Lecture 6: Assessment Methods

Isabelle Guyon
guyoni@inf.ethz.ch

Assessment Methods

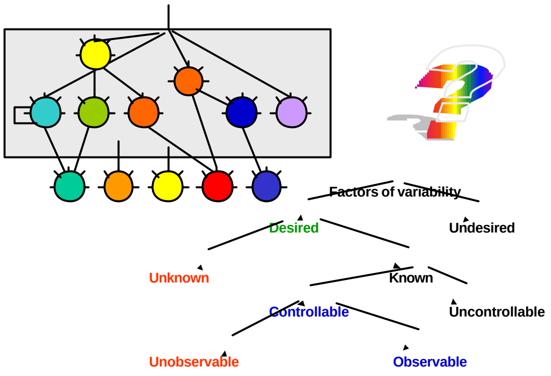
How good are the features / feature subsets we have selected?

- Classical statistics:
  - – Perform statistical tests.
- Machine learning:
  - – Use a training set and a validation set.

References: Book Chapter 2 + Book Appendix A +
What size test set gives good error rate estimates? *I. Guyon, et al.*
http://www.clopinet.com/isabelle/Papers/test-size.ps.Z

Part I: Review of previous lecture

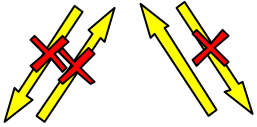
Reverse Engineering



Factors of variability
Desired
Undesired
Unknown
Known
Controllable
Uncontrollable
Unobservable
Observable

Relevance/Usefulness/Causality

Relevance (existence of a dependency)



Usefulness to the predictor
(prediction improvement)

Causality
(direct dependency as opposed to common cause; direction of dependency)

Definition of Relevance

- Surely irrelevant feature:

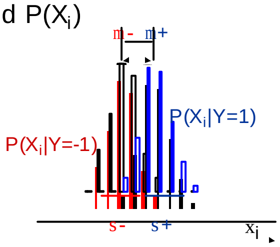
  $P(X_i, Y\ |\mathbf{X}^{-i}) = P(X_i\ |\mathbf{X}^{-i})P(Y\ |\mathbf{X}^{-i})$

  for all assignment of values to $\mathbf{X}^{-i}$
- Relevance:

  Find and index that measures the dissimilarity between $P(X_i, Y\ |\mathbf{X}^{-i})$ and $P(X_i\ |\mathbf{X}^{-i})P(Y\ |\mathbf{X}^{-i})$

Individual Relevance

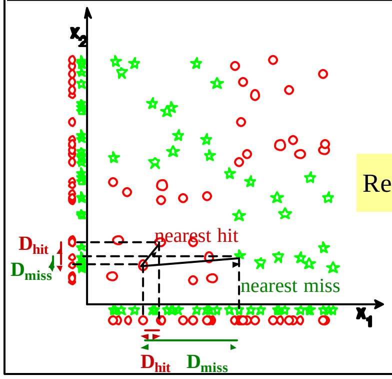
- We drop the conditioning on $\mathbf{X}^{-i}$
- Compare $P(X_i|Y)$ and $P(X_i)$
- Examples:
  - MI(X_i, Y)
  - Pearson(X_i, Y)
  - S2N(X_i, Y)
  - Fisher(X_i, Y)



Conditional Relevance

- We “simplify” the conditioning on $\mathbf{X}^{-i}$
- Examples:
  - Relief(X_i, Y)
  - CMI(X_i, Y)
  - GS(X_i, Y)

Relief



Relief=<D_{miss}/D_{hit}>

Forward Selection with GS

Stoppiglia, 2002. Gram-Schmidt orthogonalization.

- Select a first feature $X_{?(1)}$with maximum cosine with the target cos($\mathbf{x}_i$, $\mathbf{y}$)=$\mathbf{x}.\mathbf{y}$/||$\mathbf{x}$|| ||$\mathbf{y}$||
- For each remaining feature X_i
  - Project X_i and the target Y on the null space of the features already selected
  - Compute the cosine of X_i with the target in the projection
- Select the feature $X_{?(k)}$with maximum cosine with the target in the projection.

Sensitivity to outliers

- Ranking indices (e.g. correlation coefficients) may be sensitive to outliers.
- Idea: use the jackknife or bootstrap estimates!

