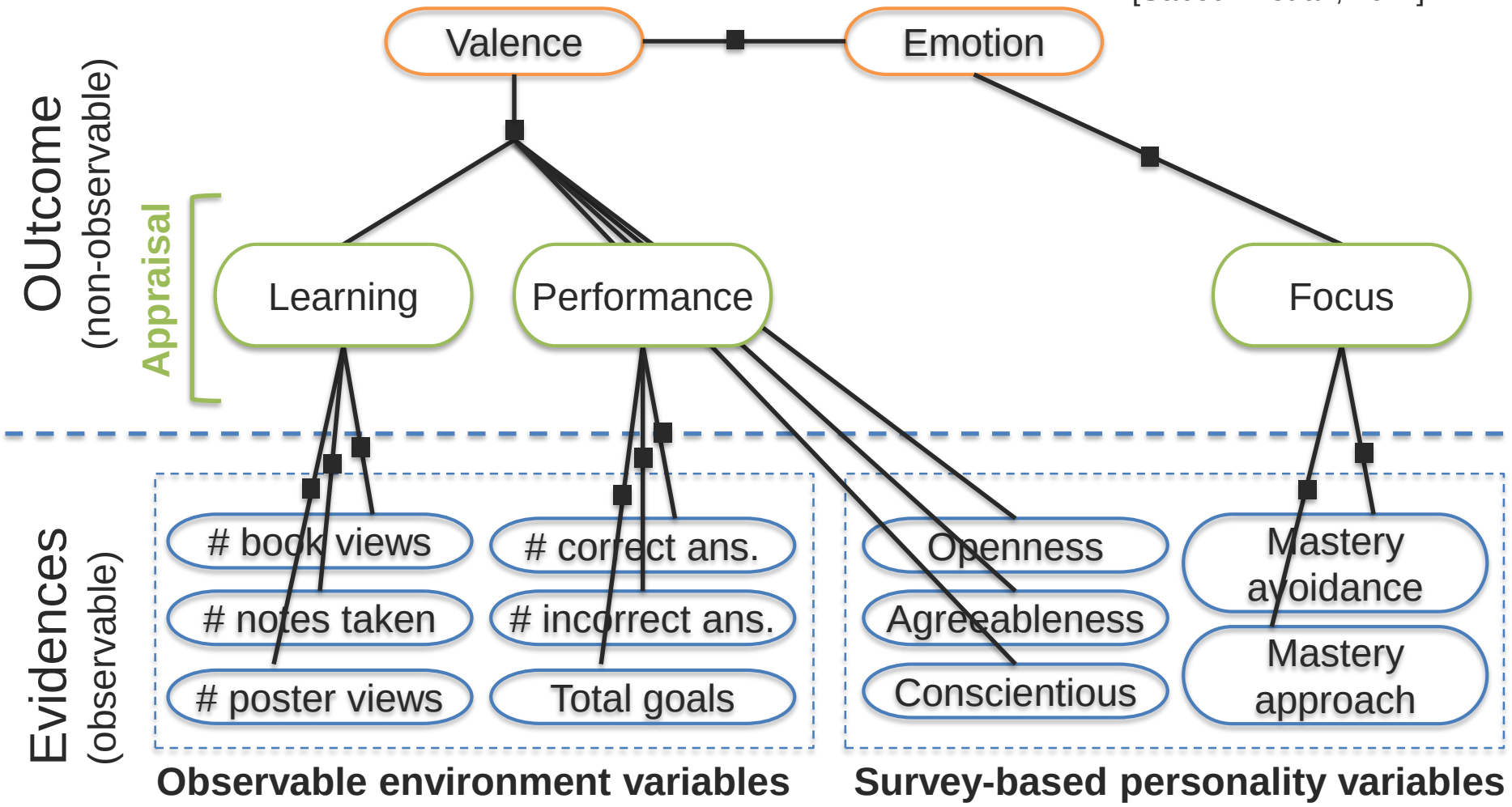
Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



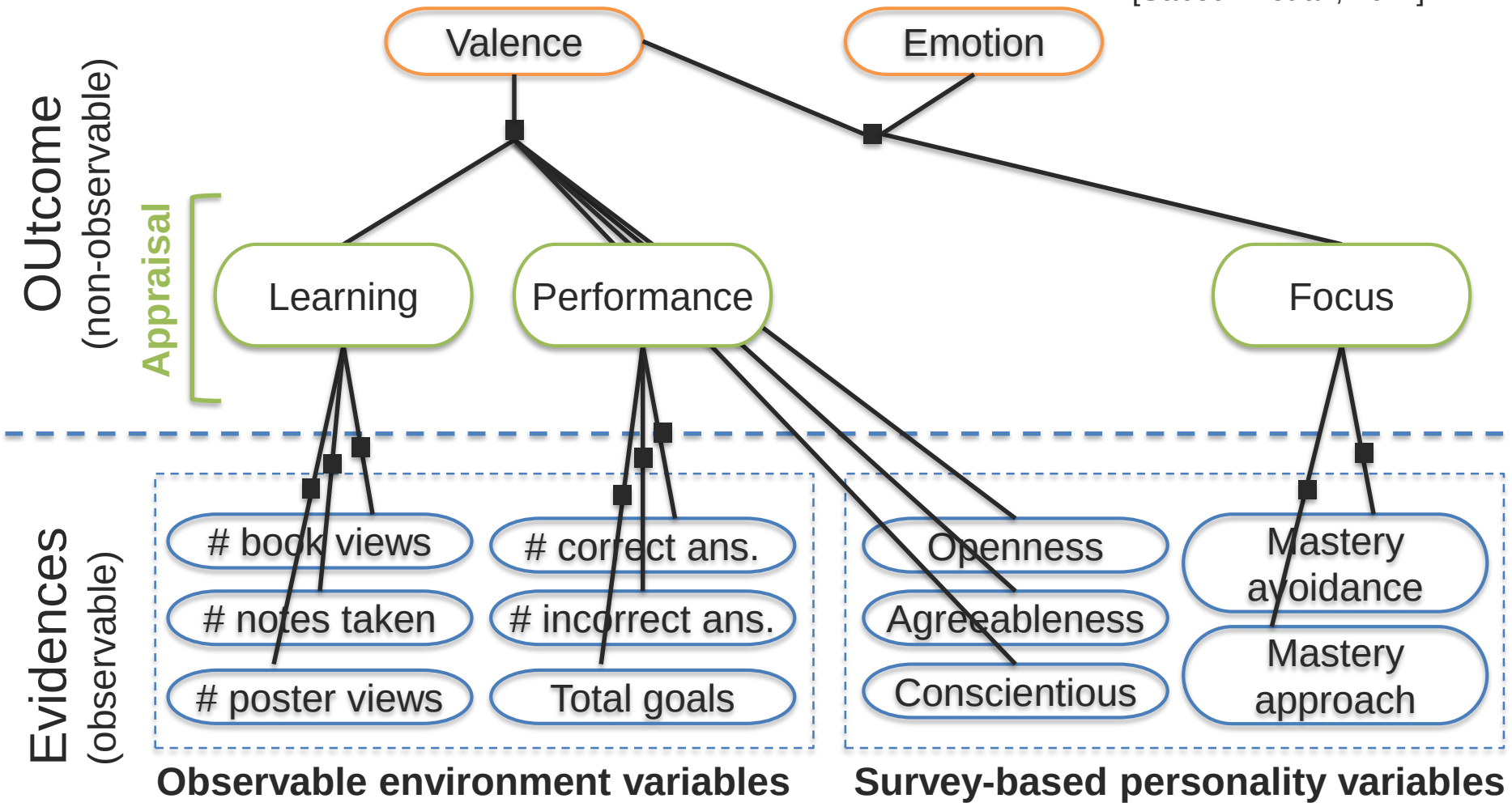
Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



Multimodal Machine Learning: Core Technical Challenges

Core Challenges in “Deep” Multimodal ML

Representation

Alignment

Fusion

Translation

Co-Learning

Multimodal Machine Learning: A Survey and Taxonomy

By Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency

https://arxiv.org/abs/1705.09406

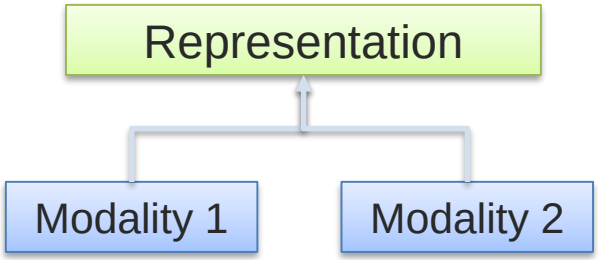
- ☑ **5 core challenges**
- ☑ **37 taxonomic classes**
- ☑ **253 referenced citations**

These challenges are non-exclusive.

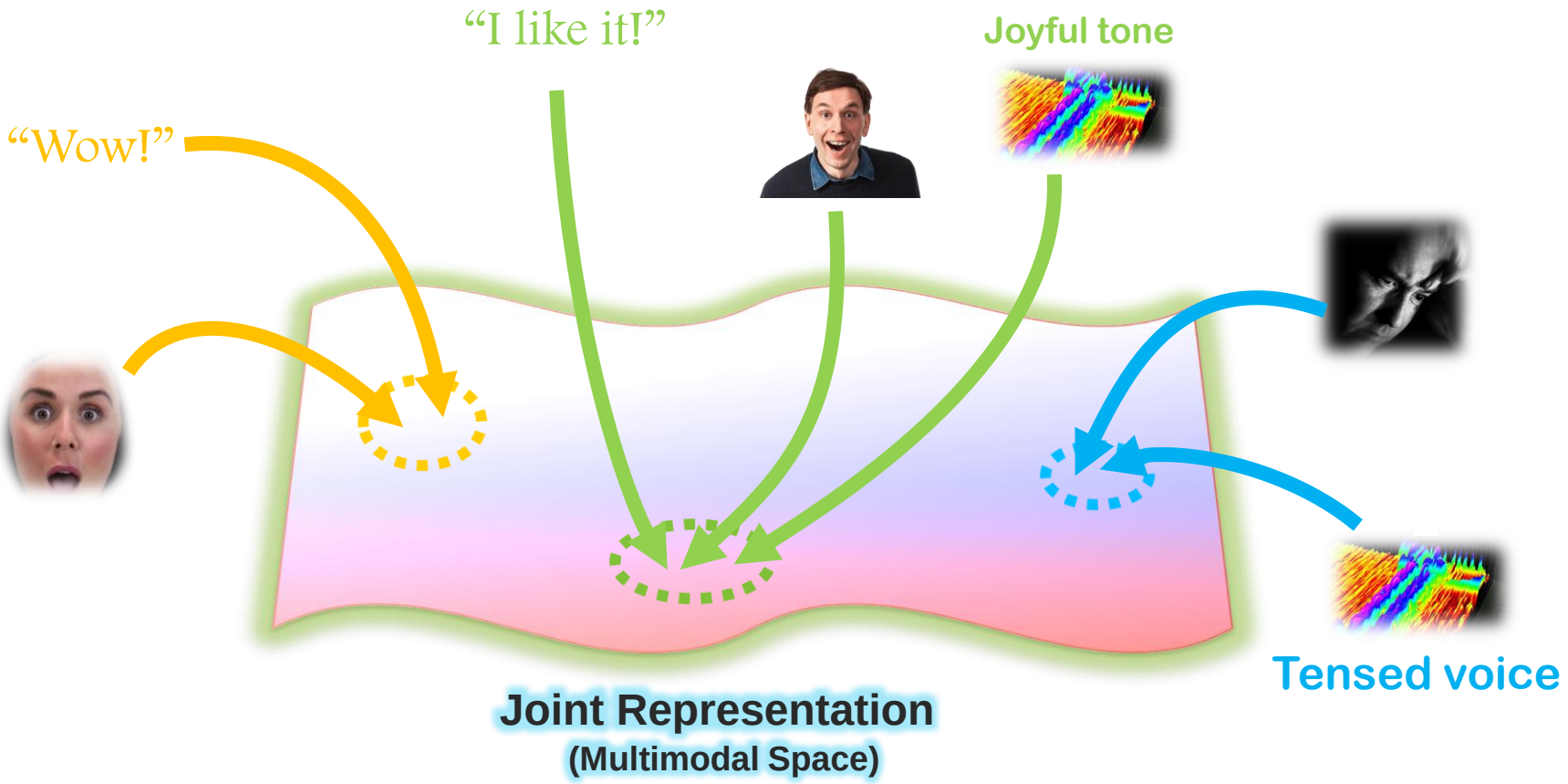
Core Challenge 1: Representation

Definition: Learning how to represent and summarize multimodal data in away that exploits the complementarity and redundancy.

