Machine Learning

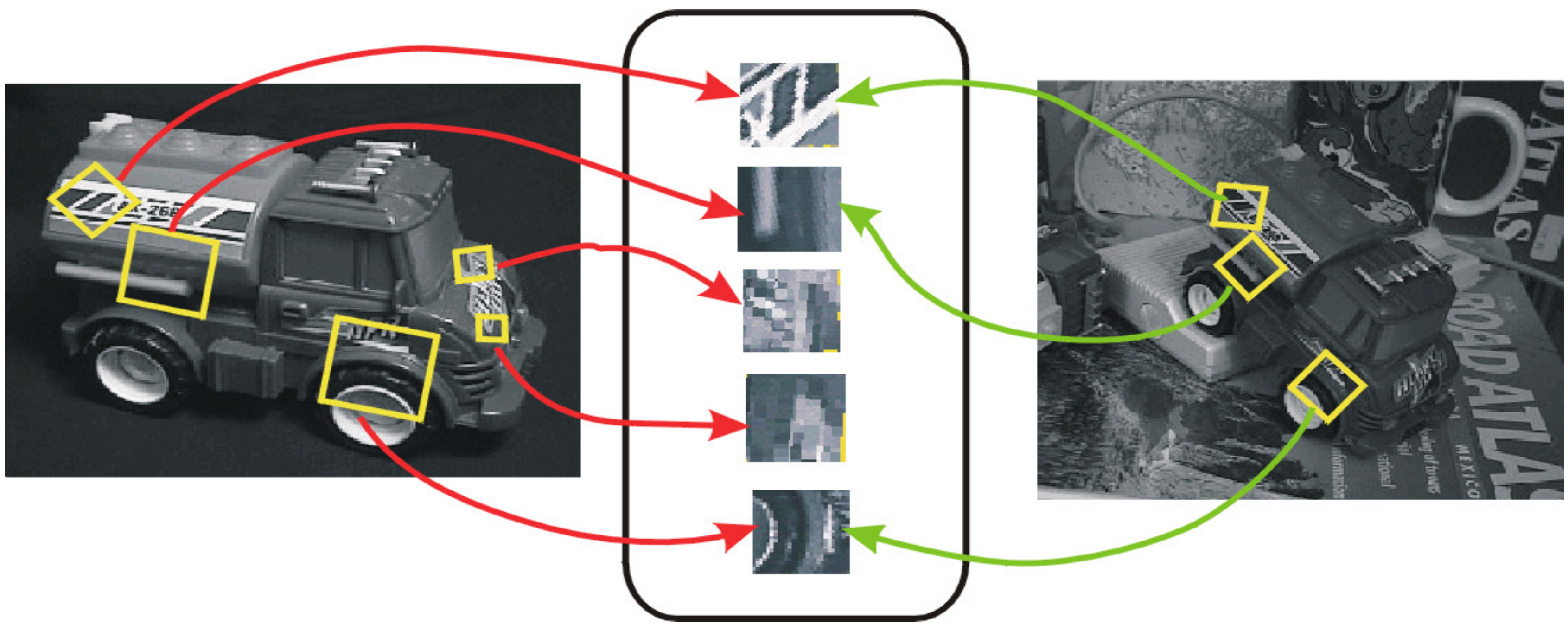
Feature Creation and Selection

Feature creation

- Well-conceived new features can sometimes capture the important information in a dataset much more effectively than the original features.
- Three general methodologies:
  - Feature extraction
    - typically results in significant reduction in dimensionality
    - domain-specific
  - Map existing features to new space
  - Feature construction
    - combine existing features

Scale invariant feature transform (SIFT)

