Robust Margin Based Classifiers For Small Sample Data

Sidharth Gupta

1Department of Computer Science
Arizona State University

April 20, 2011
Advisor

Dr Seungchan Kim

Members

Dr Bruno Welfert

Dr Baoxin Li

Welcome all
Committee

Advisor

Dr Seungchan Kim

Members

Dr Bruno Welfert

Dr Baoxin Li

Welcome all
1 Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2 Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3 Experimental results
   - Synthetic data-sets
   - Applications

4 Summary, my contribution, issues and future work
Outline

1 Robust Support Vector Machines
   - What is the problem
     - The Support Vector Machine
     - SVM and its limitations
     - A side note on cross validation
     - Previous work

2 Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3 Experimental results
   - Synthetic data-sets
   - Applications

4 Summary, my contribution, issues and future work
What is the problem?

- Biological experimental data has few samples, viz: cancer patients, test subjects etc.
  - Hard to gather data at will.
  - Cancer studies are fewer in comparison to other modes of data generation.

- Thus classification task for small sample data phenomenon needs addressing (Example, cancer vs normal). But many outliers and noisy samples in data causing a problem in classification.

- SVM’s are not robust to outliers:
  - Problem more pronounced in small sample setting.
  - SVM Loss functions only approximate 0-1 loss not actual nature of loss functions.
What is the problem?

- Biological experimental data has few samples, viz: cancer patients, test subjects etc.
  - Hard to gather data at will.
  - Cancer studies are fewer in comparison to other modes of data generation.

- Thus classification task for small sample data phenomenon needs addressing (Example, cancer vs normal). But many outliers and noisy samples in data causing a problem in classification.

- SVM’s are not robust to outliers:
  - Problem more pronounced in small sample setting.
  - SVM Loss functions only approximate 0-1 loss not actual nature of loss functions.
What is the problem?

- Biological experimental data has few samples, viz: cancer patients, test subjects etc.
  - Hard to gather data at will.
  - Cancer studies are fewer in comparison to other modes of data generation.

- Thus classification task for small sample data phenomenon needs addressing (Example, cancer vs normal). But many outliers and noisy samples in data causing a problem in classification.

- SVM’s are not robust to outliers:
  - Problem more pronounced in small sample setting.
  - SVM Loss functions only approximate 0-1 loss not actual nature of loss functions.
Outline

1. Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2. Mathematical formulation
   - New Loss function
   - RSVM formulation
   - Newton Method based formulation

3. Experimental results
   - Synthetic data-sets
   - Applications

4. Summary, my contribution, issues and future work
Maximum margin of the separating hyperplane from the data:

Figure: The geometric interpretation of the maximum margin.
Given a hypothesis space:

\[ \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq n}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{+1, -1\} \]

Choose a hyperplane, \(\mathbf{w} \cdot \mathbf{x}_i + b\), to maximize the margin of separation.

Maximize the distance \(\frac{2}{||\mathbf{w}||}\), between these two hyperplanes

\[ \mathbf{w} \cdot \mathbf{x}_i + b = 1 \]

and

\[ \mathbf{w} \cdot \mathbf{x}_i + b = -1 \]

Or minimize \(||\mathbf{w}||^2\)
Mathematical Formulation of the SVM
An overview

Given a hypothesis space:

\[ \{(x_i, y_i)\}_{1 \leq i \leq n}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{+1, -1\} \]

Choose a hyperplane, \( \langle w, b \rangle \), to maximize the margin of separation.

Maximize the distance \( \frac{2}{||w||} \), between these two hyperplanes

\[ w.x_i + b = 1 \]

and \[ w.x_i + b = -1 \]

Or minimize \( ||w||^2 \)
Given a hypothesis space:

\[ \{(x_i, y_i)\}_{1 \leq i \leq n}, \ x_i \in \mathbb{R}^d, \ y_i \in \{+1, -1\} \]

Choose a hyperplane, \( \langle w, b \rangle \), to maximize the margin of separation.

Maximize the distance \( \frac{2}{||w||} \), between these two hyperplanes

\[ w \cdot x_i + b = 1 \]

\[ \text{and} \quad w \cdot x_i + b = -1 \]

Or minimize \( ||w||^2 \)
Given a hypothesis space:

\[ \{(x_i, y_i)\}_{1 \leq i \leq n}, \ x_i \in \mathbb{R}^d, \ y_i \in \{+1, -1\} \]

Choose a hyperplane, \(< w, b >\), to maximize the margin of separation.

Maximize the distance \(2 \frac{||w||}{\|w\|}\), between these two hyperplanes

\[ w \cdot x_i + b = 1 \]

and  \[ w \cdot x_i + b = -1 \]

Or minimize \(||w||^2\)
The loss function
An overview

- Need to have a “soft margin” because of misclassified examples.

- Relax the constraints with “slack” variables

- Instead of \( y_i(w \cdot x_i + b) = 1 \) choose:
  \[
  y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
  \]

- This is the loss function. So our new optimization expression is:
  \[
  \min_{w,b} ||w||^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)
  \]
  under the constraints:
  \[
  y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
  \]
Need to have a “soft margin” because of misclassified examples.