Part II:
Assessment Methods:
Classical Statistics Viewpoint

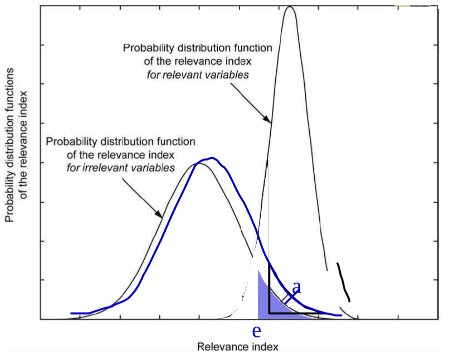
Filtering Distracters

- Assess the "statistical significance" of the relevance of given features.
- For a training set of size m, the ranking index is a random variable R.
- A feature is probably approximately irrelevant iff

Proba(R > ε) ≤ δ

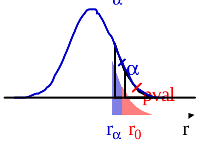
ε and δ are positive values. ε is the level of approximation. δ is the risk of being wrong.

Relevance Index Distribution

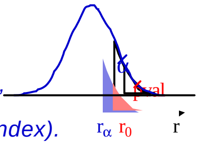


P-values

- Assume we know the distribution for **irrelevant features** Proba(R > ε).
- Select a risk α or a relevance threshold r_α such that α=Proba(R > r_α).
- Compute a realization r_0 of R for your training data.
- Compute the p-value: pval=Proba(R > r_0).
- Select the features with r_0 > r_α or pval ≤ α, because only a fraction α of irrelevant features have a relevance score larger than r_α.



Hypothesis Testing

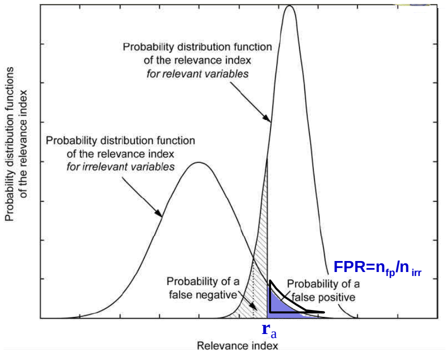


Ingredients:

- A "null hypothesis" H_0.
  "H_0: The feature is irrelevant"
- A test statistic R *(relevance index).*
- A distribution of R if H_0 is true *(null distribution)* Proba(R > ε).
- A risk value α and its corresponding threshold r_α, such that α=Proba(R > r_α).
- A realization r_0 of R from the training samples.

If r_0> r_α, reject H_0, with risk α of being wrong.

Relevance Index Distribution



Random Probes

- We may not know the distribution Proba(R > ε) for irrelevant features.
- But, we can create n_p "random probes", which are irrelevant features that look like the features in our data set, e.g. by randomizing the values of real features.
- We can compute the "relevance" of the probes.
- For a given relevance threshold r_α, the fraction of selected probes is the false positive rate:

$$FPR = n_{fp}/n_{irr} \cong n_{sp}/n_p$$

- This allows us to compute p-values.

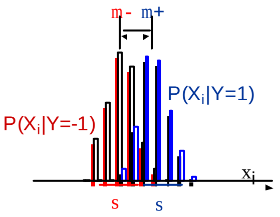
Random Probes are Cool!

- Random probes allow us to assess the significance of features for ANY ranking index. This includes:
  - "non-linear" indices like MI,
  - "context sensitive" indices like Relief,
  - "conditional relevance" indices involved in forward selection like GS and CMI.
- They are more general than "classical univariate tests".
- But, they add some computational burden and require that the probes be good representatives of truly irrelevant features.

Some Univariate Tests

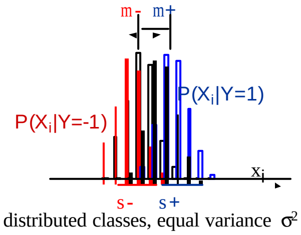
- Some relevance indices assume a simple model of $P(Y|X_i)$ and $P(X_i)$. Proba(R > ε) for irrelevant features is then known.
- Examples:
  - Z-test
  - T-test
  - ANOVA test

Z-test



- Normally distributed classes, equal variance σ^2 known, equal number of example per class.
- Null hypothesis H_0: $\mu+ = \mu-$
- Z statistic: $z= (\mu+ - \mu-)/(\sigma/\sqrt{m}) \sim N(0, 1)$, if H_0 is true.