Ⓐ Joint representations:



Joint Multimodal Representation



Joint Multimodal Representations

Audio-visual speech recognition

[Ngiam et al., ICML 2011]

- Bimodal Deep Belief Network

Image captioning

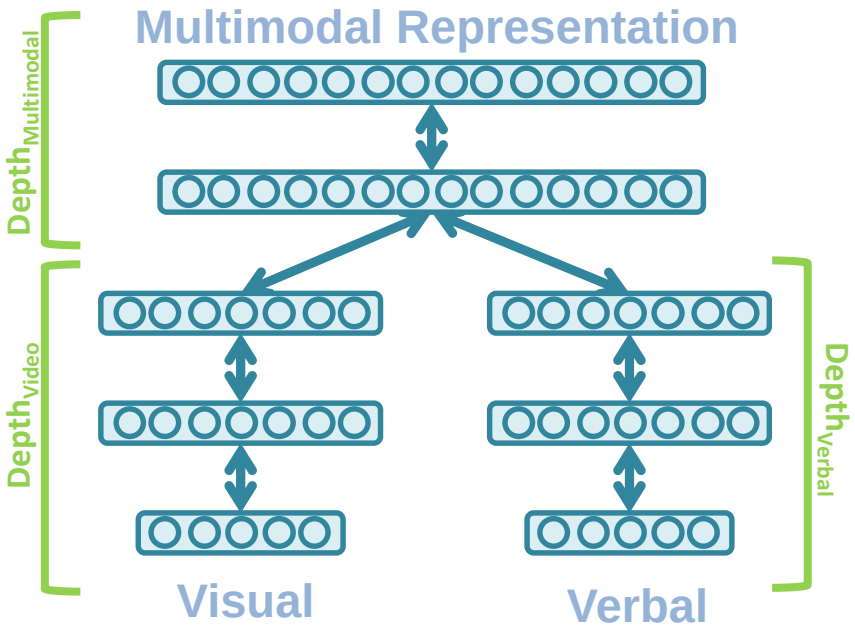
[Srivastava and Salahutdinov, NIPS 2012]

- Multimodal Deep Boltzmann Machine

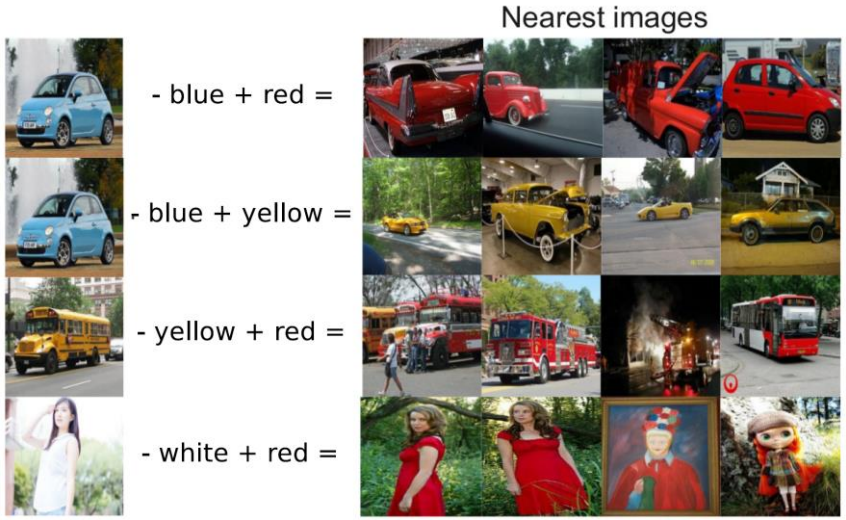
Audio-visual emotion recognition

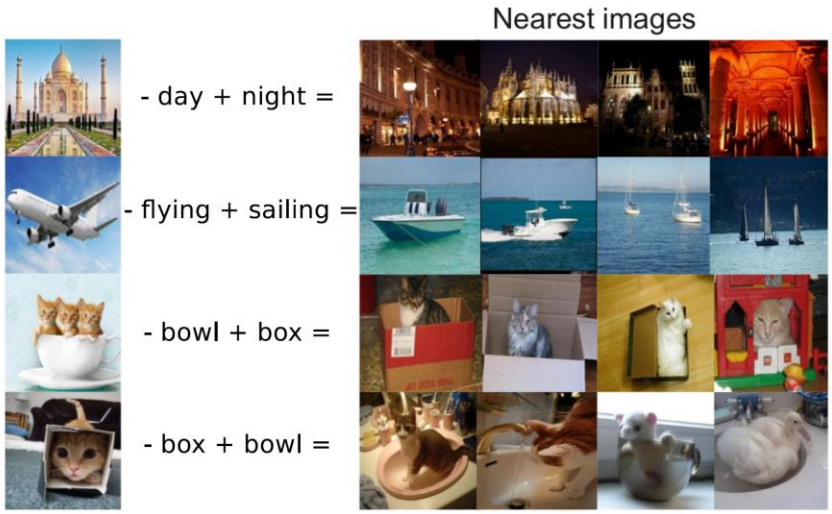
[Kim et al., ICASSP 2013]

- Deep Boltzmann Machine



Multimodal Vector Space Arithmetic



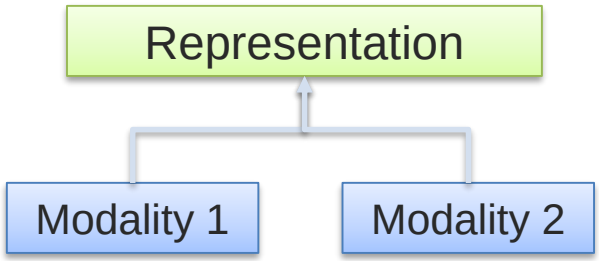


[Kiros et al., Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models, 2014]

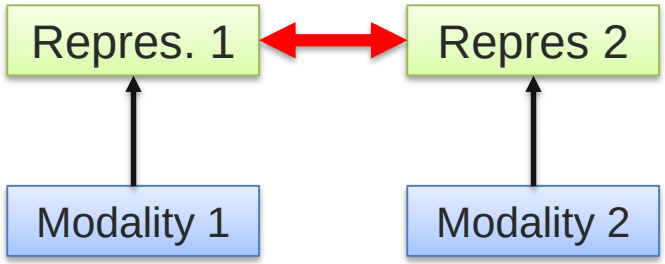
Core Challenge 1: Representation

Definition: Learning how to represent and summarize multimodal data in away that exploits the complementarity and redundancy.

A Joint representations:



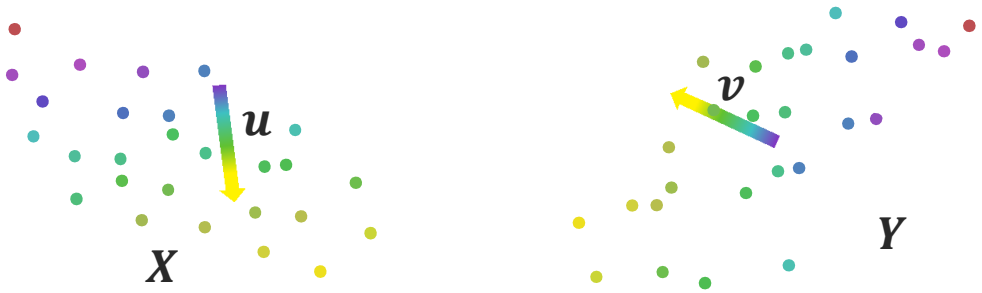
B Coordinated representations:

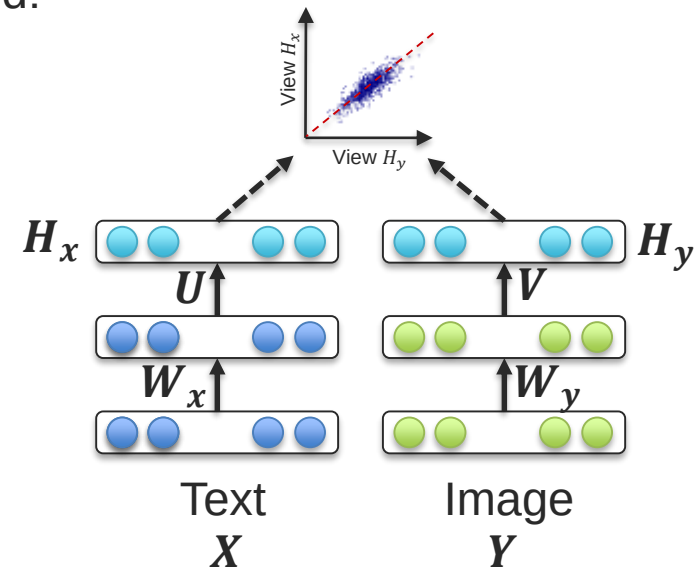


Coordinated Representation: Deep CCA

Learn linear projections that are maximally correlated:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmax}}\, corr\left(\boldsymbol{u}^T\boldsymbol{X}, \boldsymbol{v}^T\boldsymbol{Y}\right)$$

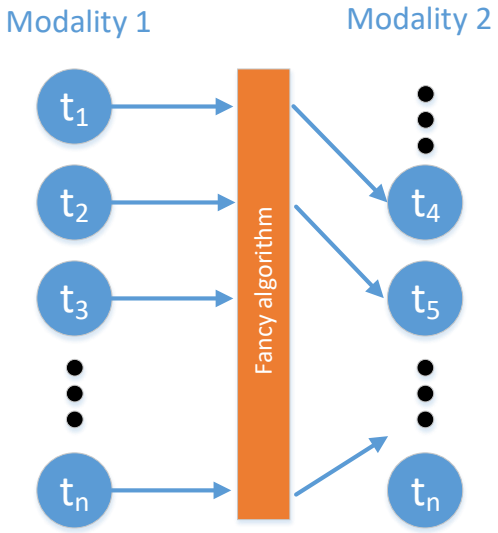




Andrew et al., ICML 2013

Core Challenge 2: Alignment

Definition: Identify the direct relations between (sub)elements from two or more different modalities.



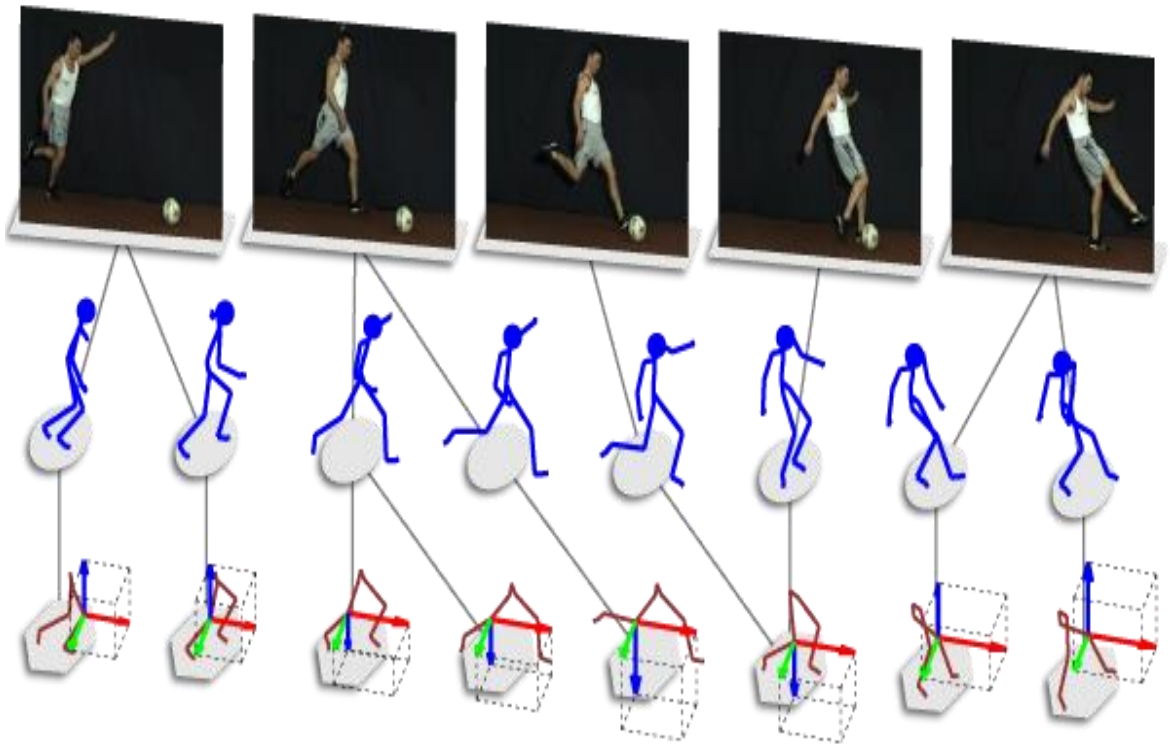
A Explicit Alignment

The goal is to directly find correspondences between elements of different modalities

