Machine Learning for Speech & Language Processing

Mark Gales

28 June 2004

Cambridge University Engineering Department

Foresight Cognitive Systems Workshop

Overview

- Machine learning.
- Feature extraction:
  - Gaussianisation for speaker normalisation.
- Dynamic Bayesian networks:
  - multiple data stream models
  - switching linear dynamical systems for ASR.
- SVMs and kernel methods:
  - rational kernels for text classification.
- Reinforcement learning and Markov decision processes:
  - spoken dialogue system policy optimisation.

Machine Learning

- One definition is (Mitchell):

  "A computer program is said to learn from experience (E) with some class of tasks (T) and a performance measure (P) if its performance at tasks in T as measured by P improves with E"

  alternatively

  "Systems built by analysing data sets rather than by using the intuition of experts"

- Multiple specific conferences:
  - {International,European} Conference on Machine Learning;
  - Neural Information Processing Systems;
  - International Conference on Pattern Recognition etc etc;
- as well as sessions in other conferences:
  - ICASSP - machine learning for signal processing.

“Machine Learning” Community

“You should come to NIPS. They have lots of ideas.
The Speech Community has lots of data.”

- Some categories from Neural Information Processing Systems:
  - clustering;
  - dimensionality reduction and manifolds;
  - graphical models;
  - kernels, margins, boosting;
  - Monte Carlo methods;
  - neural networks;
  - ...
  - speech and signal processing.
- Speech and language processing is just an application

Too Much of a Good Thing?

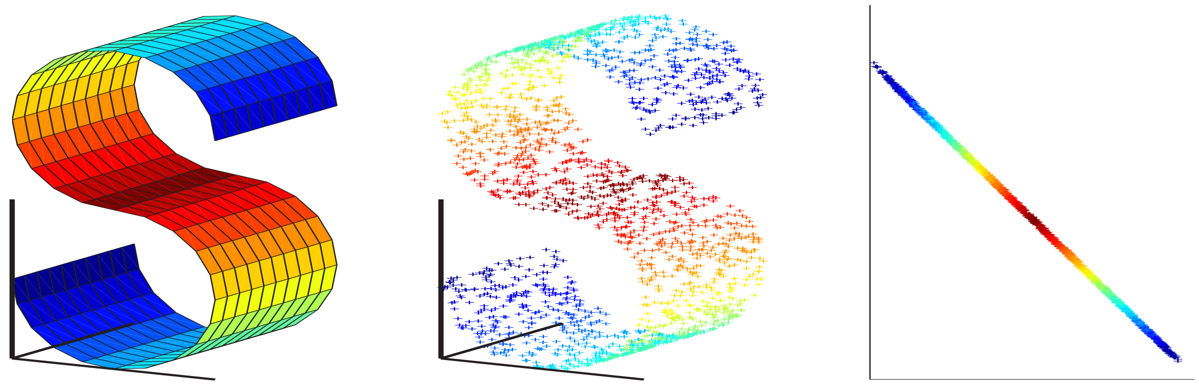
"You should come to NIPS. They have lots of ideas.
Unfortunately, the Speech Community has lots of data."

- Text data: used to train the ASR language model:
  - large news corpora available;
  - systems built on $>$ 1 billion words of data.

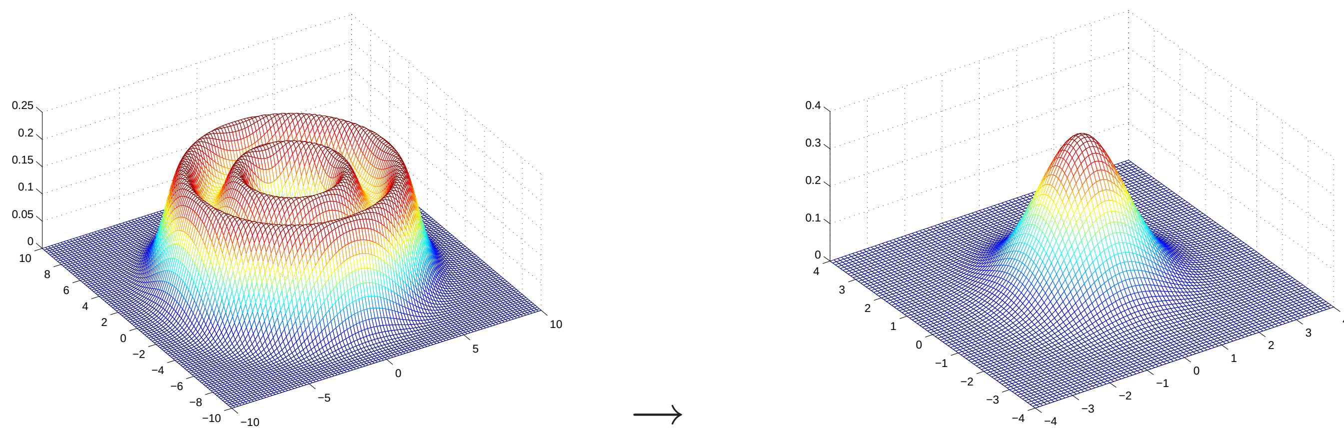
- Acoustic data: used to train the ASR acoustic models:
  - $>$ 2000 hours speech data
    ($\sim$ 20 million words, $\sim$ 720 million frames of data);
  - rapid transcriptions/closed caption data.