Relax the constraints with “slack” variables

Instead of $y_i(w \cdot x_i + b) = 1$ choose:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

This is the loss function. So our new optimization expression is:

$$\min_{w,b} ||w||^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)$$

under the constraints:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$
The loss function
An overview

- Need to have a “soft margin” because of misclassified examples.

- Relax the constraints with “slack” variables

- Instead of $y_i(w \cdot x_i + b) = 1$ choose:
  
  $$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

- This is the loss function. So our new optimization expression is:

  $$\min_{w,b} ||w||^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)$$

  under the constraints:

  $$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$
- Need to have a “soft margin” because of misclassified examples.

- Relax the constraints with “slack” variables

- Instead of \( y_i(w \cdot x_i + b) = 1 \) choose:
  \[
y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
  \]

- This is the loss function. So our new optimization expression is:
  \[
  \min_{w, b} ||w||^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)
  
  under the constraints:
  \[
  y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
  \]
Features of SVM
An overview

- Huge improvement over classifiers like neural networks (first order methods) and linear classifiers.
- A guard against over-fitting by limiting the number of support vectors (by selecting the right hyperparamaters).
- Linearly inseparable data can be dealt with using Kernel Functions $\phi(x_i) \cdot \phi(x_j)$ instead of $x_i \cdot x_j$ in dual[8].

Figure: Kernelization: Projecting the data into a higher dimensional space to make it linearly separable.
Features of SVM
An overview

- Huge improvement over classifiers like neural networks (first order methods) and linear classifiers.

- A guard against over-fitting by limiting the number of support vectors (by selecting the right hyperparameters).

- Linearly inseparable data can be dealt with using Kernel Functions $\phi(x_i) \cdot \phi(x_j)$ instead of $x_i \cdot x_j$ in dual[8].

Figure: Kernalization: Projecting the data into a higher dimensional space to make it linearly separable.
Features of SVM
An overview

- Huge improvement over classifiers like neural networks (first order methods) and linear classifiers.

- A guard against over-fitting by limiting the number of support vectors (by selecting the right hyperparamaters).

- Linearly inseparable data can be dealt with using Kernel Functions $\phi(x_i)\phi(x_j)$ instead of $x_i.x_j$ in dual[8].

**Figure:** Kernelization: Projecting the data into a higher dimensional space to make it linearly separable.
Outline

1. Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2. Mathematical formulation
   - New Loss function
   - RSVM formulation
   - Newton Method based formulation

3. Experimental results
   - Synthetic data-sets
   - Applications

4. Summary, my contribution, issues and future work
SVM and its limitation

- Loss function. Simple vs Natural loss.
- Over fitting of non-linear SVM.
  - Over-fitting discovered in case of non-linear SVM’s and application of kernel methods[7].
  - Need to do CV for hyperparameter selection to guard against over fitting.
  - But CV cant be done reliably with small sample data-sets (discussed later)[1].
- Outlier robustness issue with SVM’s
SVM and its limitation

- Loss function. Simple vs Natural loss.

- Over fitting of non-linear SVM.
  - Over-fitting discovered in case of non-linear SVM’s and application of kernel methods[7].
  - Need to do CV for hyperparameter selection to guard against over fitting.
  - But CV cant be done reliably with small sample data-sets (discussed later)[1].

- Outlier robustness issue with SVM’s
SVM and its limitation

- Loss function. Simple vs Natural loss.

- Over fitting of non-linear SVM.
  - Over-fitting discovered in case of non-linear SVM’s and application of kernel methods[7].
  - Need to do CV for hyperparameter selection to guard against over fitting.
  - But CV cant be done reliably with small sample data-sets (discussed later)[1].

- Outlier robustness issue with SVM’s
Loss Function(s) used by SVM

Their features

- **Hinge Loss**

\[
\sum_{i=1}^{n} \max(0, 1 - y_i (w \cdot x_i))^p
\]

- **Features**
  - Analytically easy to treat. Convex and can be made continuous.
  - \( p=1 \) yields a Sparse SVM increasing Bias and lowering variance. \( p=2 \) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.
  - Other loss functions yield less sparse solutions or affected by probability space of data[5]
Loss Function(s) used by SVM

Their features

- Hinge Loss

\[ \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i))^p \]

- Features
  - Analytically easy to treat. Convex and can be made continuous.
  - \( p=1 \) yields a Sparse SVM increasing Bias and lowering variance. \( p=2 \) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.
  - Other loss functions yield less sparse solutions or affected by probability space of data[5]
Loss Function(s) used by SVM

Their features

- Hinge Loss

\[ \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i))^p \]

- Features
  - Analytically easy to treat. Convex and can be made continuous.
  - \( p=1 \) yields a Sparse SVM increasing Bias and lowering variance. \( p=2 \) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.
  - Other loss functions yield less sparse solutions or affected by probability space of data[5]
Loss Function(s) used by SVM

Their features

- **Hinge Loss**

\[
\sum_{i=1}^{n} \max(0, 1 - y_i(w.x_i))^p
\]

- **Features**
  - Analytically easy to treat. Convex and can be made continuous.
  - \(p=1\) yields a Sparse SVM increasing Bias and lowering variance. \(p=2\) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.