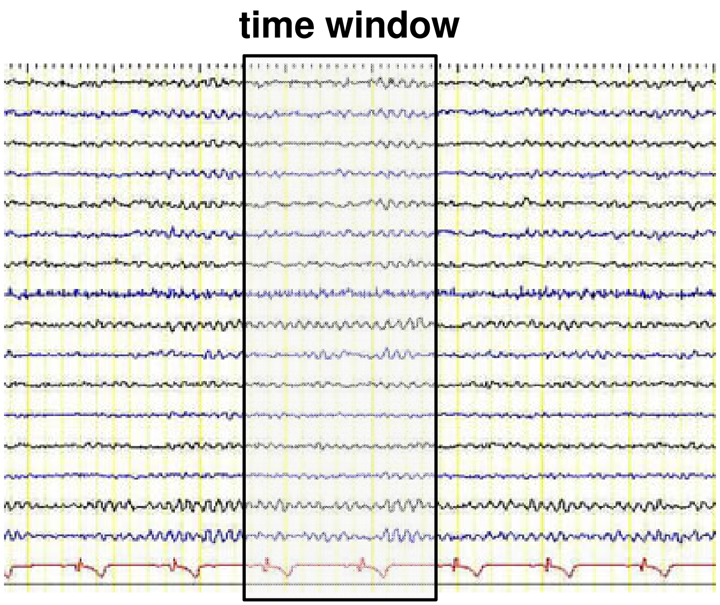
Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.



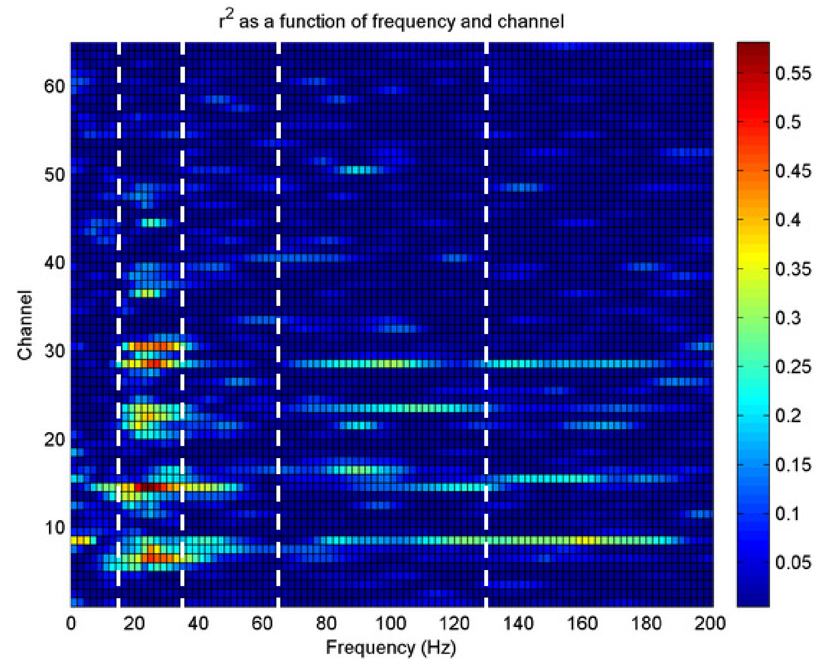
SIFT features

Extraction of power bands from EEG

1. Select time window
2. Fourier transform on each channel EEG to give corresponding channel power spectrum
3. Segment power spectrum into bands
4. Create channel-band feature by summing values in band



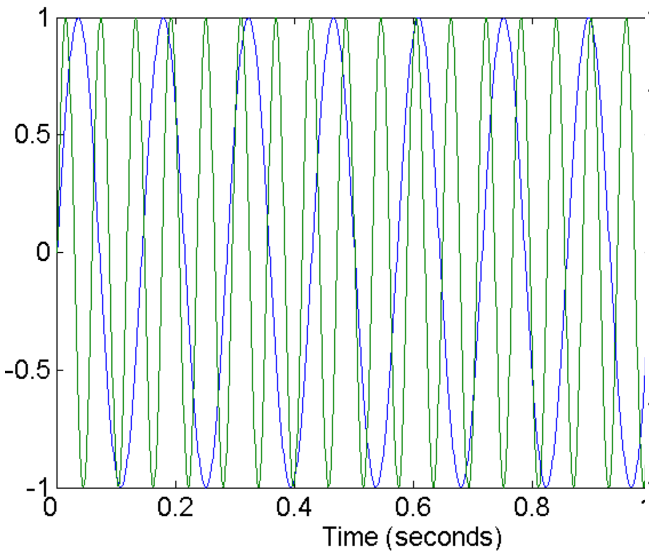
Multi-channel EEG recording
(time domain)