- Solutions required to be scalable:
  - heavily influences (limits!) machine learning approaches used;
  - additional data masks many problems!

Feature Extraction



Low-dimensional non-linear projection (example from LLE)

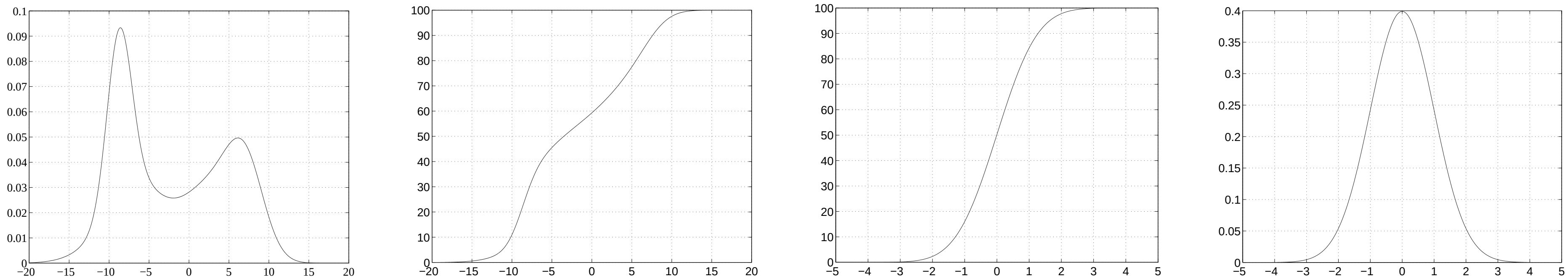


Feature transformation (Gaussianisation)

Gaussianisation for Speaker Normalisation

1. Linear projection and "decorrelation of the data" (heteroscedastic LDA)

2. Gaussianise the data for each speaker:



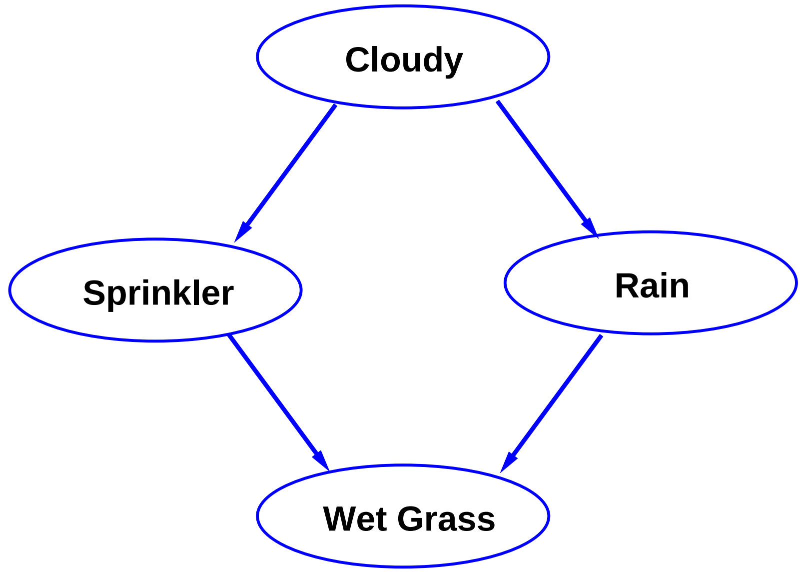
Source PDF | Source CDF | Target CDF | Target PDF

   (a) construct a Gaussian mixture model for each dimension;
   (b) non-linearly transform using cumulative density functions.