- Other loss functions yield less sparse solutions or affected by probability space of data[5]
Loss Function(s) used by SVM

Their features

- **Hinge Loss**

\[ \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i))^p \]

- **Features**
  - Analytically easy to treat. Convex and can be made continuous.
  - \( p=1 \) yields a Sparse SVM increasing Bias and lowering variance. \( p=2 \) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.

- Other loss functions yield less sparse solutions or affected by probability space of data\(^5\)
Loss Function(s) used by SVM

Their features

- Hinge Loss

\[ \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i))^p \]

- Features
  - Analytically easy to treat. Convex and can be made continuous.
  - \( p=1 \) yields a Sparse SVM increasing Bias and lowering variance. \( p=2 \) is less sparse but yields scalable solutions.
  - Can be kernelized easily.
  - Gives good results with most real life data-sets when data samples are enough.

- Other loss functions yield less sparse solutions or affected by probability space of data[5]
Loss Function(s) used by SVM
...and their limitations

- Hinge loss is not a “proper scoring” function (no probability estimates)[5].
- Does not estimate a class probability but just a real valued loss.
  - Used because it gets the sign right to estimate the 0-1 loss.
  - Argument is because it is analytically simple to deal with no need to try more difficult loss functions.
- Most importantly, outliers have maximal bearing on the hinge loss (discussed next).
Loss Function(s) used by SVM
…and their limitations

- Hinge loss is not a “proper scoring” function (no probability estimates)[5].
- Does not estimate a class probability but just a real valued loss.
  - Used because it gets the sign right to estimate the 0-1 loss.
  - Argument is because it is analytically simple to deal with no need to try more difficult loss functions.
- Most importantly, outliers have maximal bearing on the hinge loss (discussed next).
Loss Function(s) used by SVM
...and their limitations

- Hinge loss is not a “proper scoring” function (no probability estimates) [5].
- Does not estimate a class probability but just a real valued loss.
  - Used because it gets the sign right to estimate the 0-1 loss.
  - Argument is because it is analytically simple to deal with no need to try more difficult loss functions.
- Most importantly, outliers have maximal bearing on the hinge loss (discussed next).
Loss Function(s) used by SVM
...and their limitations

- Hinge loss is not a “proper scoring” function (no probability estimates) [5].

- Does not estimate a class probability but just a real valued loss.
  - Used because it gets the sign right to estimate the 0-1 loss.
  - Argument is because it is analytically simple to deal with no need to try more difficult loss functions.

- Most importantly, outliers have maximal bearing on the hinge loss (discussed next).
Outlier robustness

- Has been discussed in many studies on discriminative classification [10].

- SVM’s have issue with outliers, especially in small sample setting. This type of setting is not studied particularly well in earlier studies.
Outlier robustness

- Has been discussed in many studies on discriminative classification[10].

- SVM’s have issue with outliers, especially in small sample setting. This type of setting is not studied particularly well in earlier studies.
Figure: The Robust Margin classifier shown to be more robust to outliers than the SVM.
Outline

1 Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2 Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3 Experimental results
   - Synthetic data-sets
   - Applications

4 Summary, my contribution, issues and future work
To Cross Validate or not?
That is the question

- Classifiers must minimize expected loss on future examples (Expected Prediction error).
- But cross-validation has no unbiased estimators for variance of the error.[1].
- Gets worst on small samples as IID assumption fails.
  - We can’t find pristine examples[5].
  - Error bounds don’t hold even for distribution free bounds[4].
- Need a better error estimation technique. Another way suggested in this study.
To Cross Validate or not?
That is the question

- Classifiers must minimize expected loss on future examples (Expected Prediction error).

- But cross-validation has no unbiased estimators for variance of the error.[1].
  - Gets worst on small samples as IID assumption fails.
    - We can’t find pristine examples[5].
    - Error bounds don’t hold even for distribution free bounds[4].

- Need a better error estimation technique. Another way suggested in this study.
To Cross Validate or not?
That is the question

- Classifiers must minimize expected loss on future examples (Expected Prediction error).
- But cross-validation has no unbiased estimators for variance of the error.[1].
- Gets worst on small samples as IID assumption fails.
  - We can’t find pristine examples[5].
  - Error bounds don’t hold even for distribution free bounds[4].

- Need a better error estimation technique. Another way suggested in this study.
To Cross Validate or not?
That is the question

- Classifiers must minimize expected loss on future examples (Expected Prediction error).
- But cross-validation has no unbiased estimators for variance of the error.[1].
- Gets worst on small samples as IID assumption fails.
  - We can’t find pristine examples[5].
  - Error bounds don’t hold even for distribution free bounds[4].
- Need a better error estimation technique. Another way suggested in this study.
Outline

1 Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2 Mathematical formulation
   - New Loss function
   - RSVM formulation
   - Newton Method based formulation

3 Experimental results
   - Synthetic data-sets
   - Applications

4 Summary, my contribution, issues and future work
Previous work

- Kim et al: Addresses issue with LDA classifier in small sample setting.

- Xu 2006: Addresses issue of outliers in SVM but not in small sample setting.