B Implicit Alignment

Uses internally latent alignment of modalities in order to better solve a different problem

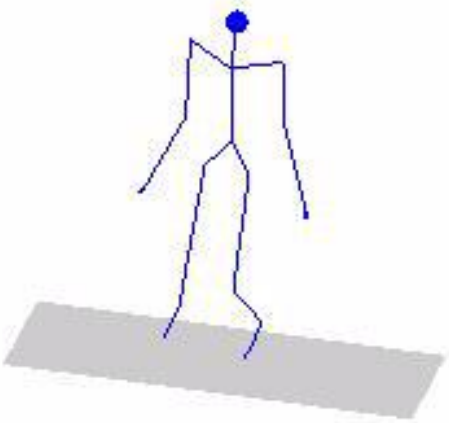
Temporal sequence alignment



Applications:
- Re-aligning asynchronous data
- Finding similar data across modalities (we can estimate the aligned cost)
- Event reconstruction from multiple sources

Alignment examples (multimodal)

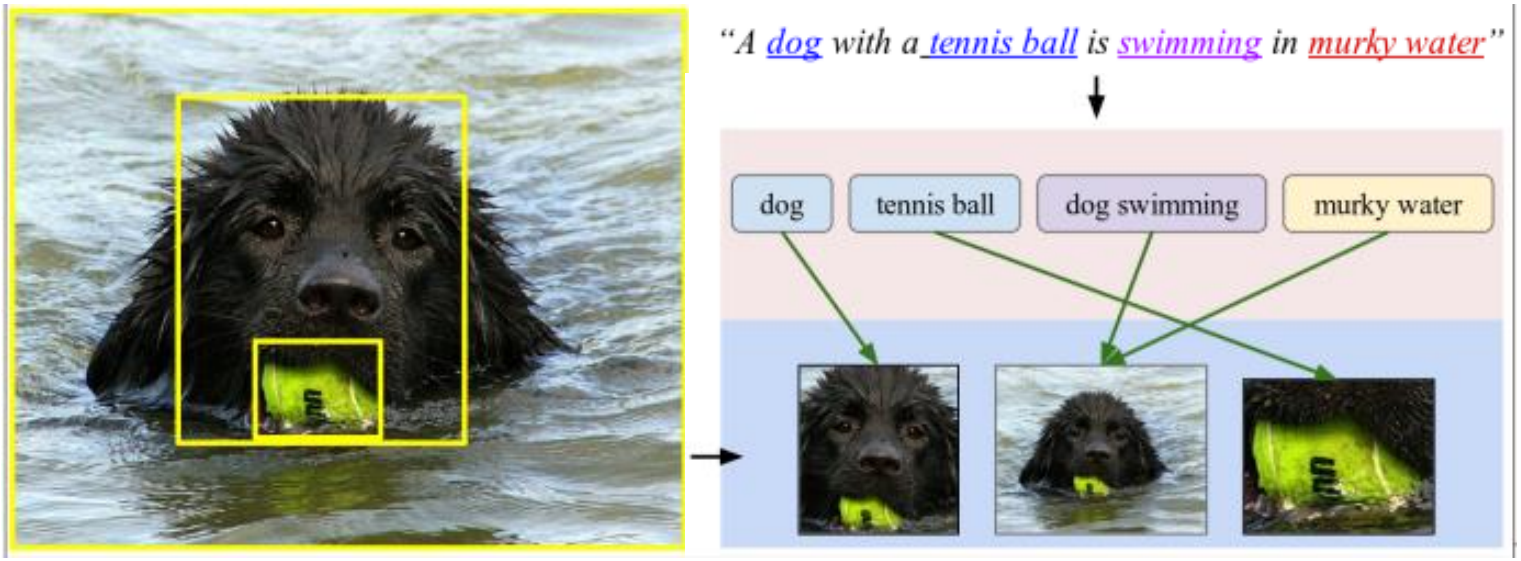
1/273



1/51

1/127

Implicit Alignment



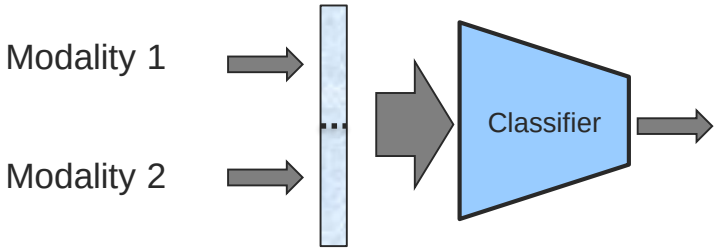
Karpathy et al., Deep Fragment Embeddings for Bidirectional Image Sentence Mapping, https://arxiv.org/pdf/1406.5679.pdf

Core Challenge 3: Fusion

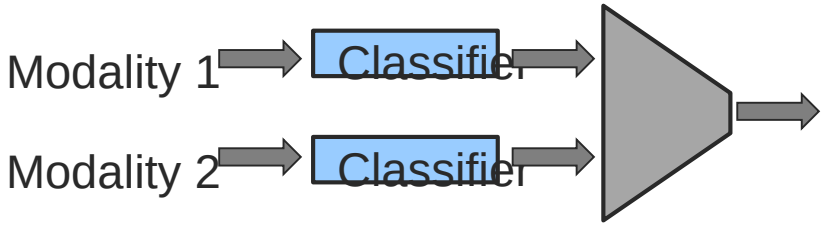
Definition: To join information from two or more modalities to perform a prediction task.

A Model-Agnostic Approaches

1) Early Fusion



2) Late Fusion



Core Challenge 3: Fusion

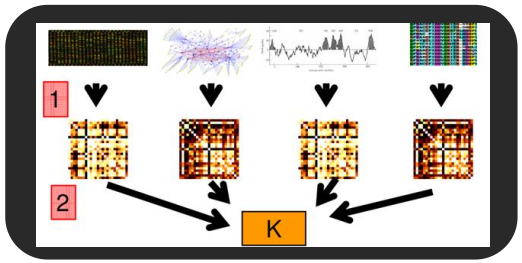
Definition: To join information from two or more modalities to perform a prediction task.

B **Model-Based (Intermediate) Approaches**

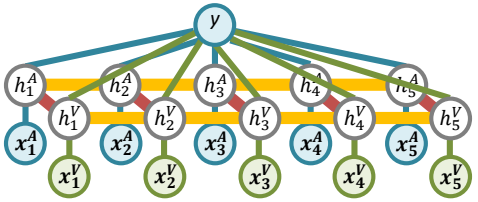
1) **Deep neural networks**

2) **Kernel-based methods**

3) **Graphical models**



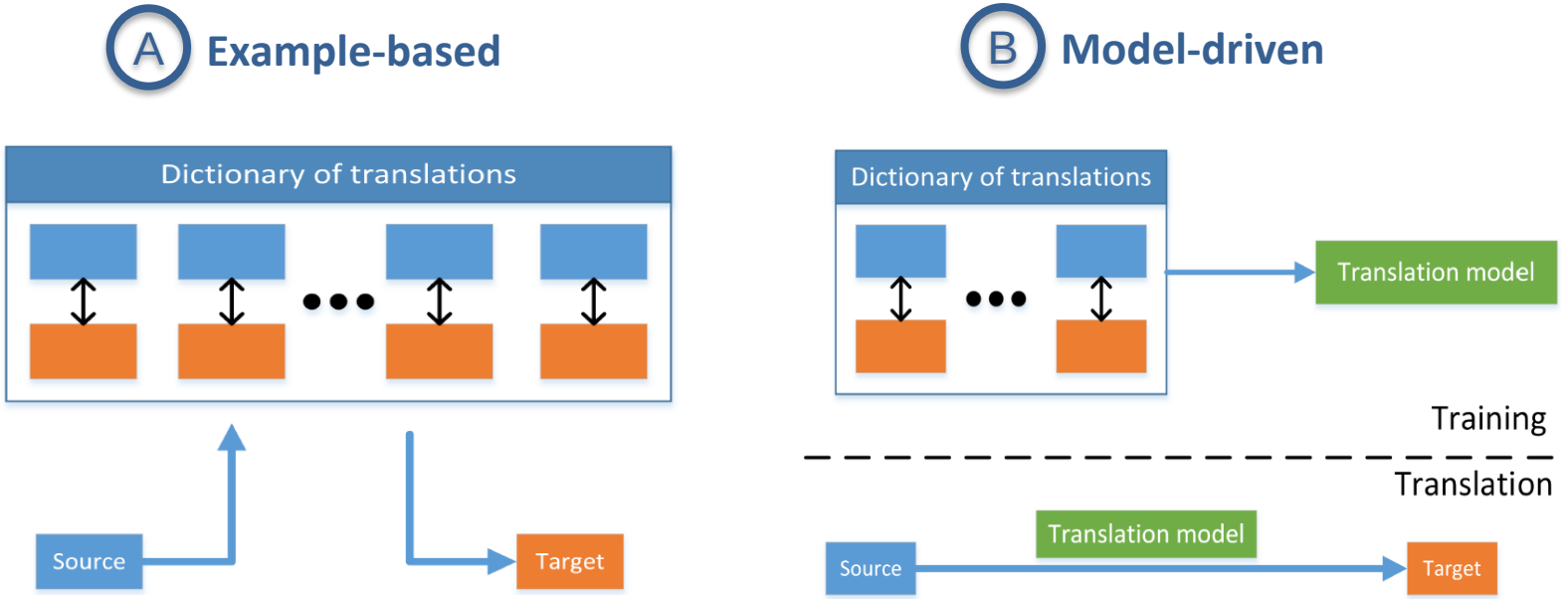
Multiple kernel learning



Multi-View Hidden CRF

Core Challenge 4: Translation

Definition: Process of changing data from one modality to another, where the translation relationship can often be open-ended or subjective.



Core Challenge 4 – Translation

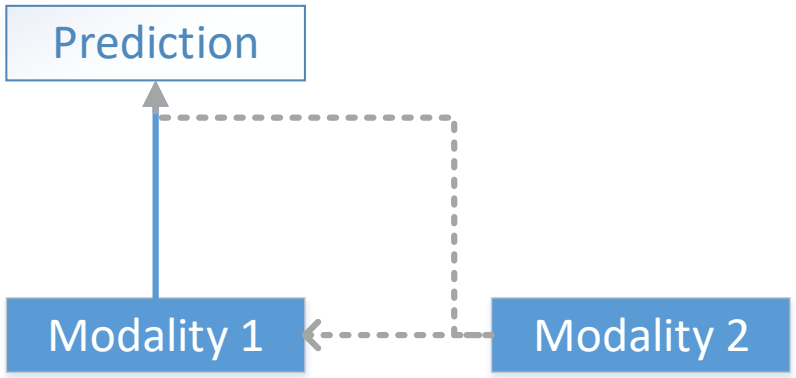
Visual gestures
(both speaker and listener gestures)