- May view as higher-moment version of mean and variance normalisation:
  - single component/dimension GMM equals CMN plus CVN
- Performance gains on state-of-the-art tasks

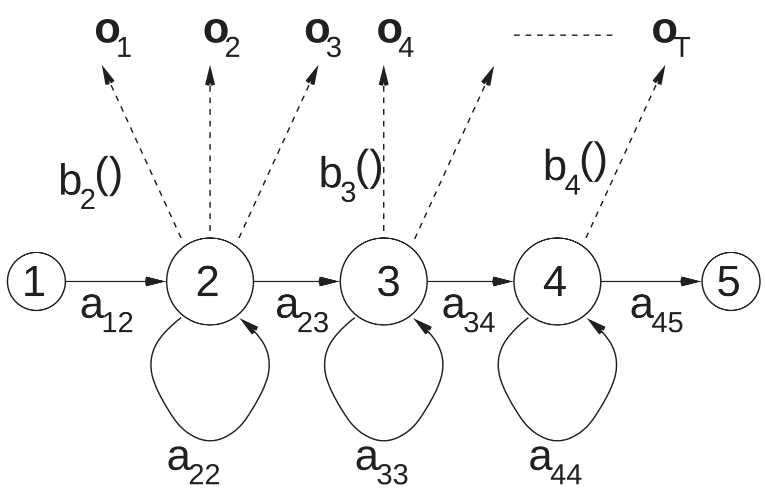
Bayesian Networks

- Bayesian networks are a method to show conditional independence:

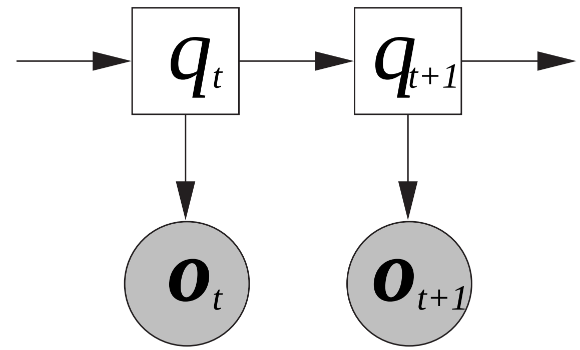


  - whether the grass is wet, W, depends on :
    whether the sprinkler used, S, and whether it has rained; R.
  - whether sprinkler used (or it rained) depends on: whether it is cloudy C.
- W is conditionally independent of C given S and R.
- Dynamic Bayesian networks handle variable length data.

Hidden Markov Model - A Dynamic Bayesian Network



(a) Standard HMM phone topology



(b) HMM Dynamic Bayesian Network

- Notation for DBNs:

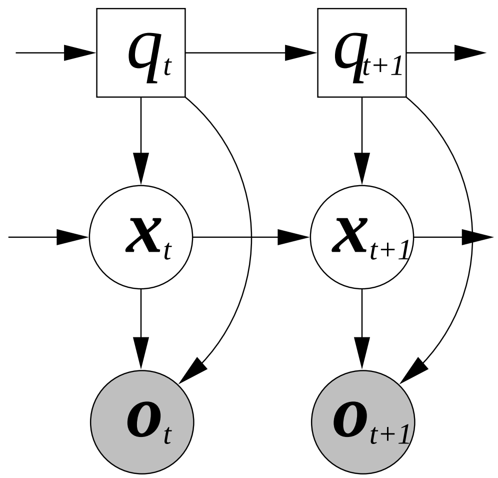
  circles - continuous variables   squares - discrete variables
  shaded - observed variables   non-shaded - unobserved variables

- Observations conditionally independent of other observations given state.
- States conditionally independent of other states given previous states,
- Poor model of the speech process - piecewise constant state-space.

Alternative Dynamic Bayesian networks

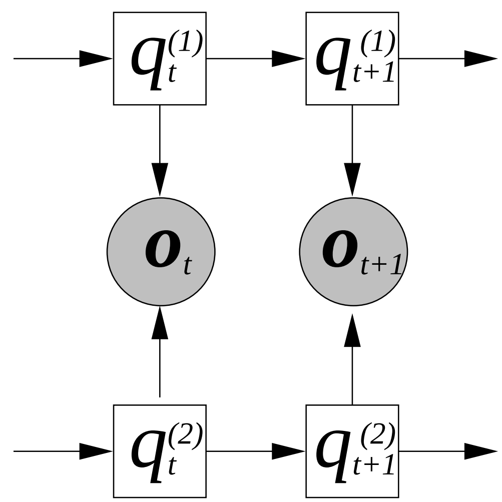
Switching linear dynamical system:

- discrete and continuous state-spaces
- observations conditionally independent given continuous and discretes state;
- exponential growth of paths, $O(N_s^T)$
  $\Rightarrow$ approximate inference required.



