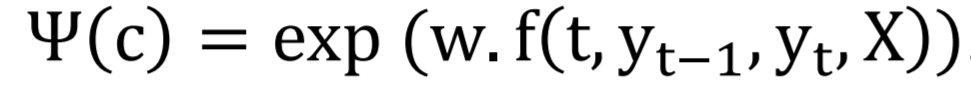
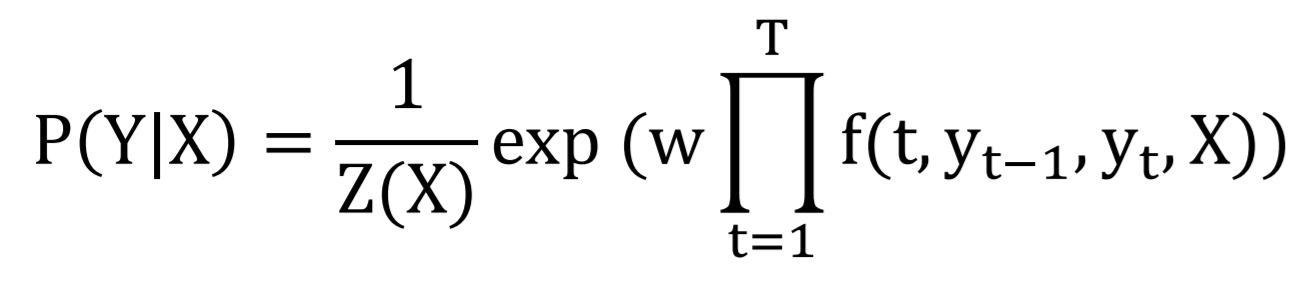
**Paper Review: “Conditional Random Fields for Action Recognition in Smart Environments”**