- Raudy and Jain 1991: Recommendations for small sample size effects
  - Investigate bias and variance of error estimates
  - Attention to feature selection
  - Presence of outliers is important
  - Number of training and test samples required

*But SVM performance in small sample data setting with outliers still not addressed*
Previous work

- Kim et al: Addresses issue with LDA classifier in small sample setting.

- Xu 2006: Addresses issue of outliers in SVM but not in small sample setting.

- Raudy and Jain 1991: Recommendations for small sample size effects
  - Investigate bias and variance of error estimates
  - Attention to feature selection
  - Presence of outliers is important
  - Number of training and test samples required

But SVM performance in small sample data setting with outliers still not addressed
Previous work

- Kim et al: Addresses issue with LDA classifier in small sample setting.
- Xu 2006: Addresses issue of outliers in SVM but not in small sample setting.
- Raudy and Jain 1991: Recommendations for small sample size effects
  - Investigate bias and variance of error estimates
  - Attention to feature selection
  - Presence of outliers is important
  - Number of training and test samples required

But SVM performance in small sample data setting with outliers still not addressed
Previous work

- Kim et al: Addresses issue with LDA classifier in small sample setting.
- Xu 2006: Addresses issue of outliers in SVM but not in small sample setting.
- Raudy and Jain 1991: Recommendations for small sample size effects
  - Investigate bias and variance of error estimates
  - Attention to feature selection
  - Presence of outliers is important
  - Number of training and test samples required

But SVM performance in small sample data setting with outliers still not addressed
Outline

1 Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2 Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3 Experimental results
   - Synthetic data-sets
   - Applications

4 Summary, my contribution, issues and future work
Gaussian loss

\[ L_g(w, x_i) = \begin{cases} 
  \frac{\text{erfc}(\frac{d_i}{\sqrt{2}\sigma_{ck}})}{2} & 0 < y_i(w.x_i) \leq 1 \\
  \frac{1}{2} + \frac{\text{erf}(\frac{d_i}{\sqrt{2}\sigma_{ck}})}{2} & y_i(w.x_i) < 0 
\end{cases} \]

- Loss function is defined as “sample spread”. Motivated from earlier study[4].
- Proven convexity

- Loss function parameters \( \sigma_{ck} \) are estimated from data. \( d_i \) is sample distance from hyper plane. \( d_i = \frac{|w.x_i|}{\|w\|\sqrt{2}\sigma_{ck}} \).
Robust Support Vector Machines

Mathematical formulation

Experimental results

Summary, my contribution, issues and future work

New Loss function

RSVM formulation.

Newton Method based formulation

Gaussian loss

$L_g(w, x_i) = \begin{cases} 
\frac{\text{erfc}(\frac{d_i}{\sqrt{2}\sigma_{ck}})}{2} & 0 < y_i(w.x_i) \leq 1 \\
\frac{1}{2} + \frac{\text{erf}(\frac{d_i}{\sqrt{2}\sigma_{ck}})}{2} & y_i(w.x_i) < 0 
\end{cases}$

- Loss function is defined as “sample spread”. Motivated from earlier study[4].
- Proven convexity
- Loss function parameters $\sigma_{ck}$ are estimated from data. $d_i$ is sample distance from hyper plane. $d_i = \frac{|w.x_i|}{||w||\sqrt{2}\sigma_{ck}}$. 
Figure: An intuitive view of the loss function. The SVM (right) only tends to penalize the margin based on the support vectors near the margin, but the RSVM (left) takes the sample spreading across the margin into account while calculating the penalization. The area of the spread across the hyperplane is the loss function and is the integral of the Gaussian or the error function. The spreading variance is obtained from the data itself and is not a hyperparameter.
Robust Support Vector Machines

1. Mathematical formulation
   - New Loss function
   - RSVM formulation
   - Newton Method based formulation

2. Experimental results
   - Synthetic data-sets
   - Applications

3. Summary, my contribution, issues and future work

Outlier robustness in classifiers

Sidharth Gupta
Robust SVM optimizes the same margin as the SVM (but a different loss function):

$$\lambda \|w\|^2 + \sum_{i \in n_{sv}} L_g(w, x_i, b)$$

- Solved by Primal Optimization
  - Primal and the dual recover the same solution.[2]
  - Uses newton raphson to compute optimal solution.

- Analytical form of Hessian completed.
- Newton updates: \( w = w + \eta H^{-1} \nabla \)
Robust-SVM (R-SVM)

- Robust SVM optimizes the same margin as the SVM (but a different loss function):
  \[ \lambda \|w\|^2 + \sum_{i \in n_{sv}} L_g(w, x_i, b) \]

- Solved by Primal Optimization
  - Primal and the dual recover the same solution. [2]
  - Uses Newton-Raphson to compute optimal solution.

- Analytical form of Hessian completed.
- Newton updates: \[ w = w + \eta H^{-1} \nabla \]
Robust-SVM (R-SVM)

- Robust SVM optimizes the same margin as the SVM (but a different loss function):
  \[ \lambda \|w\|^2 + \sum_{i \in n_{sv}} L_g(w, x_i, b) \]

- Solved by Primal Optimization
  - Primal and the dual recover the same solution.[2]
  - Uses Newton Raphson to compute optimal solution.