Multiple data stream DBN:

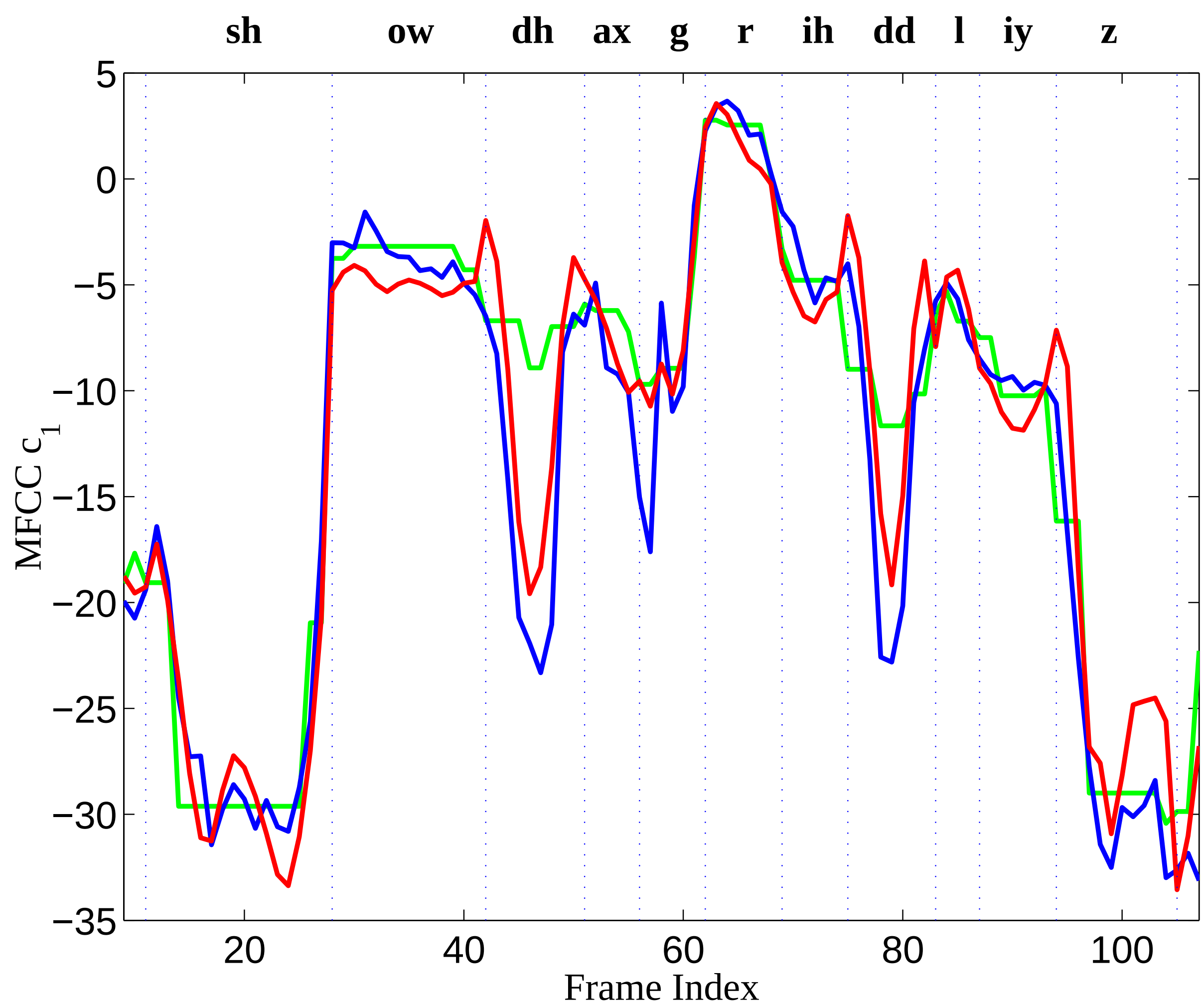
- e.g. factorial HMM/mixed memory model;
- asynchronous data common:
  - speech and video/noise;
  - speech and brain activation patterns.
- observation depends on state of both streams



SLDS Trajectory Modelling

Frames from phrase:

`SHOW THE GRIDLEY'S ...`

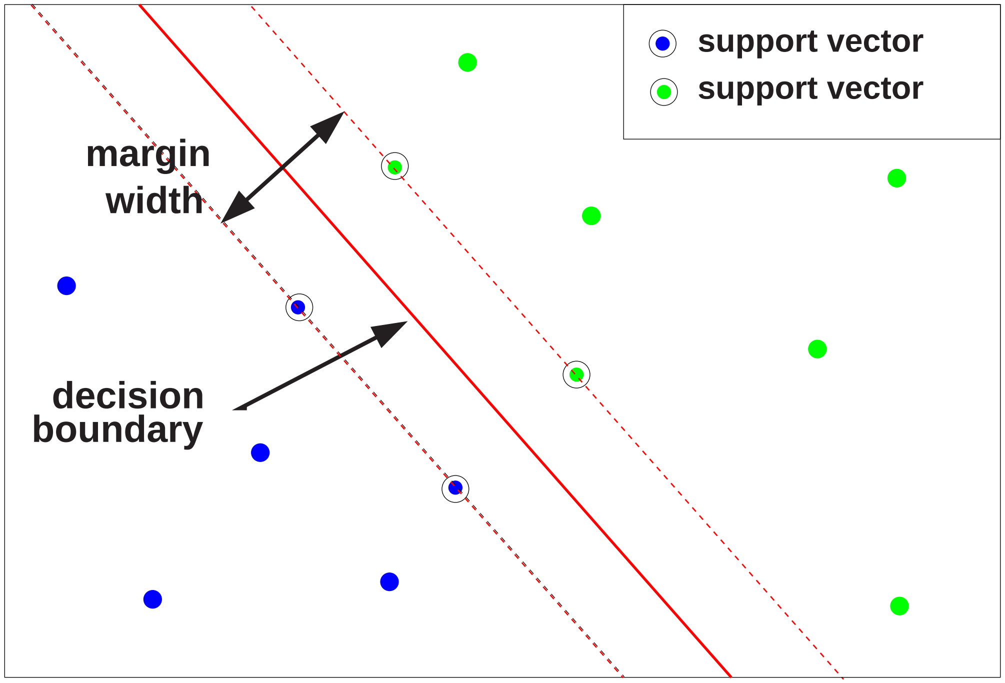


Legend

- True
- HMM
- SLDS

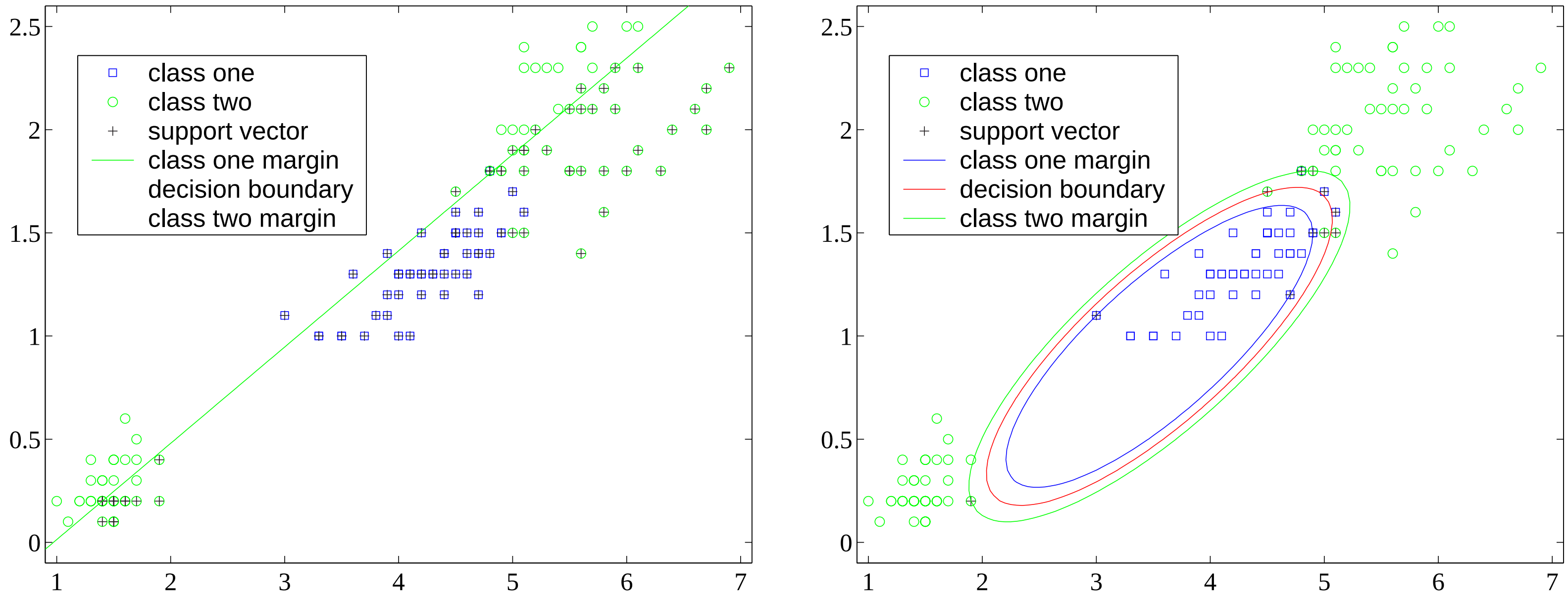
- Unfortunately doesn't currently classify better than an HMM!