Transcriptions
+
Audio streams

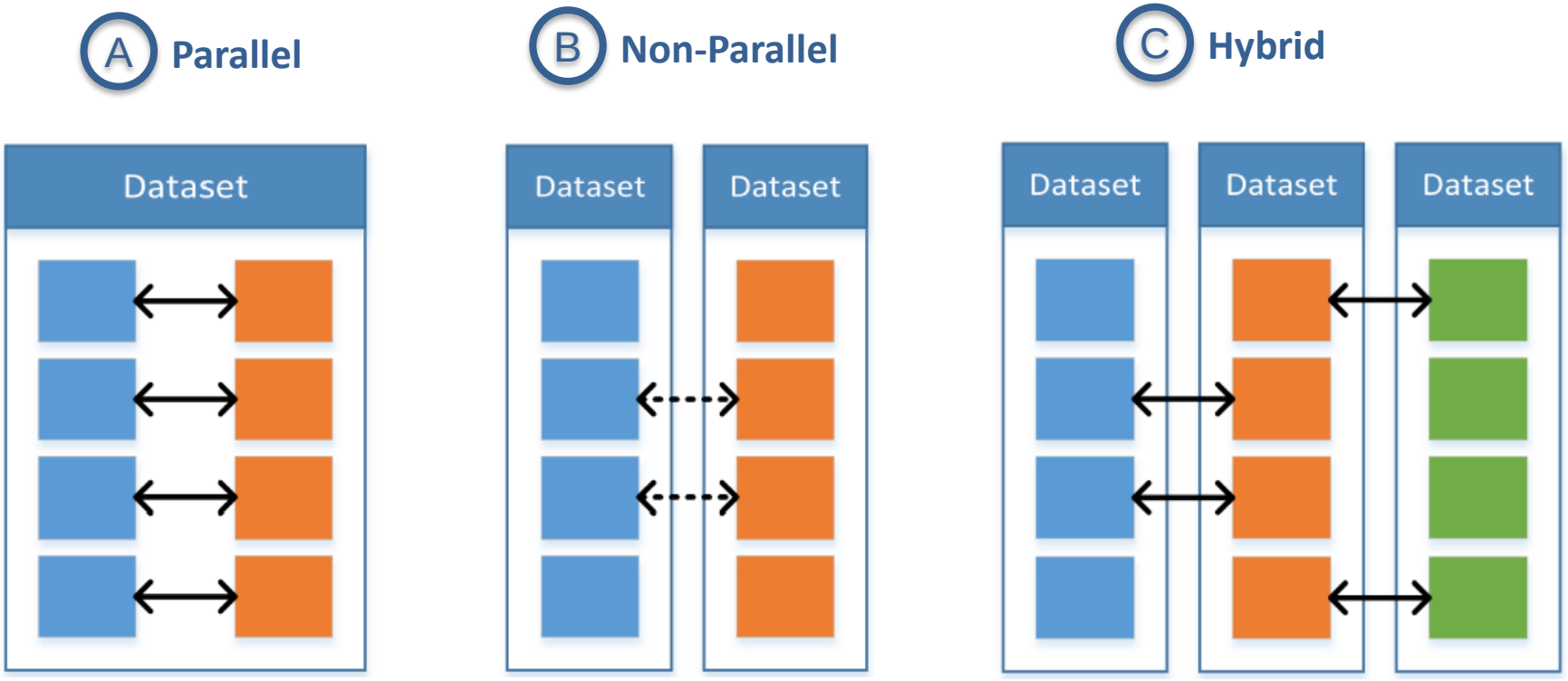
Marsella et al., Virtual character performance from speech, SIGGRAPH/Eurographics Symposium on Computer Animation, 2013

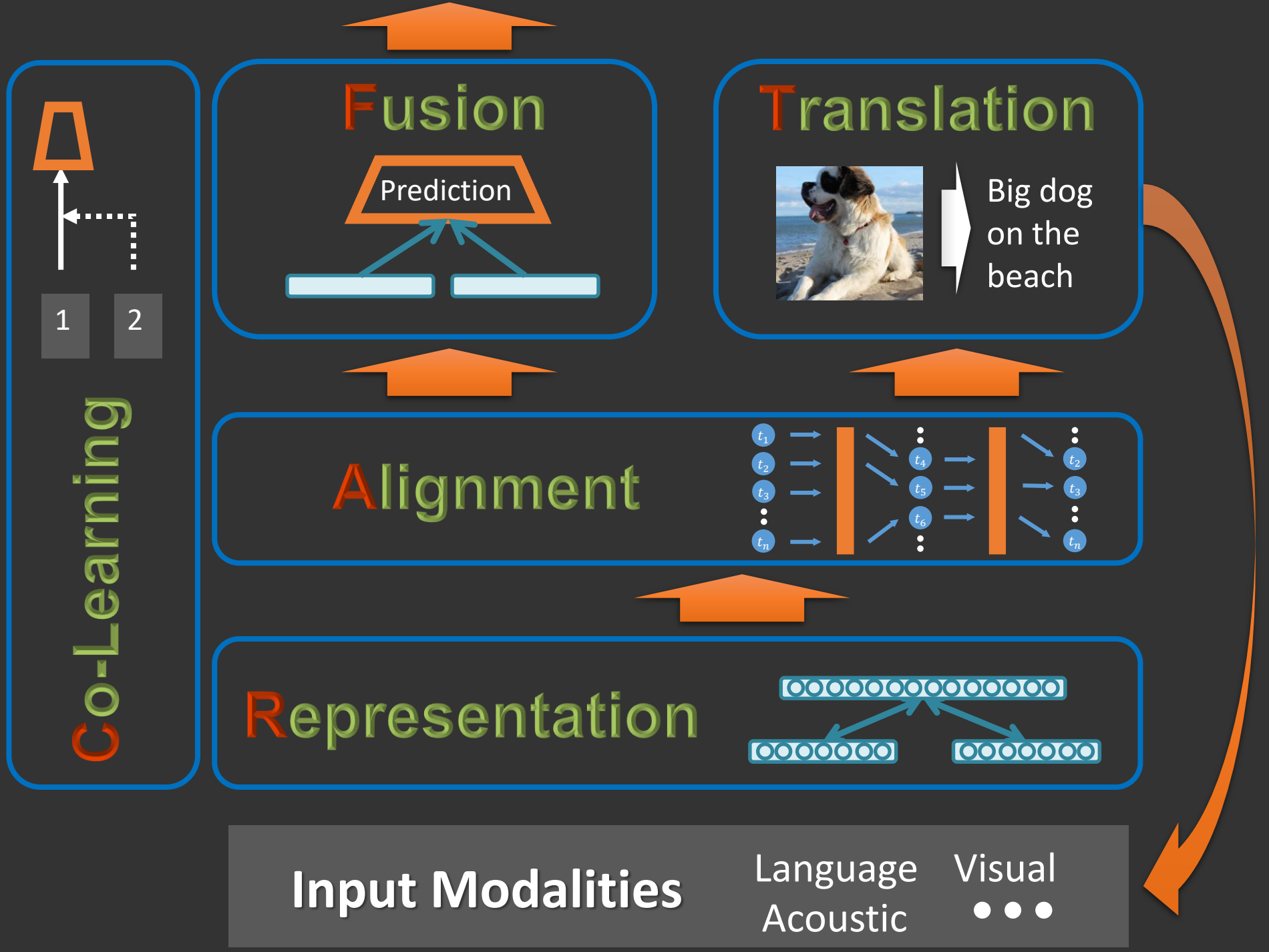
Core Challenge 5: Co-Learning

Definition: Transfer knowledge between modalities, including their representations and predictive models.



Core Challenge 5: Co-Learning




Co-Learning
1
2
Fusion
Prediction
Translation
Big dog on the beach
Alignment
Representation
Input Modalities
Language
Acoustic
Visual

Taxonomy of Multimodal Research [https://arxiv.org/abs/1705.09406]

Representation

- Joint
  - *Neural networks*
  - *Graphical models*
  - *Sequential*
- Coordinated
  - *Similarity*
  - *Structured*

Translation

- Example-based
  - *Retrieval*
  - *Combination*
- Model-based
  - *Grammar-based*
  - *Encoder-decoder*
  - *Online prediction*

Alignment

- Explicit
  - *Unsupervised*
  - *Supervised*
- Implicit
  - *Graphical models*
  - *Neural networks*

Fusion

- Model agnostic
  - *Early fusion*
  - *Late fusion*
  - *Hybrid fusion*
- Model-based
  - *Kernel-based*
  - *Graphical models*
  - *Neural networks*

Co-learning

- Parallel data
  - *Co-training*
  - *Transfer learning*
- Non-parallel data
  - *Zero-shot learning*
  - *Concept grounding*
  - *Transfer learning*
- *Hybrid data*
  - *Bridging*

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency, Multimodal Machine Learning: A Survey and Taxonomy

Real world tasks tackled by MMML

- Affect recognition
  - Emotion
  - Persuasion
  - Personality traits
- Media description
  - Image captioning
  - Video captioning
  - Visual Question Answering
- Event recognition
  - Action recognition
  - Segmentation
- Multimedia information retrieval
  - Content based/Cross-media

Multimodal Applications

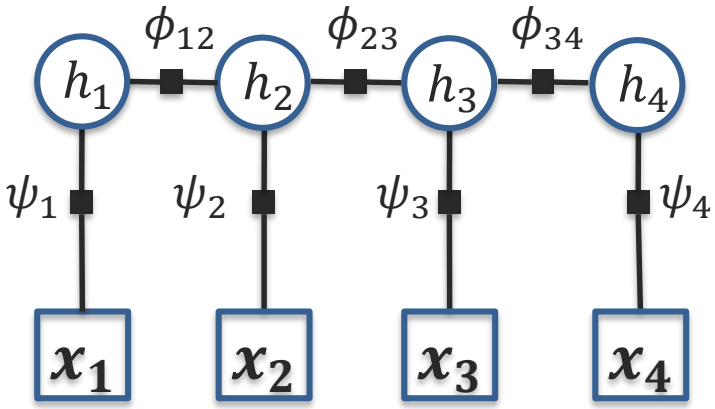
[https://arxiv.org/abs/1705.09406]

	Challenges				
Applications	Representation	Translation	Fusion	Alignment	Co-learning
Speech Recognition and Synthesis					
Audio-visual Speech Recognition	✓		✓	✓	✓
(Visual) Speech Synthesis	✓	✓			
Event Detection					
Action Classification	✓		✓		✓
Multimedia Event Detection	✓		✓		✓
Emotion and Affect					
Recognition	✓		✓	✓	✓
Synthesis	✓	✓			
Media Description					
Image Description	✓	✓		✓	✓
Video Description	✓	✓	✓	✓	✓
Visual Question-Answering	✓		✓	✓	✓
Media Summarization	✓	✓	✓		
Multimedia Retrieval					
Cross Modal retrieval	✓	✓		✓	✓
Cross Modal hashing	✓				✓

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency, Multimodal Machine Learning: A Survey and Taxonomy

Discriminative Graphical Models

Generative versus Discriminative



Generative or Discriminative?

Answer: *It depends on the loss function!*

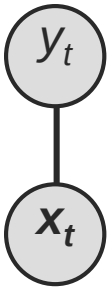
Generative loss function:
(joint probability)

$$L(\theta) = \sum_{j=1}^{N} P(\boldsymbol{h}^{(j)}, \boldsymbol{X}^{(j)}; \theta)$$

Discriminative loss function:
(conditional probability)

$$L(\theta) = \sum_{j=1}^{N} \log P\left(\boldsymbol{h}^{(j)} \middle| \boldsymbol{X}^{(j)}; \theta\right)$$