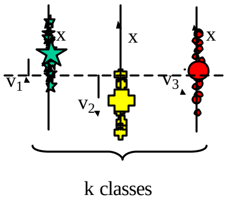
T-test



- Normally distributed classes, equal variance σ^2 unknown; estimated from data as σ^2_{within}.
- Null hypothesis H_0: $\mu+ = \mu-$
- T statistic: If H_0 is true,

$t= (\mu+ - \mu-)/(\sigma_{within}\sqrt{1/m^+ + 1/m^-}) \sim \text{Student}\,(m^+ + m^- - 2 \text{ d.f.})$

ANOVA test



$$F= \frac{\text{variance explained}}{\text{residual variance}}$$

$$F= \frac{\sigma_{between}}{\sigma_{within}} \sim \text{Snedecor}(k-1)(m-k)$$

H_0: All class means are equal.

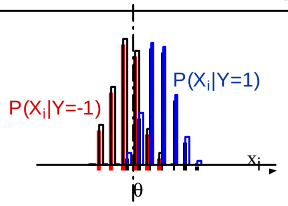
ANOVA model: $x_{ij} = \mu + v_j + \varepsilon_{ij}$

Reminder: model of the effect on observations x – the feature – of a **systematic** factor of variability – the multi-class target – $v \in \{v_1, v_2, \ldots v_j, \ldots\}$ and **intrinsic** variability ε (random error, normally distributed).

Non-parametric Tests

- Simple models of $P(Y|X_i)$ and $P(X_i)$ lead to parametric tests (e.g. T-test, ANOVA test).
- Non-parametric tests make no such assumptions, but compare distributions on the basis of low order statistics, e.g. the median. Proba($R > \varepsilon$) for irrelevant features is then also known.

AUC Ranking Index



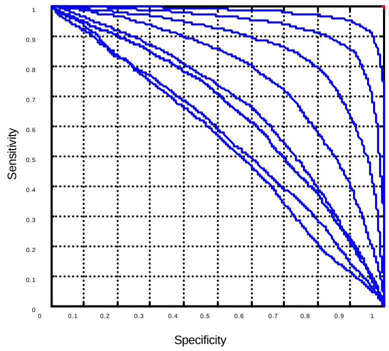
For each threshold value θ we have:

		Prediction	
		Class -1	Class +1
Truth	Class -1	tn	fp
	Class +1	fn	tp

- Sensitivity = error rate of the positive class = fn/(fn+ tp)
- Specificity = error rate of the negative class = fp/(fp+tn)

AUC=Area Under Curve sensitivity=f(specificity)

AUC Example



Wilcoxon –Mann-Whitney

- H0: The distribution of the 2 classes is the same.
- H1: There is a displacement.
- Rank sum statistic W: rank all the feature values; W=sum of the ranks of the values corresponding to the positive class.
- W is tabulated and approx. normal for n>7.
- The ranking obtained with the two-tailed pval(W) is the same as the one obtained with abs(AUC-0.5).

Multiple Testing

- If a **single** feature is tested, a threshold α on FPR=pval indicates the risk of making a wrong decision.
- In n features are tested simultaneously, will FPR indicate the fraction of incorrect decisions?
- No! If n independent tests are performed, the fraction of correct decisions will be $(1\text{-pval})^n$.
- Bonferroni correction: Replace pval by n pval or α by α/n.

Bonferroni Correction

- With Bonferroni correction:
  - pval' = n pval overestimated, or
  - risk threshold $\alpha' = \alpha/n$ underestimated.
- Without correction:
  - pval underestimated, or
  - α overestimated.

False Discovery Rate

$$FDR = n_{fp}/n_{sc}$$

fp=false positive=features falsely found relevant

sc= selected candidate features

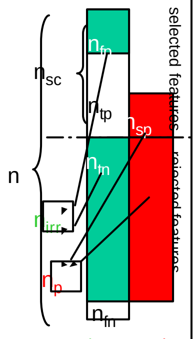
n_{fp} is unknown, but FPR can be calculated from pval or using the probe method. Bound the FDR:

$FPR = n_{fp}/n_{irr} \geq n_{fp}/n$ (irr=irrelevant feat.)

$FDR = (n_{fp}/n)\,(n/n_{sc}) \leq FPR\; n/n_{sc}$

$$FDR \leq FPR\; n/n_{sc} \leq \alpha$$

We obtain $FPR \leq \alpha\, n_{sc}/n$, intermediate between $FPR \leq \alpha$ and $FPR \leq \alpha/n$.