Support Vector Machines



- SVMs are a maximum margin, binary, classifier:
  - related to minimising generalisation error;
  - unique solution (compare to neural networks);
  - may be kernelised - training/classification a function of dot-product $(\mathbf{x}_i.\mathbf{x}_j)$.
- Successfully applied to many tasks - how to apply to speech and language?

The "Kernel Trick"



- SVM decision boundary linear in the feature-space
  - may be made non-linear using a non-linear mapping $\phi()$ e.g.

$$\phi\left(\left[\begin{array}{c} x_1 \\ x_2 \end{array}\right]\right) = \left[\begin{array}{c} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{array}\right], \quad K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$$

- Efficiently implemented using a Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j)^2$

String Kernel

- For speech and text processing input space has variable dimension:
  - use a kernel to map from variable to a fixed length;
  - Fisher kernels are one example for acoustic modelling;
  - String kernels are an example for text.
- Consider the words `cat`, `cart`, `bar` and a character string kernel

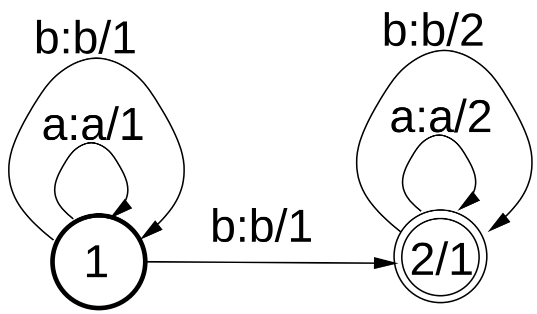
	c-a	c-t	c-r	a-r	r-t	b-a	b-r
$\phi(\texttt{cat})$	1	λ	0	0	0	0	0
$\phi(\texttt{cart})$	1	λ^2	λ	1	1	0	0
$\phi(\texttt{bar})$	0	0	0	1	0	1	λ

$$K(\texttt{cat},\texttt{cart}) = 1+\lambda^3, \quad K(\texttt{cat},\texttt{bar}) = 0, \quad K(\texttt{cart},\texttt{bar}) = 1$$

- Successfully applied to various text classification tasks:
  - how to make process efficient (and more general)?

Weighted Finite-State Transducers

- A weighted finite-state transducer is a weighted directed graph:
  - transitions labelled with an input symbol, output symbol, weight.
- An example transducer, T, for calculating binary numbers: a=0, b=1



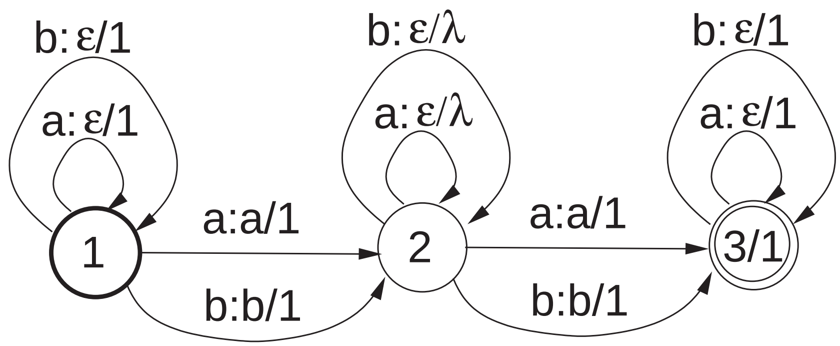
Input	State Seq.	Output	Weight
bab	1 1 2	bab	1
	2 1 1	bab	4

For this sequence output weight: $w\left[\text{bab} \circ T\right] = 5$

- Standard (highly efficient) algorithms exist for various operations:
  - combining transducer, $T_1 \circ T_2$;
  - inverse, T^{-1}, swap the input and output symbols in the tranducer.
- May be used for efficient implementation of string kernels.