Discriminative Model: Logistic classifier



Label : {0:`Dominant, 1:Not-dominant`}

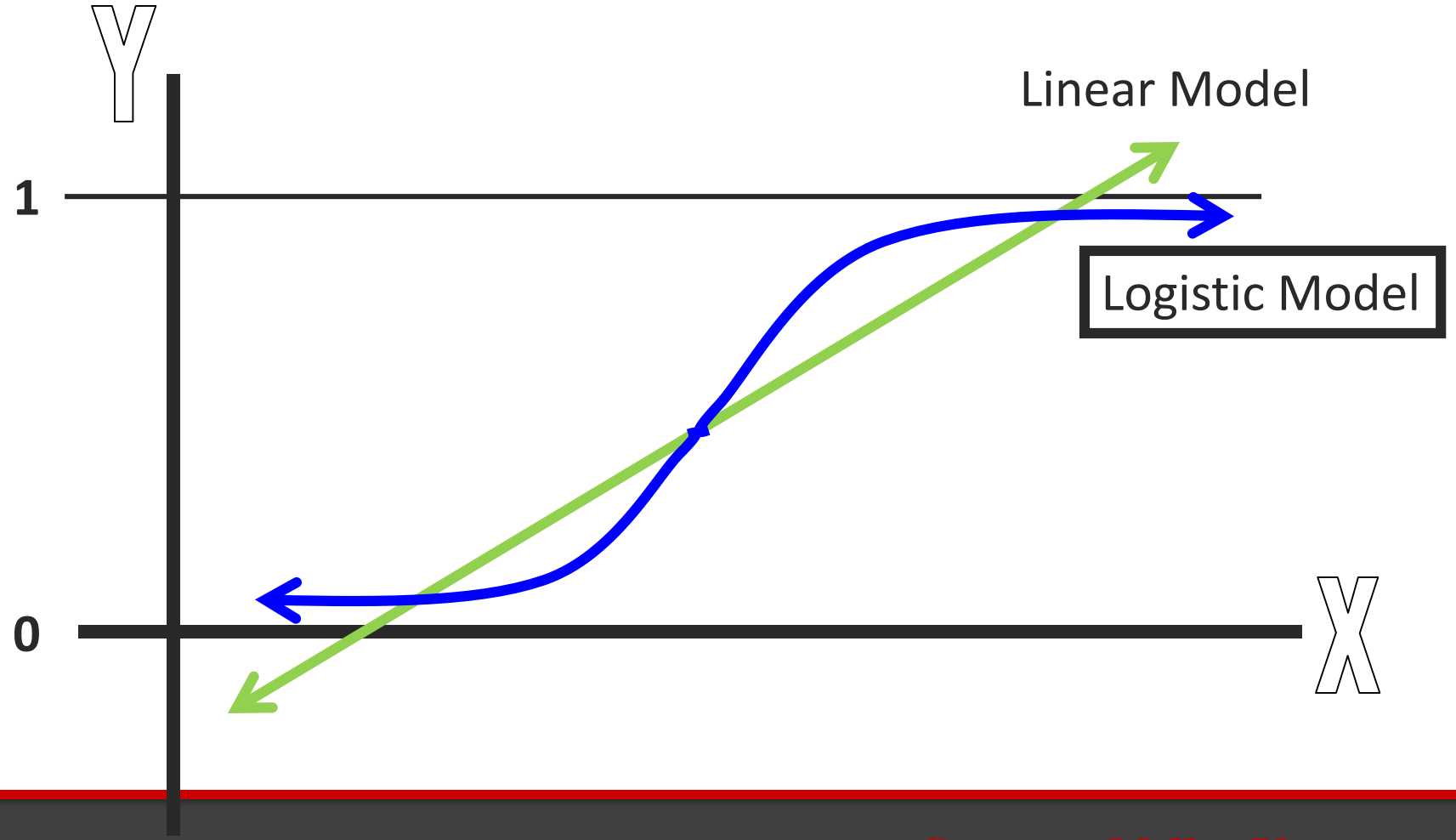
Observation vector: [speech-energy, gaze, turn-taking]

Score function:

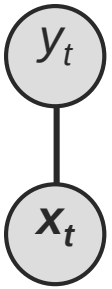
$$P(y_t = 1|\boldsymbol{x}_t) = \frac{1}{1 + \exp(-\boldsymbol{\theta}\boldsymbol{x}_t)}$$ Binary form

$$P(y_t = c|\boldsymbol{x}_t) = \frac{\exp(\boldsymbol{\theta}_c\boldsymbol{x}_t)}{\sum_{k=1}^{K}\exp(\boldsymbol{\theta}_k\boldsymbol{x}_t)}$$ Multinomial form

Comparing Linear and Logistic Models



Discriminative Model: Logistic classifier



Label : {0:`Dominant, 1:Not-dominant`}

Observation vector: [speech-energy, gaze, turn-taking]

Score function:

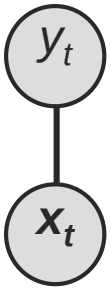
$$P(y_t = c|\boldsymbol{x}_t) = \frac{\exp(\boldsymbol{\theta}_c \boldsymbol{x}_t)}{\sum_{k=1}^{K} \boldsymbol{\theta}_c \boldsymbol{x}_t}$$

Familiar multinomial form

$$P(y_t|\boldsymbol{x}_t) = \frac{1}{\mathcal{Z}(\boldsymbol{x}_t)} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x}_t)\right)$$

General form

Discriminative Model: Logistic classifier



Label : {0:`Dominant, 1:Not-dominant`}

Observation vector: [speech-energy, gaze, turn-taking]

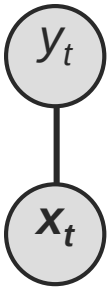
$$P(y_t|\boldsymbol{x}_t) = \frac{1}{\mathcal{Z}(\boldsymbol{x}_t)} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x}_t)\right) = \frac{1}{\mathcal{Z}(\boldsymbol{x}_t)} \exp\left(\sum_{k=1}^{K} \phi_k(y_t, \boldsymbol{x}_t; \theta_k)\right)$$

Partition function

Weights

Feature functions

Feature Functions



Label : {0:Dominant, 1:Not-dominant}

2

Observation vector: [speech-energy, gaze, turn-taking]

3

$$P(y_t|\boldsymbol{x_t}) = \frac{1}{\mathcal{Z}(\boldsymbol{x_t})} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x_t})\right)$$

K = 6

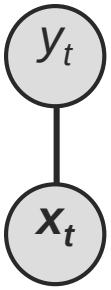
$$f_0(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t0}, & y_t = 0 \\ 0, & Otherwise \end{cases} \qquad f_1(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t0}, & y_t = 1 \\ 0, & Otherwise \end{cases}$$

$$f_2(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t1}, & y_t = 0 \\ 0, & Otherwise \end{cases} \qquad f_3(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t1}, & y_t = 1 \\ 0, & Otherwise \end{cases}$$

$$f_4(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t2}, & y_t = 0 \\ 0, & Otherwise \end{cases} \qquad f_5(y_t, \boldsymbol{x_t}) = \begin{cases} x_{t2}, & y_t = 1 \\ 0, & Otherwise \end{cases}$$

	y_t=0	y_t=1
x_{t0}	θ_0	θ_1
x_{t1}	θ_2	θ_3
x_{t2}	θ_4	θ_5

Partition Function: Normalizing Constant



Label : {0:Dominant, 1:Not-dominant}

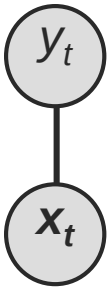
Observation vector: [speech-energy, gaze, turn-taking]

$$P(y_t|\boldsymbol{x}_t) = \frac{1}{\mathcal{Z}(\boldsymbol{x}_t)} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x}_t)\right)$$

$$\mathcal{Z}(\boldsymbol{x}_t) = \sum_{y'=0}^{|Y|} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y', \boldsymbol{x}_t)\right)$$

So that $P(y_t|\boldsymbol{x}_t)$ stays between 0 and 1.

Training and Loss Function



Label : {0:`Dominant, 1:Not-dominant`}

Observation vector: [speech-energy, gaze, turn-taking]

$$P(y_t|\boldsymbol{x}_t) = \frac{1}{\mathcal{Z}(\boldsymbol{x}_t)} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x}_t)\right)$$

Loss function: Conditional log likelihood

$$L(\theta) = \sum_{j=1}^{N} \log P\left(\boldsymbol{y}^{(j)} \middle| \boldsymbol{X}^{(j)}; \theta\right) \quad - R(\theta)$$

Regularization

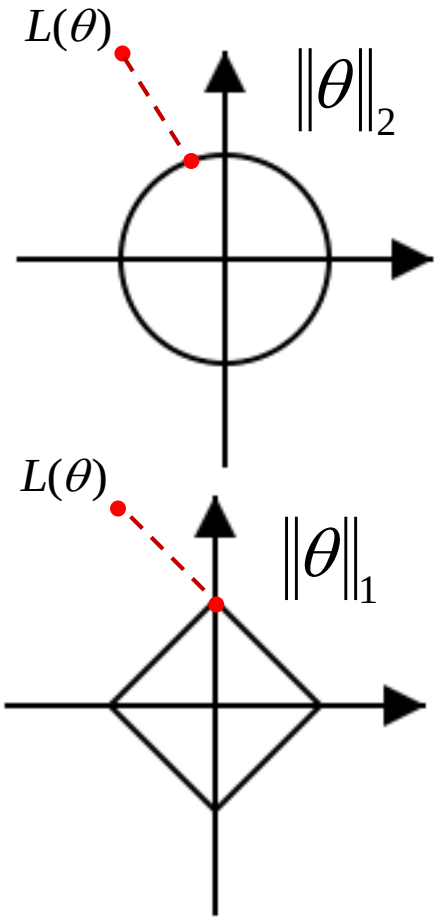
$$L(\theta) = \sum_{j=1}^{N} \log P\left(\boldsymbol{y}^{(j)} \middle| \boldsymbol{X}^{(j)}; \theta\right) - R(\theta)$$

- L-2 Norm (Gaussian prior):

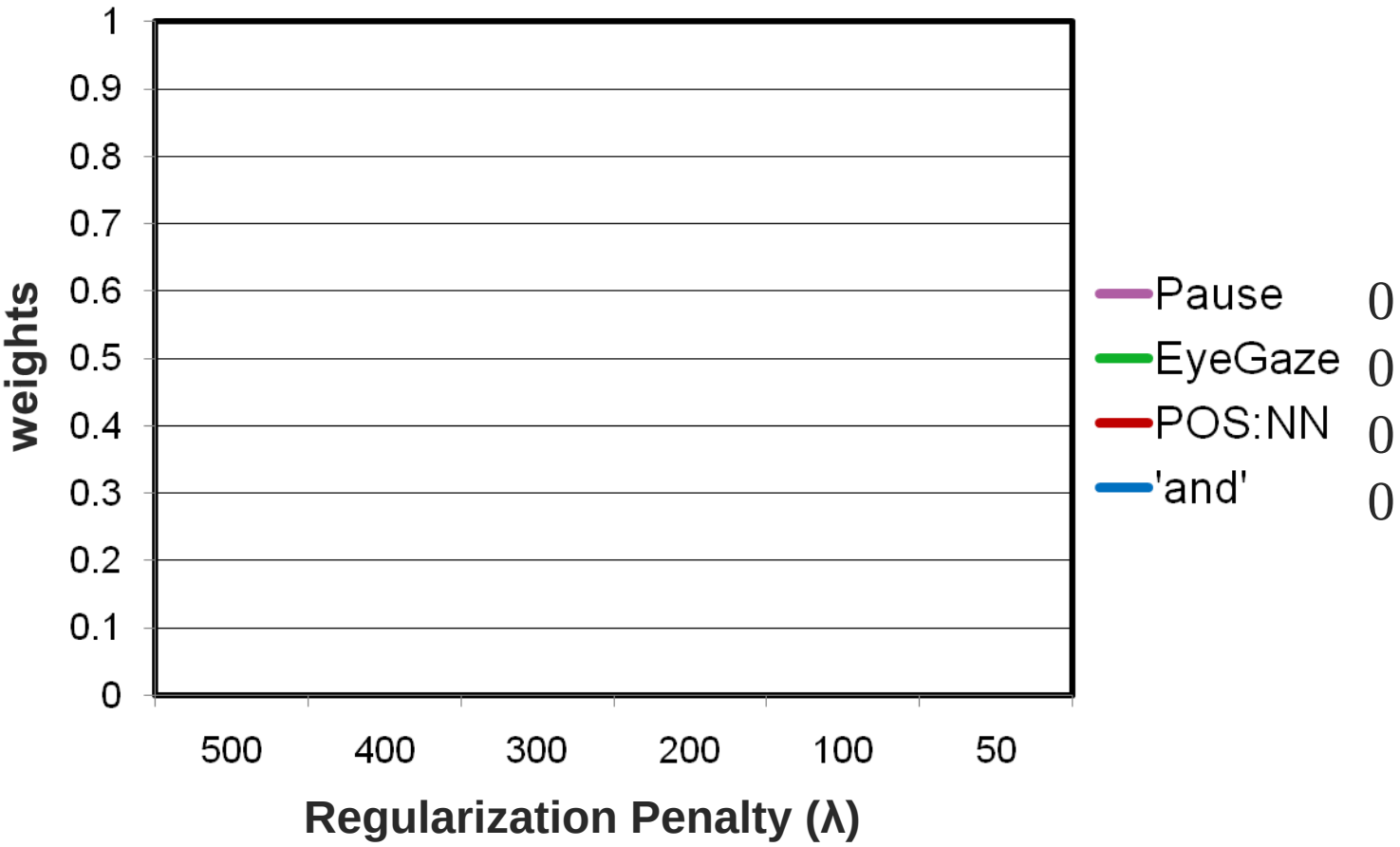
$$R(\theta) = \lambda \|\theta\|_2$$

- L-1 Norm (Laplacian prior):