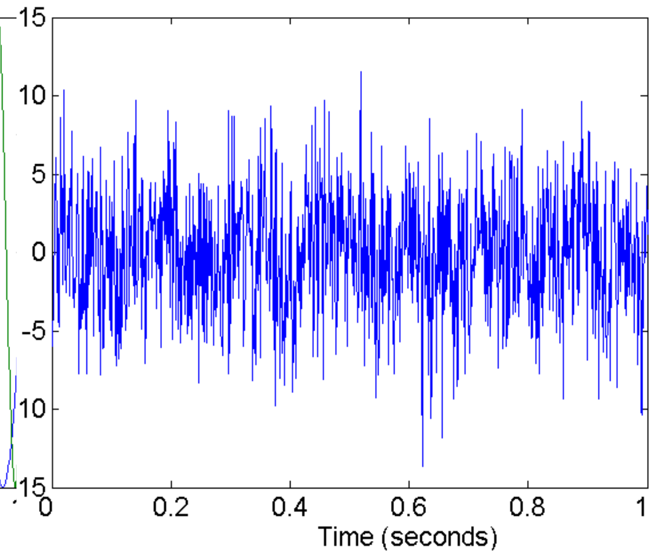
Multi-channel power spectrum
(frequency domain)

Map existing features to new space

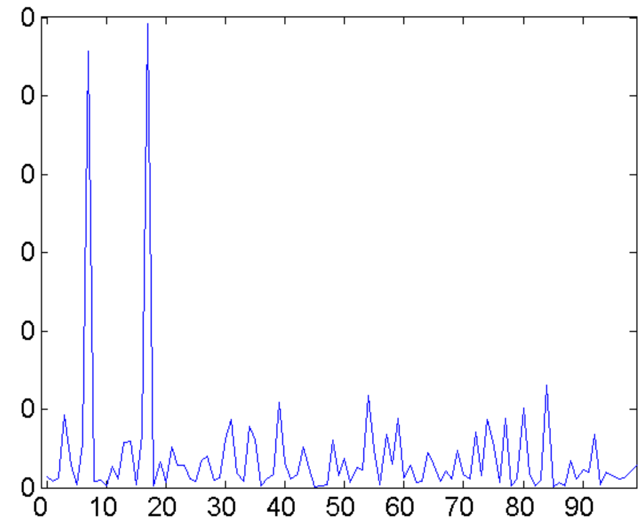
- Fourier transform
  - Eliminates noise present in time domain