Rational Kernels

- A transducer, T, for the string kernel (gappy bigram) (vocab $\{\mathtt{a}, \mathtt{b}\}$)

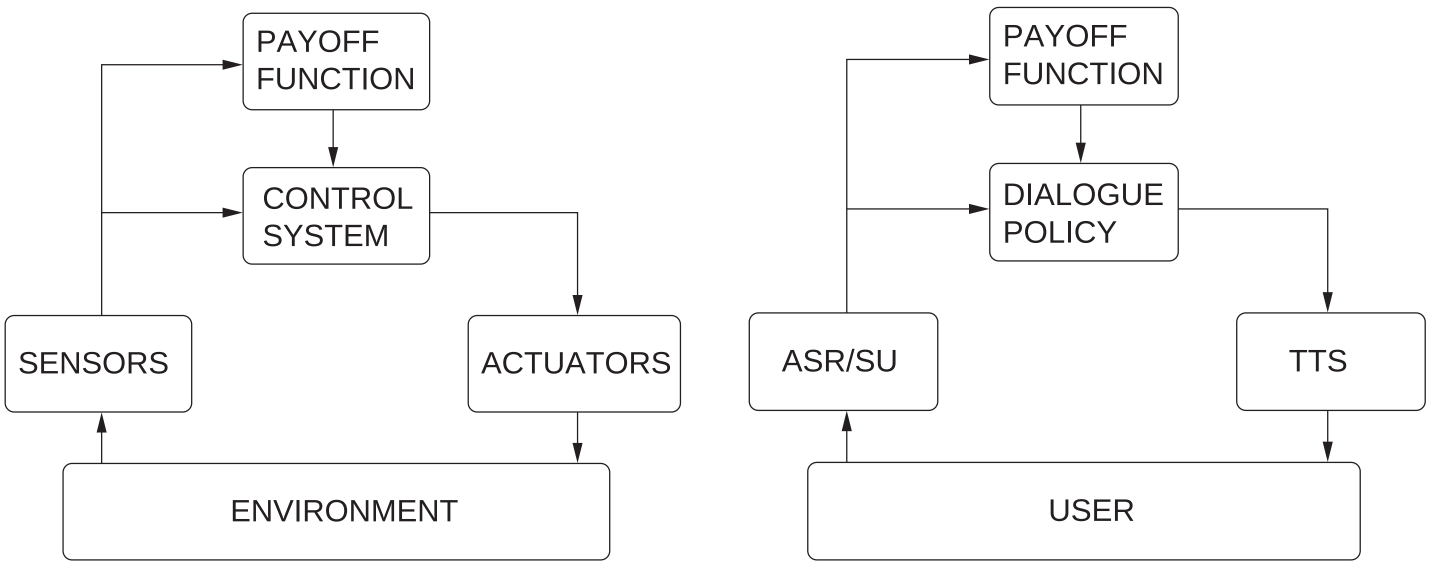


The kernel is: $K(\boldsymbol{O}_i, \boldsymbol{O}_j) = w\left[\boldsymbol{O}_i \circ (T \circ T^{-1}) \circ \boldsymbol{O}_j\right]$

- This form can also handle uncertainty in decoding:
  - lattices can be used rather than the 1-best output ($\boldsymbol{O}_i$).
- This form encompasses various standard feature-spaces and kernels:
  - bag-of-words and N-gram counts, gappy N-grams (string Kernel),
- Successfully applied to a multi-class call classification task.

Reinforcement Learning

- Reinforcement learning is a class of training methods:
  - problem defined by payoffs;
  - aims to learn the policy that maximises the payoff;
  - no need for a mathematical model of environment.



Reinforcement learning   Spoken dialogue system

- Dialogue policy learning fits nicely within this framework.

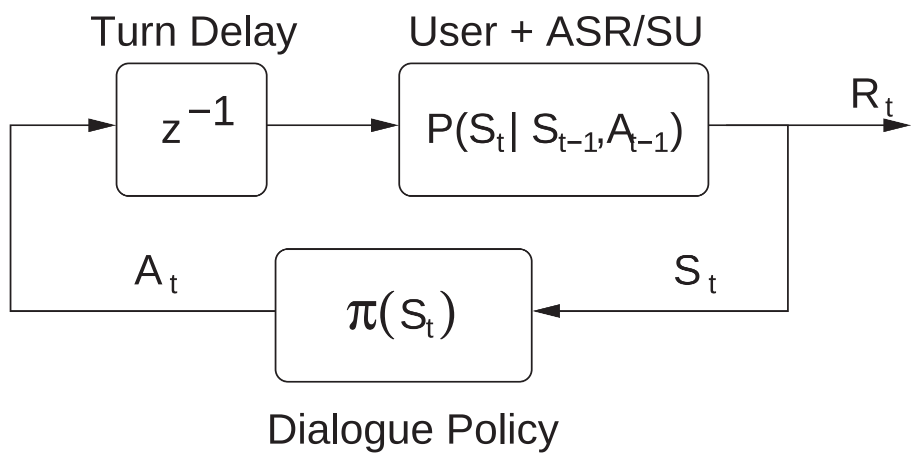
Example Dialogue

```
S1: Welcome to NJFun. How may I help you?
U1: I'd like to find um winetasting in Lambertville in the morning
[ ASR: I'd like to find out wineries in the Lambertville in the
       morning ]
S2: Did you say you are interested in Lambertville?
U2: Yes.
S3: Did you want to go in the morning?
U3: Yes.
S4: I found a winery near Lambertville that is open in the morning
    It is Poor Richard's Winery in Lambertville.
```

- Variety of action choices available:
  - mixed versus system initiative;
  - explicit versus implicit confirmation.