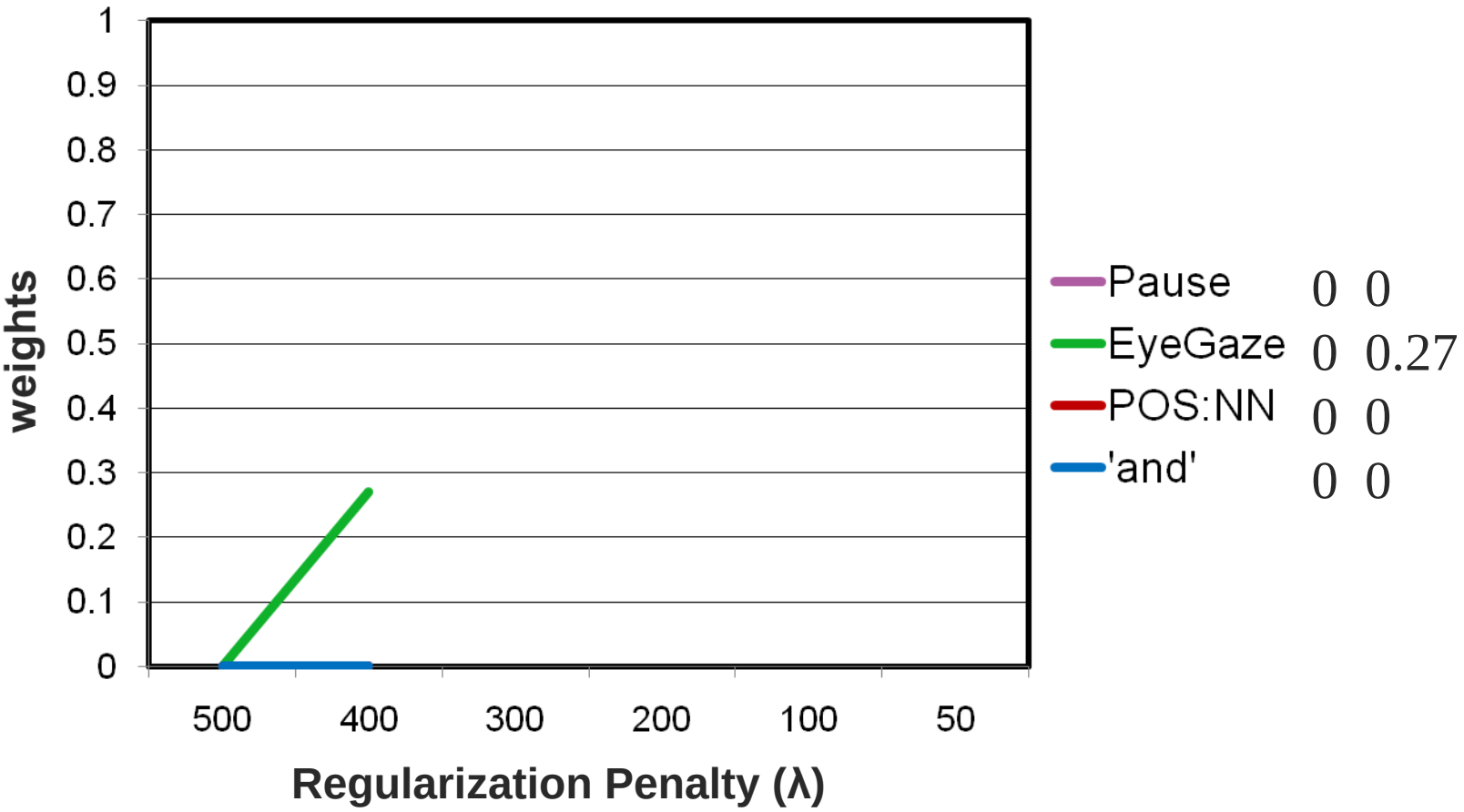
$$R(\theta) = \lambda \|\theta\|_1$$



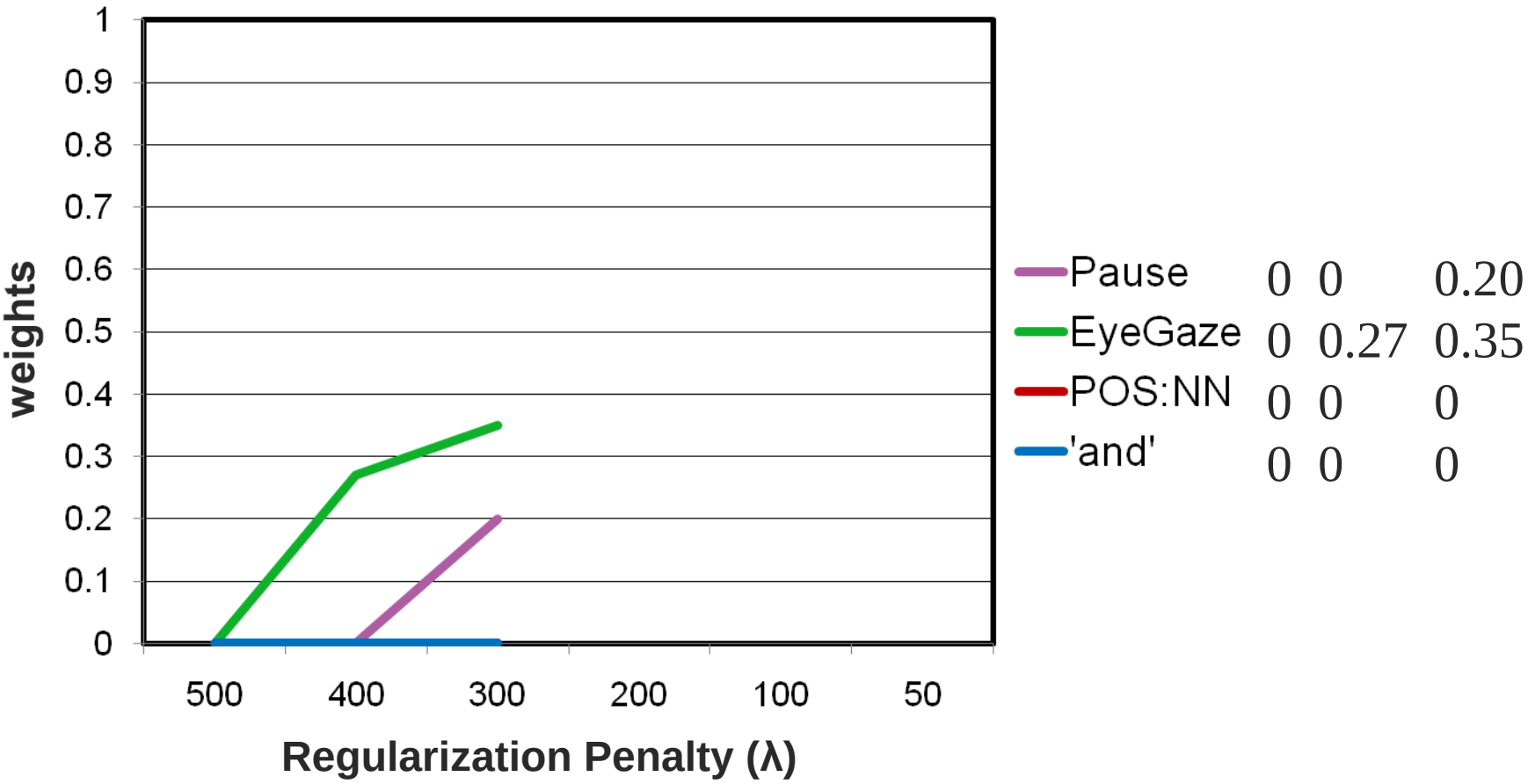
Sparse Feature Ranking



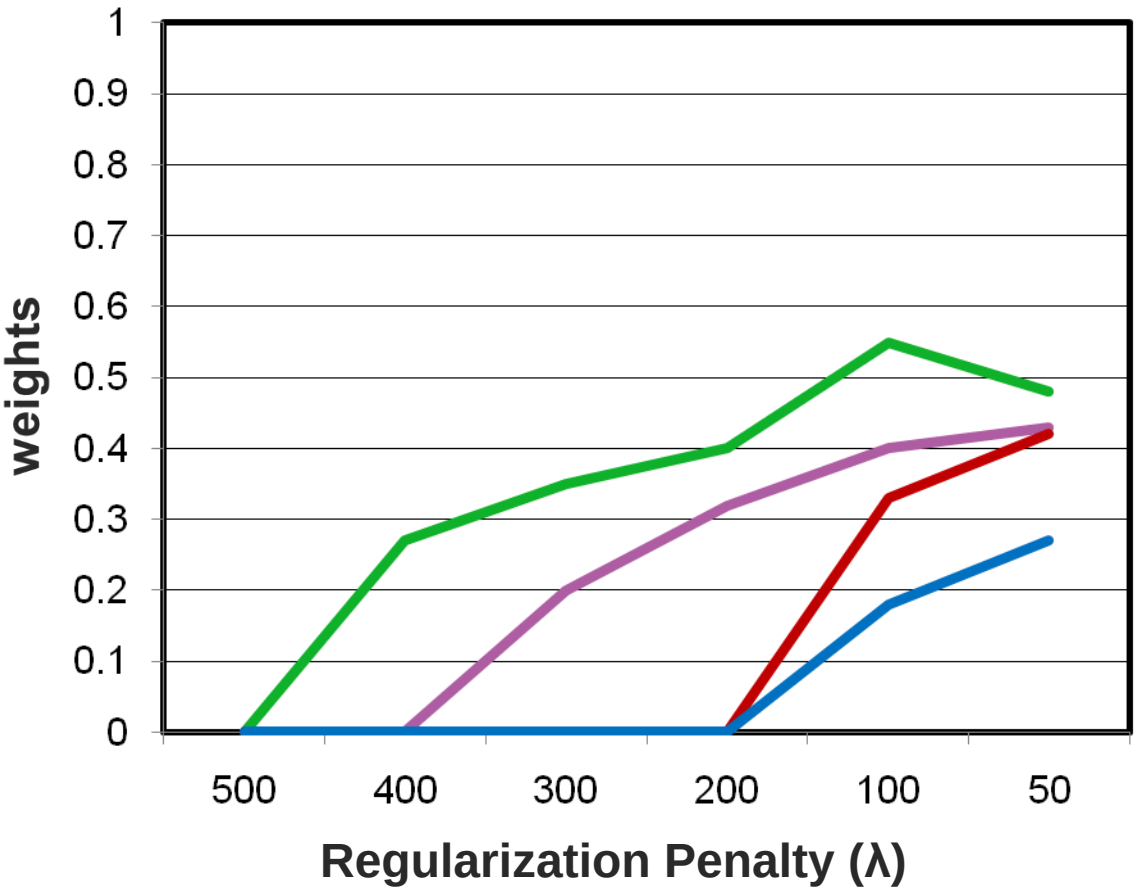
Sparse Feature Ranking



Sparse Feature Ranking

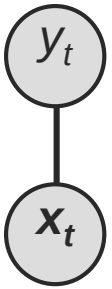


Sparse Feature Ranking



Pause	0	0	0.20	0.40
EyeGaze	0	0.27	0.35	0.55
POS:NN	0	0	0	0.33
'and'	0	0	0	0.18

LASSO and ElasticNet



Label : {0:`Dominant, 1:Not-dominant`}

Observation vector: [speech-energy, gaze, turn-taking]

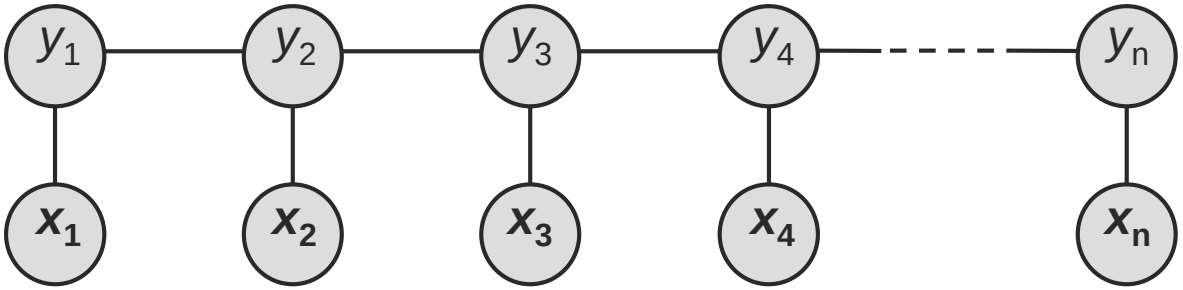
Lasso loss function: squared loss with L1 regularization

$$L(\theta) = \sum_{j=1}^{N} \left(y_j - f\left(\boldsymbol{x_j}; \theta\right)\right)^{\mathbf{2}} - \lambda \|\theta\|_1$$