Two sine waves



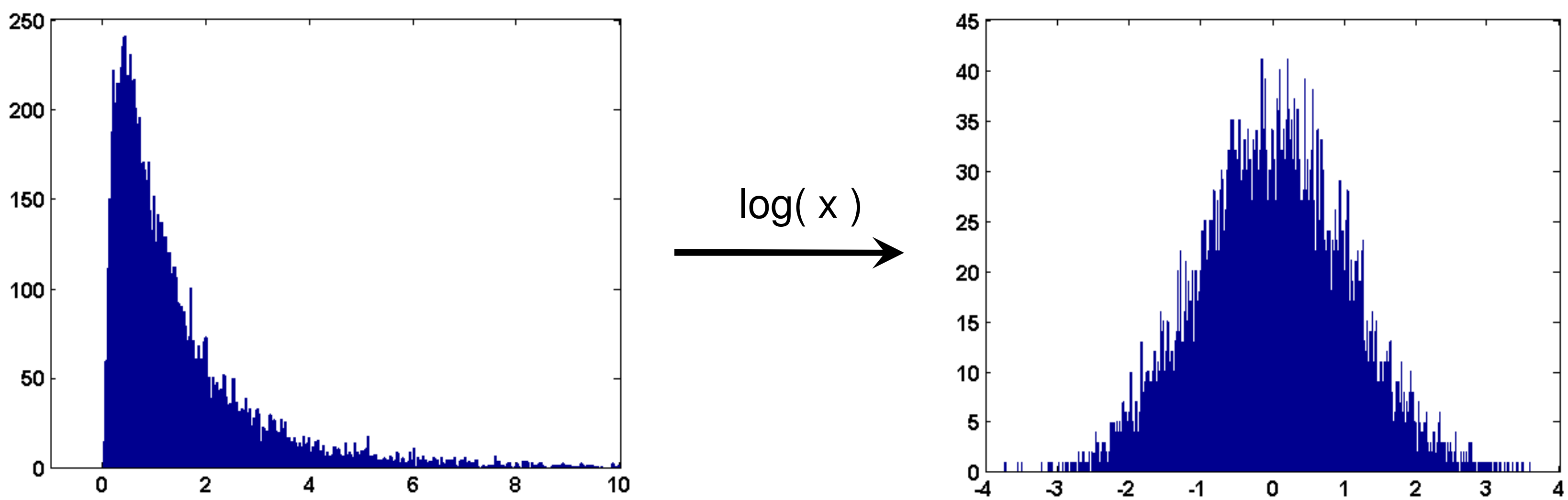
Two sine waves + noise



Frequency

Attribute transformation

- Simple functions
  – Examples of transform functions:
    x^k log(x) e^x | x |
  – Often used to make the data more like some standard distribution, to better satisfy assumptions of a particular algorithm.
    - Example: discriminant analysis explicitly models each class distribution as a multivariate Gaussian



Feature subset selection

- Reduces dimensionality of data without creating new features

- Motivations:
  - Redundant features
    - highly correlated features contain duplicate information
    - example: purchase price and sales tax paid
  - Irrelevant features
    - contain no information useful for discriminating outcome
    - example: student ID number does not predict student's GPA
  - Noisy features
    - signal-to-noise ratio too low to be useful for discriminating outcome
    - example: high random measurement error on an instrument

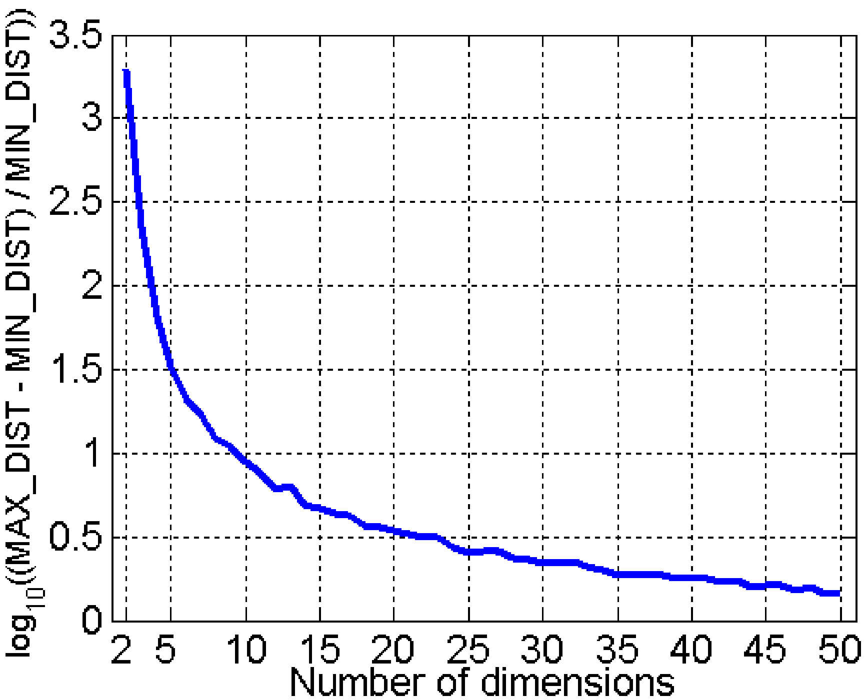
Feature subset selection

- Benefits:
  - Alleviate the curse of dimensionality
  - Enhance generalization
  - Speed up learning process
  - Improve model interpretability