- Analytical form of Hessian completed.
- Newton updates: \[ w = w + \eta H^{-1} \nabla \]
Outline

1. Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2. Mathematical formulation
   - New Loss function
   - RSVM formulation
   - Newton Method based formulation

3. Experimental results
   - Synthetic data-sets
   - Applications

4. Summary, my contribution, issues and future work

Sidharth Gupta
Outlier robustness in classifiers
How it works
Second Order solver: When you have a hammer everything else is a nail.

- First Derivative
  \[ \nabla = \begin{cases} 
    2 \cdot w - \frac{1}{\sqrt{2\pi\sigma^2_c}} \sum_{i \in n_{sv}} f \cdot \frac{x_i}{|w|} - \frac{f}{|w|} \cdot g \cdot \frac{w}{|w|} & 0 < y_i(w \cdot x_i + b) \leq 1 \\
    2 \cdot w + \frac{1}{\sqrt{2\pi\sigma^2_c}} \sum_{i \in n_{sv}} f \cdot \frac{x_i}{|w|} - \frac{f}{|w|} \cdot g \cdot \frac{w}{|w|} & y(w \cdot x_i + b) < 0 
  \end{cases} \]

- Use L-BFGS to get second derivative
  - Known to have a duality gap of zero with dual[2]
  - Use of Quasi Newton methods for second order derivative as Hessian is non PSD.

- Analytical form of Hessian was completed but was not Positive Semi Definite(PSD).
- Bias term included by expanding hyperplane: \( w \) and data matrix: \( X \) by 1 dimension each[3].
Robust Support Vector Machines

Mathematical formulation

Exp erimental results

Summary, my contribution, issues and future work

New Loss function

RSVM formulation.

Newton Method based formulation

How it works

Second Order solver: When you have a hammer everything else is a nail.

**First Derivative**

\[
\nabla = \begin{cases} 
2 \cdot w - \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f \cdot \frac{x_i}{\|w\|} - \frac{f}{\|w\|} \cdot g \cdot \frac{w}{\|w\|} & 0 < y_i(w \cdot x_i + b) \leq 1 \\
2 \cdot w + \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f \cdot \frac{x_i}{\|w\|} - \frac{f}{\|w\|} \cdot g \cdot \frac{w}{\|w\|} & y(w \cdot x_i + b) < 0 
\end{cases}
\]

**Use L-BFGS to get second derivative**

- Known to have a duality gap of zero with dual[2]
- Use of Quasi Newton methods for second order derivative as Hessian is non PSD.

- Analytical form of Hessian was completed but was not Positive Semi Definite(PSD).
- Bias term included by expanding hyperplane:w and data matrix:X by 1 dimension each[3].
How it works

Second Order solver: When you have a hammer everything else is a nail.

- **First Derivative**

\[
\nabla = \begin{cases} 
2 \cdot w - \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i\in nsv} f \cdot \frac{x_i}{||w||} - \frac{f}{||w||} \cdot g \cdot \frac{w}{||w||} & 0 < y_i(w \cdot x_i + b) \leq 1 \\
2 \cdot w + \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i\in nsv} f \cdot \frac{x_i}{||w||} - \frac{f}{||w||} \cdot g \cdot \frac{w}{||w||} & y(w \cdot x_i + b) < 0
\end{cases}
\]

- Use L-BFGS to get second derivative
  - Known to have a duality gap of zero with dual[2]
  - Use of Quasi Newton methods for second order derivative as Hessian is non PSD.

- Analytical form of Hessian was completed but was not Positive Semi Definite(PSD).

- Bias term included by expanding hyperplane:$w$ and data matrix:$X$ by 1 dimension each[3].
The formulation of the Hessian was important part of the thesis.

Particularly the derivatives of Vector Valued Functions

$$\exp(-d_i^2(w, b, x_i)) \ast \frac{x_i}{||w||} \text{ or } f_i \ast \frac{x_i}{||w||}$$

$$\frac{\partial(f_i \ast x_i)}{\partial w} = \begin{bmatrix}
  x_i^1 \frac{\partial(\frac{f_i}{||w||})}{\partial w} \\
  x_i^2 \frac{\partial(\frac{f_i}{||w||})}{\partial w} \\
  \vdots \\
  x_i^D \frac{\partial(\frac{f_i}{||w||})}{\partial w}
\end{bmatrix}$$
The formulation of the Hessian was important part of the thesis.

Particularly the derivatives of Vector Valued Functions

\[ \exp \left( -d_i^2(w, b, x_i) \right) \cdot \frac{x_i}{||w||} \text{ or } f_i \cdot \frac{x_i}{||w||} \]

\[
\frac{\partial (f_i \ast x_i)}{\partial w} = \begin{bmatrix}
    x_i^1 \frac{\partial (f_i)}{\partial w} \\
    x_i^2 \frac{\partial (f_i)}{\partial w} \\
    \vdots \\
    x_i^D \frac{\partial (f_i)}{\partial w}
\end{bmatrix}
\]
Highlights of formulation