ElasticNet: squared loss with L1 and L2 regularization

$$L(\theta) = \sum_{j=1}^{N} \left(y_j - f\left(\boldsymbol{x_j}; \theta\right)\right)^{\mathbf{2}} - \lambda \|\theta\|_1 - \lambda \|\theta\|_2$$

Conditional Random Fields (CRFs) [McCallum 2001]



$$P(\boldsymbol{y}|\boldsymbol{X}) = \frac{1}{\mathcal{Z}(\boldsymbol{X})} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, \boldsymbol{x_t}) + \sum_{l=1}^{L} \lambda_l g_l(y_t, y_{t-1})\right)$$

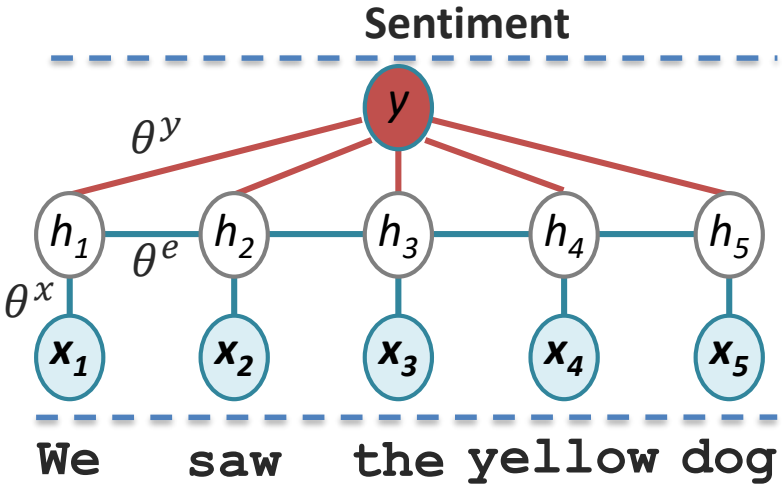
Hidden Conditional Random Field

Sequence label:

$y \in \mathcal{Y}$ for example, $\mathcal{Y}$: {positive, negative}

Latent variables with shared hidden states:

$\boldsymbol{h} = \{h_1, h_2, h_3, \ldots, h_t\}$ where $h_t \in \mathcal{H}$



$$p(y, \boldsymbol{h} \mid \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{\mathcal{Z}(\boldsymbol{x}; \boldsymbol{\theta})} \exp\left\{\sum_t \boldsymbol{\theta}^x \cdot f^x(h_t, \mathbf{x_t}) + \sum_t \boldsymbol{\theta}^e \cdot f^e(h_t, h_{t-1}, y) + \sum_t \boldsymbol{\theta}^y \cdot f^y(y, \boldsymbol{h_t})\right\}$$

$$p(y \mid \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{h}} p(\quad$$

Shared hidden states

- **Inference is tractable: $O(YH^2T)$**
  - **Linear in sequence length T !**
- **Parameter learning ($\boldsymbol{\theta^x}, \boldsymbol{\theta^e}, \boldsymbol{\theta^y}$):**
  - **Gradient descent or L-BFGS**

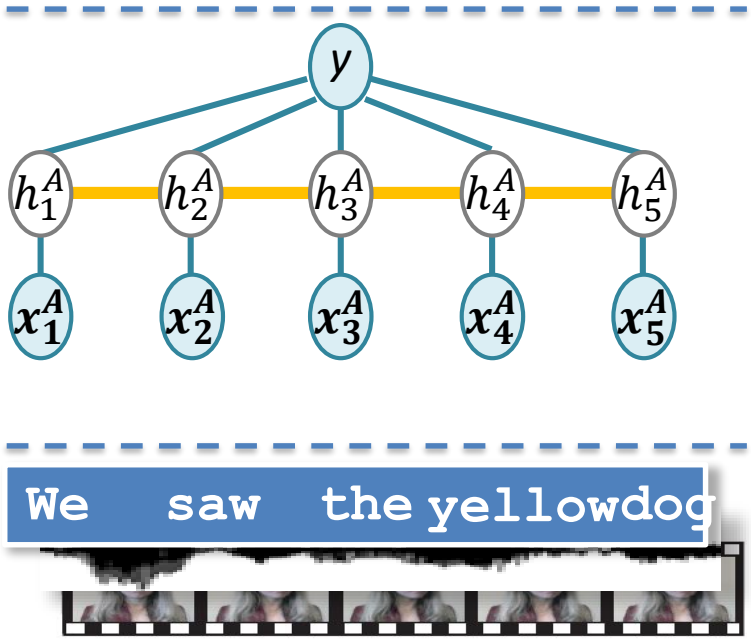
Learning Multimodal Structure

Modality-***private*** structure

- Internal grouping of observations

Modality-***shared*** structure

- Interaction and synchrony



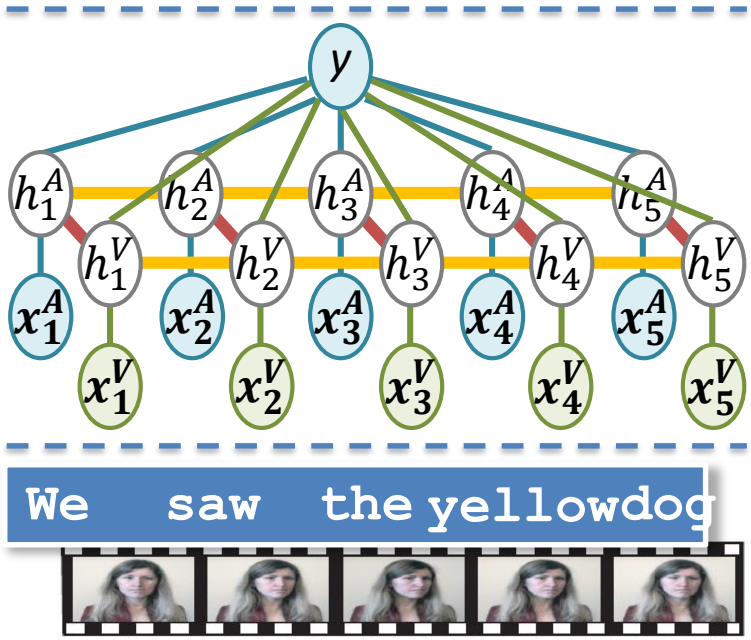
Multi-view Latent Variable Discriminative Models

Modality-***private*** structure

- Internal grouping of observations

Modality-***shared*** structure

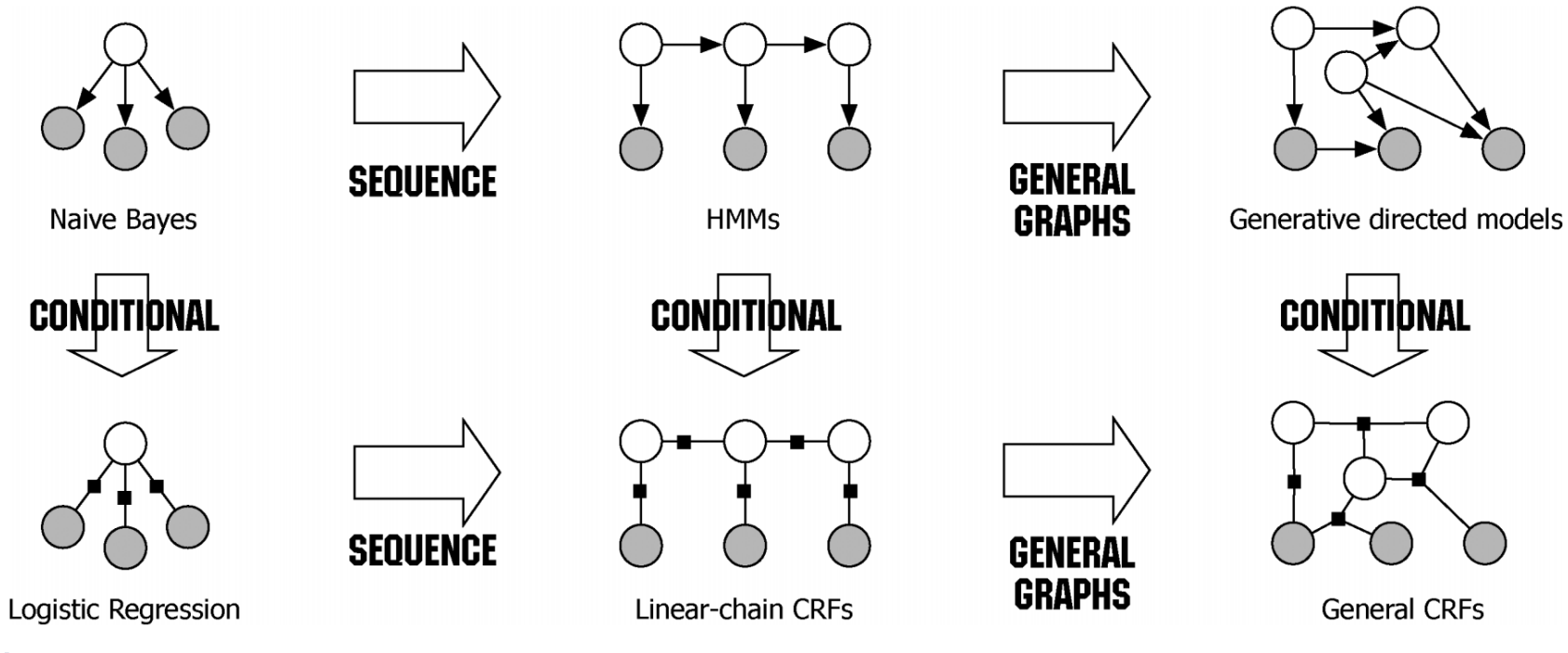
- Interaction and synchrony



$$p(y|\, \boldsymbol{x^A}, \boldsymbol{x^V}; \boldsymbol{\theta}) = \sum_{\boldsymbol{h^A}, \boldsymbol{h^V}} p\left(y, \boldsymbol{h^A}, \boldsymbol{h^V} \middle| \boldsymbol{x^A}, \boldsymbol{x^V}; \boldsymbol{\theta}\right)$$

➢ Approximate inference using loopy-belief

Recap of generative vs discriminative



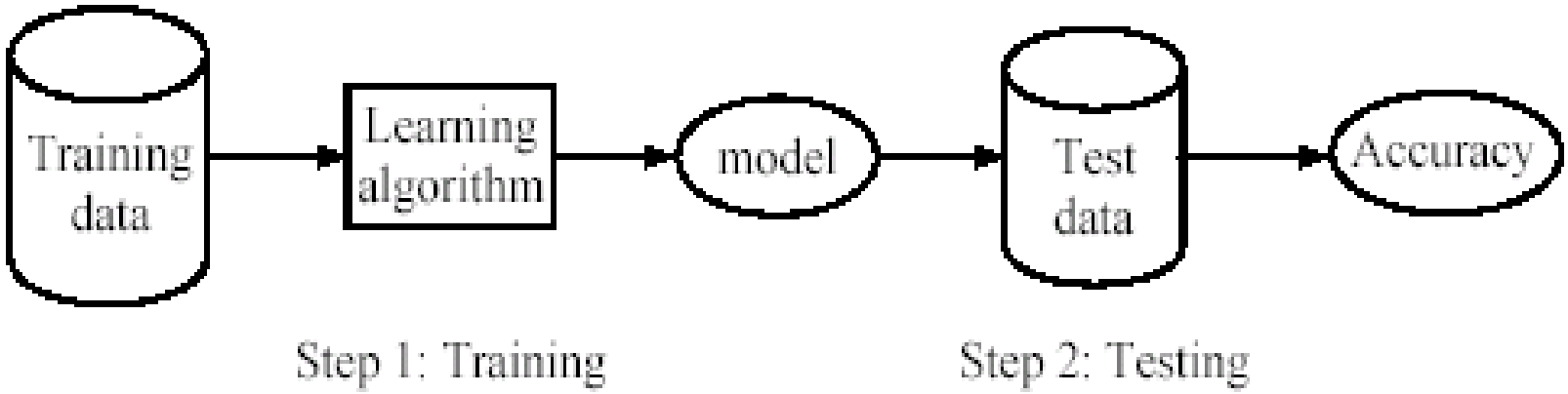
Machine Learning: Evaluation Methods

Supervised learning process: two steps

Learning (training): Learn a model using the training data

Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



Evaluation methods

- Holdout set: The available data set *D* is divided into two disjoint subsets,
  - the *training set* D_{train} (for learning a model)
  - the *test set* D_{test} (for testing the model)
- Important: training set should not be used in testing and the test set should not be used in learning.
  - Unseen test set provides a unbiased estimate of accuracy.
- The test set is also called the holdout set. (the examples in the original data set *D* are all labeled with classes.)
- This method is mainly used when the data set *D* is large.
- **Unless building person specific models the training and test sets should not contain the same person**

Evaluation methods (cont…)

- n-fold cross-validation: The available data is partitioned into *n* equal-size disjoint subsets.
- Use each subset as the test set and combine the rest *n*-1 subsets as the training set to learn a classifier.
- The procedure is run *n* times, which give *n* accuracies.
- The final estimated accuracy of learning is the average of the *n* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large.