Curse of dimensionality

- As number of features increases:
  - Volume of feature space increases exponentially.
  - Data becomes increasingly sparse in the space it occupies.
  - Sparsity makes it difficult to achieve statistical significance for many methods.
  - Definitions of density and distance (critical for clustering and other methods) become less useful.
    - all distances start to converge to a common value

Curse of dimensionality



- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

Approaches to feature subset selection

- Filter approaches:
  - Features selected before machine learning algorithm is run
- Wrapper approaches:
  - Use machine learning algorithm as black box to find best subset of features
- Embedded:
  - Feature selection occurs naturally as part of the machine learning algorithm
    - example: L1-regularized linear regression

Approaches to feature subset selection

- Both filter and wrapper approaches require:
  - A way to measure the predictive quality of the subset
  - A strategy for searching the possible subsets
    - exhaustive search usually infeasible – search space is the power set (2^d subsets)

Filter approaches

- Most common search strategy:
  1. Score each feature individually for its ability to discriminate outcome.
  2. Rank features by score.
  3. Select top *k* ranked features.
- Common scoring metrics for individual features
  - *t*-test or ANOVA (continuous features)
  - χ-square test (categorical features)
  - Gini index
  - etc.

Filter approaches

- Other strategies look at interaction among features
  - Eliminate based on correlation between pairs of features
  - Eliminate based on statistical significance of individual coefficients from a linear model fit to the data
    - example: t-statistics of individual coefficients from linear regression

Wrapper approaches

- Most common search strategies are *greedy*:
  - Random selection
  - Forward selection
  - Backward elimination
- Scoring uses some chosen machine learning algorithm
  - Each feature subset is scored by training the model using only that subset, then assessing accuracy in the usual way (e.g. cross-validation)

Forward selection

Assume d features available in dataset: $|\ FUnsel\ | == d$

Optional: target number of selected features k

Set of selected features initially empty: $FSel = \varnothing$

Best feature set score initially 0: $ScoreBest = 0$

Do

    Best next feature initially null: $FBest = \varnothing$

    For each feature $F \in FUnsel$

        Form a trial set of features $FTrial = FSel + F$

        Run wrapper algorithm, using only features *Ftrial*

        If $score(\ FTrial\ ) > scoreBest$

            $FBest = F; \quad scoreBest = score(\ FTrial\ )$

    If $FBest \neq \varnothing$

        $FSel = FSel + FBest; \quad FUnsel = FUnsel - FBest$

Until $FBest == \varnothing$ or $FUnsel == \varnothing$ or $|\ FSel\ | == k$

Return *FSel*

Random selection

Number of features available in dataset d

Target number of selected features k

Target number of random trials T

Set of selected features initially empty: $FSel = \varnothing$

Best feature set score initially 0: $ScoreBest = 0$.

Number of trials conducted initially 0: $t = 0$

Do

  Choose trial subset of features *FTrial* randomly from full set of d available features, such that $| FTrial | == k$

  Run wrapper algorithm, using only features *Ftrial*

  If $score(FTrial) > scoreBest$

    $FSel = FTrial; \quad scoreBest = score(FTrial)$

  $t = t + 1$

Until $t == T$

Return *FSel*

Other wrapper approaches

- If *d* and *k* not too large, can check all possible subsets of size *k*.
  - This is essentially the same as random selection, but done exhaustively.