Markov Decision Process

- SDS modelled as a MDP:
  - system state and action at time t: S_t and A_t;
  - transition function: user and ASR/SU model, $P(S_t|S_{t-1}, A_{t-1})$.



- Select policy to maximise expected total reward:
  - total reward: R_t sum of instantaneous rewards from t to end of dialogue;
  - value function (expected reward) for policy π in state S: $V^\pi(S)$.

Q-Learning

- In reinforcement learning use the Q-function, $Q^{\pi}(S, A)$
  - expected reward from taking action A in state S using policy π
- Best policy using π given state S_t is obtained from

$$\hat{\pi}(S_t) = \underset{A}{\operatorname{argmax}} \left(Q^{\pi}(S_t, A) \right)$$

- Transition function not normally known - one-step Q-learning algorithm:
  - learn $Q^{\pi}(S, A)$ rather than transition function;
  - estimate using difference between actual and estimated values.
- How to specify reward: simplest form assign to final state:
  - positive value for task success;
  - negative value for task failure.

Partially Observed MDP

- State-space required to encapsulate all information to make decision:
  - state space can become very large e.g. transcript of dialogue to date etc;
  - required to compress size - usually application specific choice;
  - if state-space is too small MDP not appropriate.
- Also User beliefs cannot be observed:
  - decisions required on incomplete information (POMDP);
  - use of a belief state - value function becomes

$$V^{\pi}(B) = \sum_{S} B(S) V^{\pi}(S)$$

  where $B(S)$ gives belief in a state.
- Major problem: how to obtain sufficient training data?
  - build prototype system and then refine;
  - build a user model to simulate user interaction.

Machine Learning for Speech & Language Processing

Briefly described only a few examples

- Markov chain Monte-Carlo techniques:
  - Rao-Blackwellised Gibbs sampling for SLDS one example.
- Discriminative training criteria:
  - use criteria more closely related to WER, (MMI, MPE, MCE).
- Latent variable models for language modelling:
  - Latent semantic analysis (LSA) and Probabilistic LSA.
- Boosting style schemes:
  - generate multiple complementary classifiers and combine them.
- Minimum Description Length & evidence framework:
  - automatically determine numbers of model parameters and configuration.

Some Standard Toolkits

- Hidden Markov model toolkit (HTK)
  - building state-of-art HMM-based systems
  - `http://htk.eng.cam.ac.uk/`
- Graphical model toolkit (GMTK)
  - training and inference for graphical models
  - `http://ssli.ee.washington.edu/~bilmes/gmtk/`
- Finite state transducer toolkit (FSM)
  - building, combining, optimising weighted finite state transducers
  - `http://www.research.att.com/sw/tools/fsm/`
- Support vector machine toolkit (SVM^{light})
  - training and classifying with SVMs
  - `http://svmlight.joachims.org/`

(Random) References

- *Machine Learning*, T Mitchell, McGraw Hill, 1997.
- *Nonlinear dimensionality reduction by locally linear embedding*, S. Roweis and L Saul Science, v.290 no.5500 , Dec.22, 2000. pp.2323–2326.
- *Gaussianisation*, S. Chen and R Gopinath, NIPS 2000.
- *An Introduction to Bayesian Network Theory and Usage*,T.A.Stephenson,IDIAP-RR03, 2000
- *A Tutorial on Support Vector Machines for Pattern Recognition*, C.J.C. Burges, Knowledge Discovery and Data Mining, 2(2), 1998.
- *Switching Linear Dynamical Systems for Speech Recognition*, A-V.I. Rosti and M.J.F. Gales, Technical Report CUED/F-INFENG/TR.461 December 2003,
- *Factorial hidden Markov models*, Z. Ghahramani and M.I. Jordan. Machine Learning 29:245–275, 1997.
- *The Nature of Statistical Learning Theory*, V. Vapnik, Springer, 2000.
- *Finite-State Transducers in Language and Speech Processing*, M. Mohri,Computational Linguistics, 23:2, 1997.
- *Weighted Automata Kernels - General Framework and Algorithms*, C. Cortes, P. Haffner and M. Mohri, EuroSpeech 2003.
- *An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email*, M. Walker, Journal of Artificial Intelligence Research, JAIR, Vol 12., pp. 387-416, 2000.