- Derivative of the vector valued function \( \mathbf{w} \ast \exp \left(-d_i^2(\mathbf{w}, b, \mathbf{x}_i)\right) \frac{(\mathbf{w} \cdot \mathbf{x}_i + b)}{||\mathbf{w}||^3} \) or \( ||\mathbf{w}|| * (h_i \ast g_i)\):

\[
\frac{\partial}{\partial \mathbf{w}} \left( \frac{\mathbf{w} \otimes \mathbf{w}}{||\mathbf{w}||} \ast h_i \ast g_i + \frac{\mathbf{w} \otimes (h'_i g_i + g'_i h_i)}{||\mathbf{w}||} \right) \ast \left[ \begin{array}{ccc} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{array} \right]_{D \times D}
\]

\[
+ \left( \left( \frac{1}{||\mathbf{w}||} - \frac{\mathbf{w} \cdot \mathbf{w}}{||\mathbf{w}||^3} \right) \ast h_i \ast g_i \right) \left[ \begin{array}{ccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{array} \right]_{D \times D}
\]

- Analytical existence of the loss function opens a door to researching alternate and perhaps better functions.
Derivative of the vector valued function
\[ \mathbf{w} \ast \exp \left( -d_i^2(\mathbf{w}, b, \mathbf{x}_i) \right) \frac{(\mathbf{w} \cdot \mathbf{x}_i + b)}{||\mathbf{w}||^3} \text{ or } \frac{\mathbf{w}}{||\mathbf{w}||} \ast (h_i \ast g_i): \]

\[ \frac{\partial (\frac{\mathbf{w}}{||\mathbf{w}||} \ast (h_i \ast g_i))}{\partial \mathbf{w}} = \begin{bmatrix}
\mathbf{w} \otimes \mathbf{w} / ||\mathbf{w}|| \ast h_i \ast g_i + \mathbf{w} \otimes (h'_i g_i + g'_i h_i) / ||\mathbf{w}|| \\
\end{bmatrix} \begin{bmatrix}
0 & 1 & \ldots & 1 \\
1 & 0 & \ldots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \ldots & 0
\end{bmatrix}_{D \times D}
\]

\[ + \left( \left( \frac{1}{||\mathbf{w}||} - \frac{\mathbf{w} \cdot \mathbf{w}}{||\mathbf{w}||^3} \right) \ast h_i \ast g_i \right) \begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 1
\end{bmatrix}_{D \times D} \]

Analytical existence of the loss function opens a door to researching alternate and perhaps better functions.
Outline

1. Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2. Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3. Experimental results
   - Synthetic data-sets
   - Applications

4. Summary, my contribution, issues and future work
Experimental procedure

- Generate a data set from a known distribution. The parameters are:
  - Number of points $N = [10, 30, 50]$.
  - Variance of data $\sigma_c = [0.5, 0.6, 0.7, 0.8, 0.9, 1]$.
  - Variance of noise $\sigma_N = [0.5, 0.2, 0.1, 0.05]$.
  - Samples $s = 200$.

- Train the classifiers on the noisy data.

- Estimate the error of the converged classifier against the true noise-free distribution (Bayes error).

- Calculate error confidence measures and a hypothesis testing measure to unambiguously determine the error margins.

- Better classifier should have minimal error against the true, noise-free distribution even when trained on noisy data (*central idea of thesis*).
Experimental procedure

- Generate a data set from a known distribution. The parameters are:
  - Number of points $N = [10, 30, 50]$
  - Variance of data $\sigma_c = [0.5, 0.6, 0.7, 0.8, 0.9, 1]$
  - Variance of noise $\sigma_N = [0.5, 0.2, 0.1, 0.05]$
  - Samples $s = 200$

- Train the classifiers on the noisy data.

- Estimate the error of the converged classifier against the true noise free distribution (Bayes error).

- Calculate error confidence measures and a hypothesis testing measure to unambiguously determine the error margins.

- Better classifier should have minimal error against the true, noise free, distribution even when trained on noisy data (central idea of thesis).
Experimental procedure

- Generate a data set from a known distribution. The parameters are:
  - Number of points \( \mathbb{N} = [10, 30, 50] \)
  - Variance of data \( \sigma_c = [0.5, 0.6, 0.7, 0.8, 0.9, 1] \)
  - Variance of noise \( \sigma_N = [0.5, 0.2, 0.1, 0.05] \)
  - Samples \( s = 200 \)

- Train the classifiers on the noisy data.

- Estimate the error of the converged classifier against the true noise free distribution (Bayes error).

- Calculate error confidence measures and a hypothesis testing measure to unambiguously determine the error margins.

- Better classifier should have minimal error against the true, noise free, distribution even when trained on noisy data (central idea of thesis).
Experimental procedure

- Generate a data set from a known distribution. The parameters are:
  - Number of points $N = [10, 30, 50]$
  - Variance of data $\sigma_c = [0.5, 0.6, 0.7, 0.8, 0.9, 1]$
  - Variance of noise $\sigma_N = [0.5, 0.2, 0.1, 0.05]$
  - Samples $s = 200$

- Train the classifiers on the noisy data.

- Estimate the error of the converged classifier against the true noise free distribution (Bayes error).

- Calculate error confidence measures and a hypothesis testing measure to unambiguously determine the error margins.