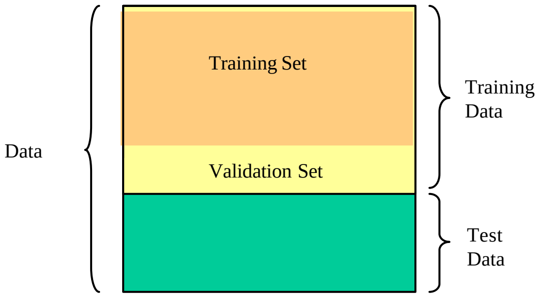
$FPR = n_{fp}/n_{irr} \cong n_{sp}/n_p$

Nested Feature Subsets

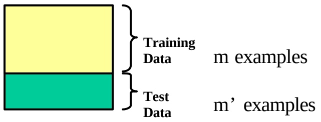
- Everything we talked about also works for forward selection methods providing nested subsets of features.
- Statistical tests may be used.
- The probe method may be used.
- See e.g. the GS example in the book.

Part III: Assessment Methods: Machine Learning Viewpoint

The problem(s)

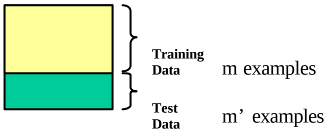


Variance of test error rate



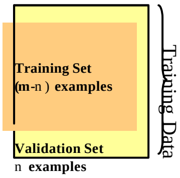
- i.i.d. errors.
- 2-class classification case: probability of error E, m' independent Bernouilli trials.
- The number of errors is distributed according to the Bionomial law of expected value m'E and variance m'E(1-E).
- The error rate (average number of errors) has variance **E(1-E)/m'**. [because var(aX)=a^2var(X)]

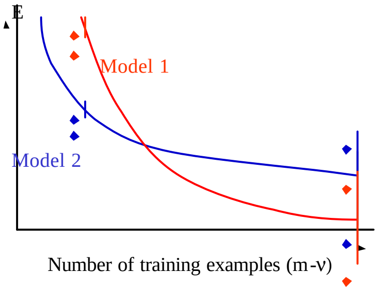
What size test set?



- Variance of test error rate σ^2= **E(1-E)/m'**.
  If E<<1, $\sigma^2 \cong$ E/m'. (1)
- Choose a given coefficient of variance σ/E=0.1, that is σ^2/E^2 = 0.01. (2)
- Combining (1) and (2):
  1/m'E=0.01
  m'=100/E

What size validation set?





- Single split.
- Variance of E: E(1-E)/ν
- Tradeoff bias/variance.

Test of significance

- What difference in error rate between 2 classifiers is statistically significant?
- McNemar paired test:

  - assume classifier 1 is better

  - ν_i=number of errors classifier i makes that the other classifier does not make.

  - if $E_2 - E_1 \geq (z_\alpha/\nu)\mathrm{sqrt}(\nu_1 + \nu_2)$ reject H_0 of equality of error rates with risk α.

  - one sided risk $\alpha=0.01$, $z_\alpha=2.33$.

Single Split

- Advantage: i.i.d errors. We can easily compute error bars and perform statistical tests.
- Disadvantage:
  - Small number of validation examples: large error bar.
  - Large number of validation examples, small number of training examples: large bias.

Cross-Validation

- Average over multiple splits
- Multiple splits with replacement (bootstrap)
- K-fold cross-validation
- Leave-one-out

Virtual LOO

- For some algorithms, it is possible to compute exactly (or approximately) the effect of removing one example on the loss function value of that example.
- Need to train only once!
- Examples:
  - Least square regression: exact formula.
  - Neural nets: approximate formula.
  - SVC: approximate formula.

Avoid biased CV!

- Wrong:
  - Rank the features with all the training set.
  - Use CV (e.g. virtual LOO) to select among subsets of variable size.
  - Cost: one training for each subset size.
- Correct:
  - Remove one example.
  - Rank the features.
  - Train on remaining examples and test on left out example for variable subset sizes.
  - Average the results for each subset size.
  - Cost: m training for each subset size.

Nested CV loops

- One should select both features and hyperparameters. Which should come first?
  - HP before feature selection
  - feature selection before HP
  - Both simultaneously
- Difficulty: both simultaneously is computationally expensive and requires a lot of data.

Variance of CV

- We average over multiple splits, but now we do not know the error bar exactly anymore (non i.i.d. errors).
- LOO has a lot of variance. Often 10-fold CV is a good choice.
- Stdev(CV-results): overestimate error bar; Stderr(CV-results): underestimate error.

Multiple Testing

- When we compare more than 2 classifiers, we perform multiple tests (explicitly or implicitly). Our risk of being wrong increases (remember Bonferroni correction).
- One should compare as few classifiers are possible:
  - Pre-rank the classifiers before your experiments
  - Of two classifiers performing similarly (within the error bar), prefer the classifier of lower rank.

