Announcements re Lab 2 (due Friday 1/21)

• If you use text (vs. indexed Matlab version), skip headers and non-text files (via flags in Rainbow)

• For people having trouble compiling Rainbow, a link to tweaked source code has been added

• You are not restricted to Rainbow & Matlab
Topics for Today

• Lab 1 Solutions
• More on Vector Classifiers
• Features
• Classification vs. Posteriors vs. Scoring
• Evaluation
Lab 1 – Part 1

• Trigram perplexity on the eval set range:
  – 79-82: using modified KN smoothing or another variation chosen in comparative expts
  – 84: using default Good-Turing smoothing

• Minor variations in results due to whether the vocabulary is specified in n-gram count (implementation issue related to OOV handling)
Lab 1 – Part 2

• Best
  – Mixture + Cache (72-73)

• Other good choices
  – Cache LM (74-76)
  – Switching long/short trigram (76)
  – Stemmer+trigram (77)
  – 4-gram (77)
  – Swb/news mixture (79)

• General findings:
  – Cache wt = 0.1-0.13
  – Swbd mix wt = 0.8-0.9

• Degradation
  – Factored LM: 290
  – Class LM w/ 200 auto classes (106)

• Vocabulary problems???
  – POS trigram (20)
  – Stemmer+trigram (58)
Lab 1 – Write-Ups

• Objectives for your lab write-ups:
  – Insights on alternatives: you must provide enough detail so we know what you did in order to compare results
  – Communication skills: you must provide detail for duplication and findings for busy people

• Bottom line:
  – Configuration details are important
  – Well motivated conclusions are valuable
Reminder re Vector Classifiers

• What leads to a vector classifier?
  – Any problem where you have a fixed number of “measurements” on the document/sentence/word, etc.
  – Applications such as: sentiment detection, personality detection, word sense disambiguation, sentence boundary detection, dialog act recognition, …. 
  – Sequence models that include vector observations (model component)

• Why should you use vector models?
  – Allows complex features
  – Storage/computation limitations
Knowledge vs. Machine Learning

• 15 years ago NLP was rule-based
• Now >90% of NLP papers are statistical
• A few years ago: data is king
• Current trend: knowledge is in
  – Linguistic models with probabilities
  – Statistical models with linguistic features (not just words)
Today: Focus on Models

• Some important models
  – Naïve Bayes (review)
  – Local learning (memory-based learning)
  – Support vector machine (SVM)
  – Neural networks
  – Rule learning
    – Loglinear, MaxEnt
• Connection to features
  – Model/feature match (e.g. cont vs. discrete vs. mixed)
  – Models as analysis tools
Naïve Bayes (Review)

• MAP rule (min error decision rule)
  \[ c^* = \arg\max_c P(c \mid x_1, x_2, \ldots, x_d) \]
  \[ = \arg\max_c P(x_1, x_2, \ldots, x_d \mid c)P(c) \]

• Assume features are independent
  \[ P(x_1, x_2, \ldots, x_d \mid c) = \prod_{i=1:d} P_i(x_i \mid c) \]

• \( x_i \) can be any feature

• Problem of naïve assumption – confidence
Local Learning

- Nearest neighbor
  - Save all training data $X^1, X^2, \ldots X^N$
  - Compare new $X$ to all training samples
  - Label $X$ with class of closest sample

- Variation: KNN – Find the $K$ nearest neighbors and vote

- Challenges:
  - Storage/computation for large training sets
  - What is the right distance measure?

Name that fruit!
Linear Functions

\[ f(x) = \sum_i w_i x_i + w_0 = w^t x' \quad x' = [x+1] \]

- Decision function: \( f(x) > 0 \) \( \Rightarrow \) class 1 (Perceptron)

- Ranking functions
  \( f(x_i) > f(x_j) \) \( \Rightarrow \) rank(i) > rank(j)
  (Rescoring, regression)
Support Vector Machines

• Goal: Decision boundary with max margin and min training error
  \[
  \text{Min } 0.5\|w\|^2 + C \sum_i H[y_if(x_i)]
  \]

• Extensions
  – Non-linear boundaries with kernels
  – Multi-class decision with multiple binary classes
Neural Network

Linear functions with non-linear operator

Sometimes used to map words to a continuous space
Rule Learning

Decision trees

- Greedy choice of questions to minimize entropy or classification error
- Leaf nodes associated with decision & posterior
- Often used for analyzing feature importance
- High variance $\Rightarrow$ bagging (combining multiple trees)
AdaBoost

• Basic idea
  – Weighted combination of simple classifiers
  – Iterative design:
    • Find best simple classifier
    • Reweight training data based on errors

• Popular simple classifier: decision stump

- $f_1(x) > T$?
  - no
  - yes

- $f_i(x) > T$?
  - no
  - yes

- $f_n(x) > T$?
  - no
  - yes
Loglinear Models

• General form: exponential model
  \[ P(y|x) = K \exp[\sum_i \lambda_i f_i(x,y)] \] (K is normalizing const)

• Common examples
  – Logistic regression
    • Typically binary \( y \), linear function of \( x \)
  – Maximum entropy model
    • Max likelihood solution to weights in exponential model is equivalent to maximizing entropy
Maximum Entropy

• Basic idea: Find $\lambda_i$ to maximize entropy subject to some constraints, e.g.
  – Mean = sample mean & variance = sample variance $\Rightarrow$ Gaussian distribution
  – Expected feature functions = empirical avg
    Language feature functions are often 1/0 indicators (is the bigram = XY?), empirical avg = relative frequency

• Practical problems:
  – Choosing the constraints (overtraining issues)
  – Solving for $\lambda_i$ (iterative scaling, hill climbing,...)
Features

• **Recall**: Features can be any measurement on a “document”

• **Beyond knowledge-driven feature design**, challenges include:
  – Feature selection
  – Feature transformation
  – Feature induction (important for domain transfer, not covered today)
Feature Selection

• From last time: mutual information with class can be used to select features in Naïve Bayes classifiers

• What if features are not independent?
  – Joint feature selection & model design
    • For small problems, certain models (decision trees)
  – Wrapper approach: evaluate feature combination in terms of classifier performance
    • Greedy solutions: forward vs. backward selection
Feature Transformation

• Supervised methods
  – Linear discriminant analysis (LDA, not to be confused with latent Dirichlet allocation)
  – Neural network mappings
  – Posterior probabilities (PLSA)

• Unsupervised methods
  – Principle components analysis, independent components analysis
  – Latent semantic analysis
Vector Models are Multi-Purpose

• Classification
  – Simple decision (2-class, K-class), classifier combination
  – Bootstrapping methods (learning with unlabeled data)

• Posterior prediction
  – Confidence estimates (for decision or learning)
  – Probability output of a component in a bigger model (e.g. in a hidden Markov model)
  – Feature transformation

• Scoring
  – Reranking the output of a simple model with more complex knowledge sources (e.g. in speech recognition, parsing, translation)
Evaluation

• Classifier decisions
  – Percent correct
  – Precision/recall curve or F-measure
  – False/miss error tradeoff curve or equal error rate (EER)

• Posterior prediction
  – Normalized cross entropy
  – Error detection curves