- Better classifier should have minimal error against the true, noise free, distribution even when trained on noisy data (central idea of thesis).
Experimental procedure

- Generate a data set from a known distribution. The parameters are:
  - Number of points $N = [10, 30, 50]$
  - Variance of data $\sigma_c = [0.5, 0.6, 0.7, 0.8, 0.9, 1]$
  - Variance of noise $\sigma_N = [0.5, 0.2, 0.1, 0.05]$
  - Samples $s = 200$

- Train the classifiers on the noisy data.

- Estimate the error of the converged classifier against the true noise free distribution (Bayes error).

- Calculate error confidence measures and a hypothesis testing measure to unambiguously determine the error margins.

- Better classifier should have minimal error against the true, noise free, distribution even when trained on noisy data (*central idea of thesis*).
Figure: Generated data snapshot: fixed variance and mean (column 1). Noise added to represent outliers (column 2). Variance of the data, here fixed at 0.5. The sample size, here fixed at 50 per class. $N \times |\sigma_c|$ data-sets like this are generated to test classifiers.
Results on synthetic data-set

**Figure:** Left to right: Three sample sizes of 10, 30, 50 samples. X-axis: Variance of data (grouped by variance of noise), Y-Axis: error of the classifier with 95% confidence interval. 4 classifiers RSVM, SVM, Sigma, Bayes classifier.
Results on synthetic data-set

A side note on error measurement

- Error against the distribution is measured by using the closed form ($d_\mu$ is distance of hyperplane from mean):

  $$\int_{d_{\mu1}}^{\infty} \exp(-x^2)dx + \int_{d_{\mu2}}^{\infty} \exp(-x^2)dx$$

![Graph showing minimal error by the Bayes classifier and a suboptimal classifier](image)

**Figure:** The minimal error (left) by the Bayes classifier and that by a suboptimal classifier (right)
Results on synthetic data-set

A side note on error measurement

- Error against the distribution is measured by using the closed form \((d_\mu \text{ is distance of hyperplane from mean)}:\)

\[
\int_{d_{\mu_1}}^{\infty} \exp(-x^2) dx + \int_{d_{\mu_2}}^{\infty} \exp(-x^2) dx
\]

**Figure:** The minimal error (left) by the Bayes classifier and that by a suboptimal classifier (right)
Outline

1. Robust Support Vector Machines
   - What is the problem
   - The Support Vector Machine
   - SVM and its limitations
   - A side note on cross validation
   - Previous work

2. Mathematical formulation
   - New Loss function
   - RSVM formulation.
   - Newton Method based formulation

3. Experimental results
   - Synthetic data-sets
   - Applications

4. Summary, my contribution, issues and future work
The Glioblastoma multiforme (GBM) data-set

- Most aggressive type of primary brain tumor in humans, involving glial cells.
- 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors.
- They are the most prevalent form of primary brain tumors according to a WHO study[6].
- The Glioblastoma multiforme (GBM) data set has 173 samples.
- 28 genes identified in the paper [9] as critical to distinguishing the classes apart.
The Glioblastoma multiforme (GBM) data-set

- Most aggressive type of primary brain tumor in humans, involving glial cells.
- 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors.
- They are the most prevalent form of primary brain tumors according to a WHO study[6].
- The Glioblastoma multiforme (GBM) data set has 173 samples.
- 28 genes identified in the paper [9] as critical to distinguishing the classes apart.
The Glioblastoma multiforme (GBM) data-set

- Most aggressive type of primary brain tumor in humans, involving glial cells.
- 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors.
- They are the most prevalent form of primary brain tumors according to a WHO study[6].
- The Glioblastoma multiforme (GBM) data set has 173 samples.
- 28 genes identified in the paper [9] as critical to distinguishing the classes apart.
The Glioblastoma multiforme (GBM) data-set

- Most aggressive type of primary brain tumor in humans, involving glial cells.

- 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors.

- They are the most prevalent form of primary brain tumors according to a WHO study[6].

- The Glioblastoma multiforme (GBM) data set has 173 samples.

- 28 genes identified in the paper [9] as critical to distinguishing the classes apart.
The Glioblastoma multiforme (GBM) data-set

- Most aggressive type of primary brain tumor in humans, involving glial cells.

- 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors.

- They are the most prevalent form of primary brain tumors according to a WHO study[6].

- The Glioblastoma multiforme (GBM) data set has 173 samples.

- 28 genes identified in the paper [9] as critical to distinguishing the classes apart.
Results on GBM Data-set

**Figure:** Four classifiers trained on the GBM data set. X axis: Five “One against Four classes” groups. Y axis: Accuracy. The sigma classifier and the RSVM are almost indistinguishable going by the confidence interval (95%) except where class imbalance shows up. Sample count/class: 'Neural': 26, Pro-Neural: 53, 'Classical': 38, 'Mesenchymal': 56
## Results on GBM Data-set