Performance Prediction Challenge

Dataset	Size	Type	Features	Training Examples	Validation Examples	Test Examples
ADA	0.6 MB	Dense	48	4147	415	41471
GINA	19.4 MB	Dense	970	3153	315	31532
HIVA	7.6 MB	Dense	1617	3845	384	38449
NOVA	2.3 MB	Sparse binary	16969	1754	175	17537
SYLVA	15.6 MB	Dense	216	13086	1308	130858

http://www.modelselect.inf.ethz.ch/

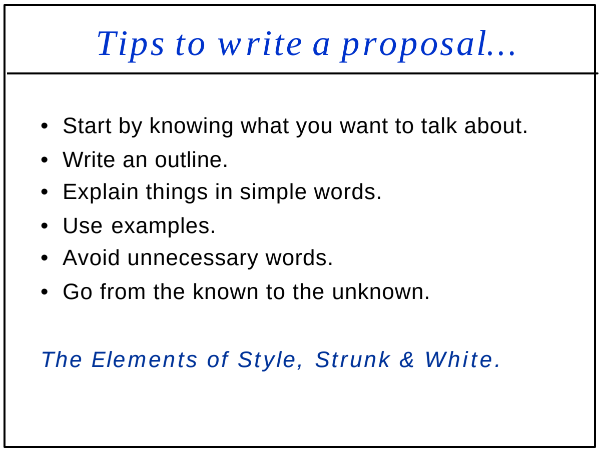
Conclusion

- No training data split:
  - – Use statistical tests or probe method to compute FPR=pval.
  - – Set threshold of significance on FDR $\cong$ FPRn/n_{sc}
- Training data split(s):
  - – One split: variance known $E(1-E)/\nu$ (but high), statistical tests can be performed.
  - – Cross-validation: variance less high but not exactly known, statistical tests less rigorous.
  - – Multiple comparisons: rank classifiers a priori.

Exercise Class

Homework 6

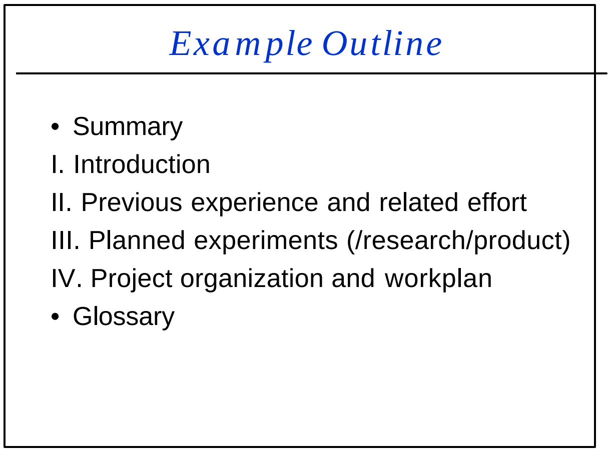
- Write a proposal for solving a problem involving feature extraction. Give special care to explaining assessment methods.
- Email the proposal to guyoni@inf.ethz.ch with subject "Homework6" no later than:
  Tuesday December 6th.



Tips to write a proposal...

- Start by knowing what you want to talk about.
- Write an outline.
- Explain things in simple words.
- Use examples.
- Avoid unnecessary words.
- Go from the known to the unknown.

The Elements of Style, Strunk & White.