Evaluation methods (cont…)

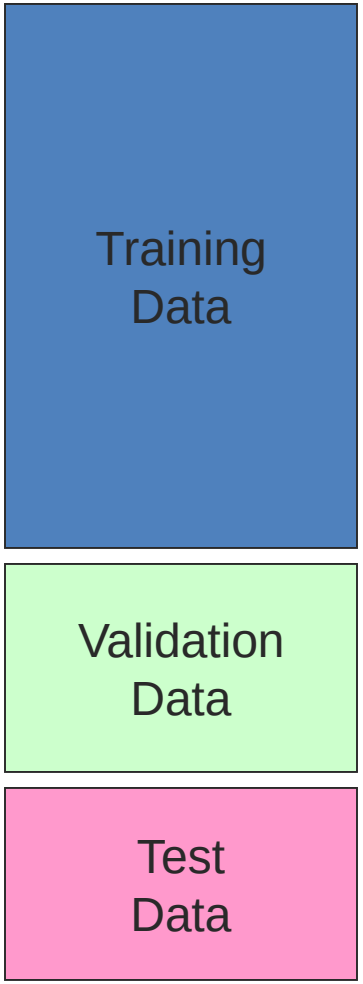
- Leave-one-out cross-validation: This method is used when the data set is very small.
- It is a special case of cross-validation
- Each fold of the cross validation has only a single test example and all the rest of the data is used in training.
- If the original data has *m* examples, this is *m*-fold cross-validation

Hyperparameters

- How do we determine C or γ for SVM training?
- Parameters that we do not learn through optimization are called hyper-parameters
- Need a way to find optimal values for our task
  - For some approaches rules of thumb exist
- Need an analytical way to do it
- Common ways
  - Grid search
  - Random search (not as bad as it sounds)

Training and Validation

- Data: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Validation set
  - Test set
- Training
  - Estimate parameters on training set
  - Tune hyperparameters on validation/development set
  - Report results on test set
  - Anything short of this yields over-optimistic claims
- Evaluation
  - Many different metrics
  - Ideally, the criteria used to train the classifier should be closely related to those used to evaluate the classifier
- Statistical issues
  - Want a classifier which does well on *test* data
  - Overfitting: fitting the training data very closely, but not generalizing well
  - Error bars: want realistic (conservative) estimates of accuracy



Take home

- **1. Never touch test data during training/validation**
- **2. Never touch test data during training/validation**
- **3. Never touch test data during training/validation**

Machine Learning: Measuring Error

Measuring Error

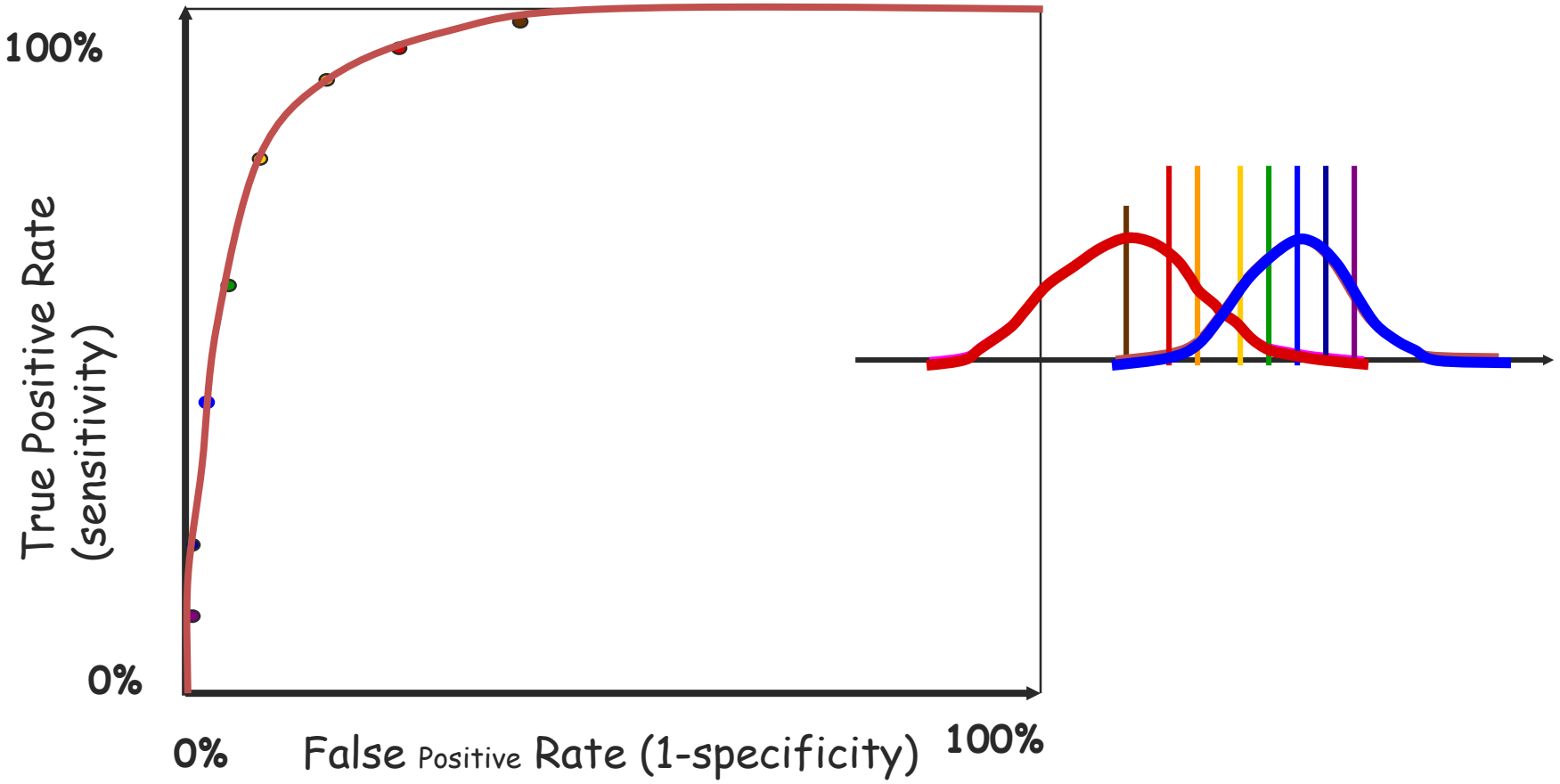
	Predicted class	
True Class	Yes	No
Yes	TP: True Positive	FN: False Negative
No	FP: False Positive	TN: True Negative

- Error rate = # of errors / # of instances = (FN+FP) / N
- Recall = # of found positives / # of positives
  = TP / (TP+FN) = sensitivity = hit rate
- Precision = # of found positives / # of found
  = TP / (TP+FP)
- Specificity = TN / (TN+FP)
- False alarm rate = FP / (FP+TN) = 1 - Specificity

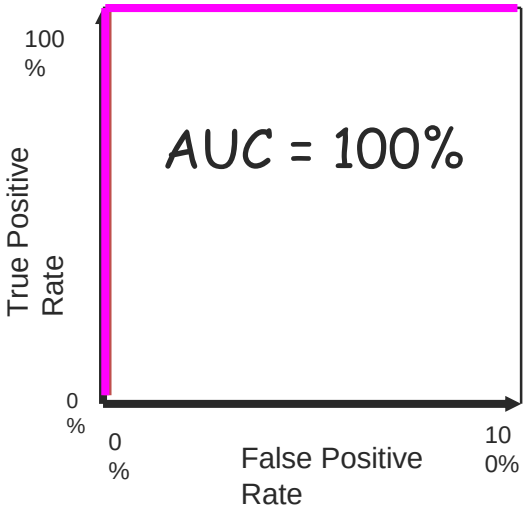
F_1-value (also called F_1-score)

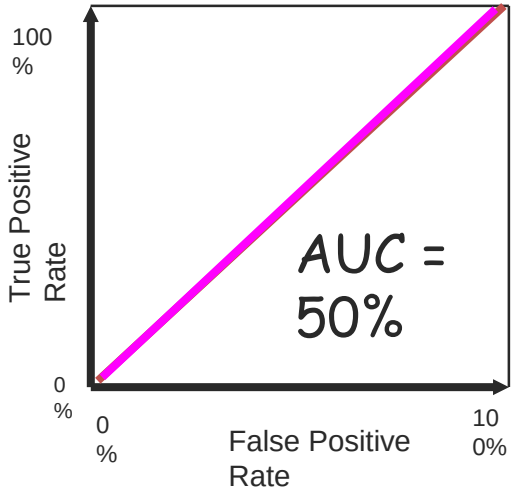
- It is hard to compare two classifiers using two measures. F_1 score combines precision and recall into one measure
  - $F_1 = \frac{2 \cdot p \cdot r}{p + r}$
  - F_1 - score is the harmonic mean of precision and recall
  - $F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$
- The harmonic mean of two numbers tends to be closer to the smaller of the two
- Preferred over accuracy when data is unbalanced
  - Why?

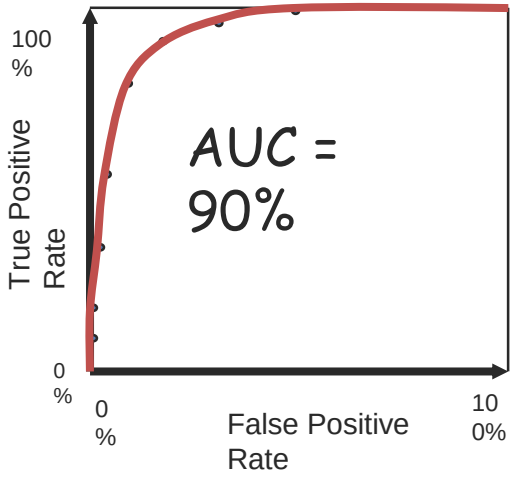
Receiver Operating Characteristic (ROC) Curve

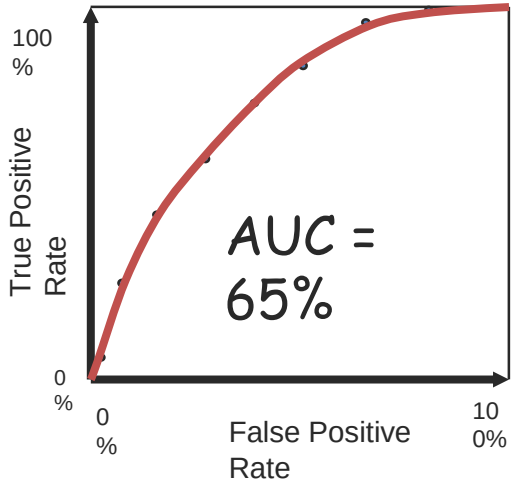


AUC for ROC curves









Evaluation of regression

- Root Mean Square Error
  - $\sqrt{\sum_i (y_i - x_i)^2}$
  - Not easily interpretable
- Correlation – trend prediction in a way
  - Nice interpretation: 0 – no relationship, 1 – perfect relationship
  - $\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sigma_x \sigma_y}$
- Concordance Correlation Coefficient (CCC)
  - A method to combine both
  - $\rho_c = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2}$, ρ – correlation coefficient
  - Has nice interpretability as well

Take home

- Error measure selection is not straightforward
  - Pick the right one for your problem
  - F1, AUC, Accuracy, RMSE, CCC
- Make sure the same measure is used for validation and testing
  - Otherwise you might be learning suboptimal models
- Wrong error measure can hide both bad and good results