* Common function of smart environments is to monitor/assist older adults with daily activities
  + Activity recognition is a key component in this application
* Activity recognition essentially a temporal (sequence) classification problem
  + In the past, modelled by Naïve Bayes and HMMs
* Here, describes use of CRFs, which is popular for HAR
  + Focus here is on using CRFs to recognize activities performed by an inhabitant in smart home environment
  + Goal is to validate claim of CRF’s higher or similar performance w.r.t. HMMs
* Growing interest in development of smart environments capable of reasoning about inhabitants to provide health monitor/assistance
* Can be used to monitor inhabitant’s activities and remotely assess their functional well-being
* Motivated by aging of population, cost of formal healthcare, and importance of people being independent in their own homes
  + Necessary, therefore, to recognize activities they usually perform
  + Thus, aim of activity recognition in smart home setting is to recognize activities of daily living (ADL)
* Observation data sequences obtained by various sensors and annotated w/ corresponding activity labels and are used to train different prediction models
* New sensor technologies and recognition algorithms include: stereo cameras for tracking, watches for activity recognition, pressure sensors in smart floor, probabilistic models of activities, etc
* Probabilistic models are best used generally for this task because sensor readings are usually noisy and activities commonly performed in non-deterministic way
  + HMMs and CRFs are the most common ones
* CRFs proposed originally for NLP, though are expanding in use to other areas like HAR< although typically dependent on costly accelerometer and RFID sensors
* Here, considering only linear-chain CRFs and application to field of HAR in the smart home
  + Also, compare HMMs vs. CRFs for HAR
* Smart home tested is 3-story house w/ a man, woman, and a pet
  + Couple’s children also regularly visit the home
  + Sensor events are generated from motion sensors and from temperature sensors
  + Mobility tracked via motion sensors in wall and ceiling to allow for tracking of people moving across the space
  + Includes analog sensors to provide temperature readings and hot water, cold water, and stove burner use
* Sensor network captures all sensor events and stores in a database
  + Each sensor data gathered is expressed by date, time, sensor ID, and sensor value
* After collection, data is then annotated for HAR based on people’s activities
  + High quality annotated data needed for probabilistic model’s learning algorithms
* W/o visualizer tool, hard to interpret raw data as activities
* Use Python visualizer ‘PyViz’ to visualize sensor events and hence help improve quality of annotated data
  + PyViz can display events in real-time or playback mode from captured file of sensor event readings
  + Also built annotation visualizer to visualize residents’ activities through annotated file
* Activity labels are optionally added at the end of each sensor event, marking status of activity as one of 8 labels: bed to toilet, breakfast, sleeping, computer work, dinner, lunch, night wandering, and taking medicine
* Features used to classify activities either generated directly from a single event or by considering a set of sensor events
* Features being considered for each event are:
  + Sensor (logical label identifying the involved sensor)
  + Time of data (discretized value of time event occurs)
  + Day of week (int value for day in question)
  + Previous activity (activity that occurred immediately before current)
  + Act length (in terms of # of generated sensor events)
* Collected data while 2 residents were living here and gathered over several months
  + More than 100K sensor events generated
* HMMs = generative models used to generate a sequence of hidden states from observable sequences
  + Strong applications in speech recognition, cryptanalysis, machine translation, and bioinformatics
* For HAR, hidden states + observations correspond to activity labels and sensor data (features), respectively
  + Therefore, given input sequence of sensor event observations, goal is to find the most likely sequence of hidden states (i.e. human activities) which could have generated the observed event sequence
* HMMs represent join prob distrib P(X,Y), where X refers to observations and Y refers to label sequences
* To calculate joint distribution, HMM considers all possibly observation sequences
  + Since it is infeasible in terms of complexity, HMM assumes observations are independent from each other but dependent only in corresponding label
* CRFs combine advantages of maximum entropy Markov models (MEMMs) w/o suffering from the label bias problem
* Discriminative models that condition probabilities to the observation sequence
  + Avoid computing the probs for every possibly observation sequence; rather than relying on P(X,Y), specify prob of label sequences given observations as P(Y|X)
* In smart home setting, also very important to consider features that link state transitions in the model directly to the observations
  + Such features are difficult to represent in HMM due to the way it factorizes probabilities, but can be handled by CRFs
* Given CRF as undirected graph G=(V, E) of form:
  + 
  + i.e. factorized maximal cliques, where the potential of each clique takes the form:
  + 
  + Which gives, given the Markov assumption between label sequences Y:
  + 
  + W = weight for feature, Z(X) = normalization term
* Can extend this to be over multiple features, i.e. add an ‘ between the ‘1/Z(X)’ and ‘exp’ above, where K= # of features, T = # of labels
* MLE most commonly used for the purpose of training to estimate the weight vector ‘w’
* Take log likelihood of above, which is convex over entire parameter space, so first order methods like gradient ascent/descent are directly applicable
* Differentiating w.r.t. ‘w’ gives max likelihood solution
  + Can be inferred that the expected value of the feature from train set must be equal to expected value of same features under the model
* Gradient computation is essentially an exponential sum over all possible sequences which are usually solved by dynamic programming algorithms like ‘forward-backward’ algorithm:
  + Large state spaces not ideal for this, as might prove to be expensive because of requirement of many calls of forward-backward algorithm
  + Hence, train complexity overwhelming for CRF and can perform worse than HMM so a sparse forward-backward technique can be used for fast training
* For the given data, the CRF is trained on the annotated data and features used are fed to algorithm to generate the CRF model
* Might be a substantial amount of interdependence between features, which typically combine info from more than 1 event
  + Unlike HMM, CRF can handle these dependences
* Used 3-fold cross validation for both HMM and CRF for all activities
* Found 30 CRF training iterations to be good trade-off between performance of algorithm and running time
* CRF gave average accuracy as high as 91% for all activities as compared to HMM which gave 82%
  + Only 2 activities (bed to toilet and taking medicine) that had HMM as better
  + This is due to both activities not involving many different types of sensors, therefore independence assumption of HMMs works better than dependency criteria of CRFs
* CRFs concluded to work quite well for data streams
* Can further use them to assess the completeness of activities (comparing steps w/ assessment by psychologist of how well it was completed)

**Significant Points and Takeaways from Paper**

* Observation data sequences obtained by various sensors and annotated w/ corresponding activity labels and are used to train different prediction models
  + Sensor events are generated from motion sensors and from temperature sensors
  + Collected data while 2 residents were living here and gathered over several months (>100K sensor events generated)
* Probabilistic models are best used generally for this task because sensor readings are usually noisy, and activities commonly performed in non-deterministic way
* To calculate joint distribution, HMM considers all possibly observation sequences
  + Since it is infeasible in terms of complexity, HMM assumes observations are independent from each other but dependent only in corresponding label
* CRFs avoid computing the probs for every possibly observation sequence; rather than relying on P(X, Y), specify prob of label sequences given observations as P(Y|X)
* In smart home setting, also very important to consider features that link state transitions in the model directly to the observations
  + Such features are difficult to represent in HMM due to the way it factorizes probabilities, but can be handled by CRFs
* Might be a substantial amount of interdependence between features, which typically combine info from more than 1 event
  + Unlike HMM, CRF can handle these dependences
* CRF gave average accuracy as high as 91% for all activities as compared to HMM which gave 82%
  + Only 2 activities (bed to toilet and taking medicine) that had HMM as better
  + This is due to both activities not involving many different types of sensors, therefore independence assumption of HMMs works better than dependency criteria of CRFs