Example Outline

- Summary

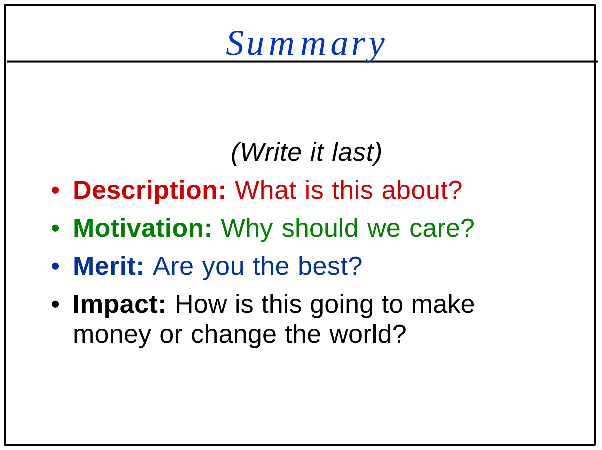
I. Introduction

II. Previous experience and related effort

III. Planned experiments (/research/product)

IV. Project organization and workplan

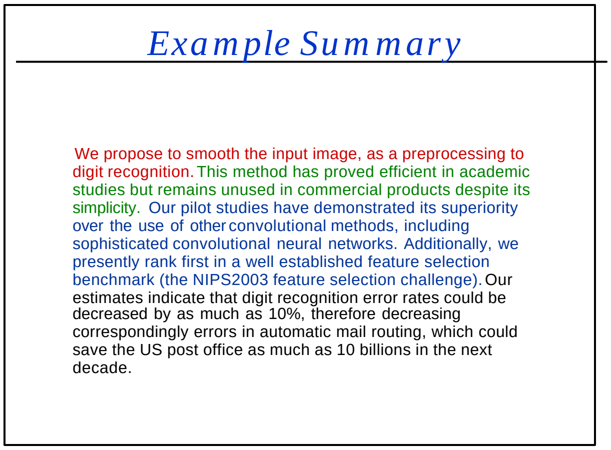
- Glossary



Summary

(Write it last)

- **Description:** What is this about?
- **Motivation:** Why should we care?
- **Merit:** Are you the best?
- **Impact:** How is this going to make money or change the world?



Example Summary

We propose to smooth the input image, as a preprocessing to digit recognition. This method has proved efficient in academic studies but remains unused in commercial products despite its simplicity. Our pilot studies have demonstrated its superiority over the use of other convolutional methods, including sophisticated convolutional neural networks. Additionally, we presently rank first in a well established feature selection benchmark (the NIPS2003 feature selection challenge). Our estimates indicate that digit recognition error rates could be decreased by as much as 10%, therefore decreasing correspondingly errors in automatic mail routing, which could save the US post office as much as 10 billions in the next decade.

Introduction

- What is the problem?
- Why is it a problem?
- How is it usually solved or why is it unsolved?
- Show you are in a good position to solve it: you have data, experience, technology.
- Give examples.
- Outline the proposed solution.

Introduction Example

We address the problem of handwritten digit recognition. This problem is central to the automatic reading of zip codes on envelopes. We focus on a subset of the problem that is particularly hard: the separation of digits “4” and “9”, which are confusable. Figure 1 shows an example of a confusable pair. To study this problem, we use the Gisette dataset of the NIPS 2003 challenge, which is a subset of the well known MNIST dataset. For this data, we have a number of baseline results from the challenge and from the literature. In the challenge, the best test results were around 1.3% balanced error rate (BER). For this problem, data representation is critical and the representation used by the challengers was naïve. We know from the academic literature that using domain knowledge is essential. Based on our preliminary experiments, we believe we can significantly improve handwriting recognition by performing good data preprocessing. Indeed, we have reached a performance of 0.91% BER with a simple smoothing of the images.

Previous experience and related efforts

- Explain your own previous experience and the way others have addressed the problem.
- By lumping both into one section, you can show you are an expert without boasting.
- For this exercise, summarize (briefly) what you learned in class.

Planned Experiments

This is the “meat” of your proposal:

- What are you going to do?
- Why do you take this approach?
- Are there alternatives? Will you eventually compare with them?
- How will you assess the results?

Example Planned Experiments

We want to study the effect of smoothing on the data. Our preferred method is to use a convolution of the image with an exponential kernel. In Figure 1, we show an example of digit before and after smoothing. Smoothing reduces the effect of small image distortions like translations, rotations, and skew. Therefore, combined with a polynomial support vector classifier (SVC), it usually improves performance. Feature selection also sometimes improve performance and we have observed in preliminary experiments that normalization also helps. Therefore, we plan to study the combination: {smoothing, feature selection, normalization, SVC}, and vary the hyper-parameters of each element of the chain. The implementation will be done in Matlab® using the spider package developed at the Max Planck Institute.

Other preprocessings are worth comparing to. We intend to compare with the preprocessings already implemented in the CLOP package provided for the feature extraction class.

For performance assessment, we will use the setup of the NIPS 2003 challenge. We are aware that comparing our results directly on the test set biases the results favorably. Hence we will provide only one final model to be tested with the test set. Hyperparameter settings, including the number of features, will be varied according to a factorial design and assessed with the balanced error rate (BER) in 10-fold cross-validation experiments.

Project Organization and Workplan

(Not necessary for this homework.)

- Split the project into tasks.
- Assign the tasks to collaborators.
- Create a timeline.
- Add “risk assessment and contingency plan”.