<table>
<thead>
<tr>
<th>Class division (4 classes)</th>
<th>Linear SVM</th>
<th>Sigma</th>
<th>R SVM</th>
<th>NL SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical vs rest</td>
<td>0.087±0.006</td>
<td><strong>0.046±0.008</strong></td>
<td>0.078±0.008</td>
<td>0.20±0.018</td>
</tr>
<tr>
<td>Mychesmal vs rest</td>
<td>0.0923±0.007</td>
<td><strong>0.037±0.018</strong></td>
<td>0.052±0.009</td>
<td>0.30±0.011</td>
</tr>
<tr>
<td>Neural vs rest</td>
<td>0.083±0.0072</td>
<td><strong>0.076±0.014</strong></td>
<td>0.164±0.0181</td>
<td>0.14±0.007</td>
</tr>
<tr>
<td>Pro Neural vs rest</td>
<td>0.077±0.0075</td>
<td><strong>0.042±0.0064</strong></td>
<td>0.049±0.0094</td>
<td>0.27±0.025</td>
</tr>
<tr>
<td>Neural, Pro Neural vs rest</td>
<td>0.14±0.012</td>
<td><strong>0.079±0.008</strong></td>
<td>0.085±0.008</td>
<td>0.45±0.042</td>
</tr>
</tbody>
</table>

**Table:** Classification error for the 5 data-sets. Row 1 represents the errors when the class “Classical” was taken in Class A and other 3 were taken in class B. The 5th row shows “Neural” and “Pro Neural” in one class and “Classical” and “Mychesmal” in the other. Sigma classifier has lowest mean error (shown in red) but not statistically significant from the RSVM (shown in green). 95% confidence margin.

Patients with BM and without BM.

Eight mi-RNAs were confirmed to be significantly differentially-expressed.

- RSVM and Sigma classifier recover the same set of strong features/genes.

- Genes miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.
Lung Cancer Data-set


- Patients with BM and without BM.

- Eight mi-RNAs were confirmed to be significantly differentially-expressed.
  - RSVM and Sigma classifier recover the same set of strong features/genes.

- Genes miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.

Patients with BM and without BM.

Eight mi-RNAs were confirmed to be significantly differentially-expressed.

- RSVM and Sigma classifier recover the same set of strong features/genes.

- Genes miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.
Lung Cancer Data-set


- Patients with BM and without BM.

- Eight mi-RNAs were confirmed to be significantly differentially-expressed.
  - RSVM and Sigma classifier recover the same set of strong features/genes.

- Genes miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.
Lung Cancer Data-set


- Patients with BM and without BM.

- Eight mi-RNAs were confirmed to be significantly differentially-expressed.
  - RSVM and Sigma classifier recover the same set of strong features/genes.

- Genes miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.
Results on Lung Cancer data-set

\[ \psi(x) = -0.751 \text{(hsa-miR-328)} - 1.425 \text{(hsa-miR-330-3p)} + 0.182 \]

**Figure:** Samples ordered by distance from the RSVM hyperplane positioned at \( Y = 0 \).
Results on Lung Cancer data-set

**Figure:** Area Under the ROC Curve of the RSVM.
## Table: Results for the 3 SVM’s on the lung cancer data-set.

The sigma classifier and the RSVM performance is almost the same (differing only by 1 example which is misclassified). The AUC shows sigma classifier and LSVM with a higher probability of making the right classification on unseen examples.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Linear SVM</th>
<th>Sigma classifier</th>
<th>RSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>0.33</td>
<td>0.2</td>
<td>0.26</td>
</tr>
<tr>
<td>Area Under Curve (AUC)</td>
<td>0.739</td>
<td>0.807</td>
<td>0.705</td>
</tr>
</tbody>
</table>
Summary and my contributions

- Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

- Although LDA classifier with same approach exists[4]. It is important to:
  - Have SVM based classifier because of their popularity and analytically proven strength.

- A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

- Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

- Testing on real life data-sets like GBM and lung cancer.
Summary and my contributions

• Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

• Although LDA classifier with same approach exists[4]. It is important to:
  • Have SVM based classifier because of their popularity and analytically proven strength.

• A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

• Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

• Testing on real life data-sets like GBM and lung cancer.
Summary and my contributions

- Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

- Although LDA classifier with same approach exists[4]. It is important to:
  - Have SVM based classifier because of their popularity and analytically proven strength.

- A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

- Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

- Testing on real life data-sets like GBM and lung cancer.
Summary and my contributions

- Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

- Although LDA classifier with same approach exists[4]. It is important to:
  - Have SVM based classifier because of their popularity and analytically proven strength.

- A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

- Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

- Testing on real life data-sets like GBM and lung cancer.
Summary and my contributions

- Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

- Although LDA classifier with same approach exists[4]. It is important to:
  - Have SVM based classifier because of their popularity and analytically proven strength.

- A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

- Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

- Testing on real life data-sets like GBM and lung cancer.
Summary and my contributions

- Robustness on synthetic data sets to outliers evaluated and found scope for improvement.

- Although LDA classifier with same approach exists[4]. It is important to:
  - Have SVM based classifier because of their popularity and analytically proven strength.

- A more natural view of the loss function developed that is in-line with sample spreading of biological experiments.

- Not let mathematical complexity be a plausible excuse for not evaluating new loss functions.

- Testing on real life data-sets like GBM and lung cancer.
Issues and future work

- Possible Kernalization to support non linearity.
- Class imbalance and sub quadratic convergence due to LBFGS used.
- Implementation of complete second order instead of LBFGS.
- Improve performance on real life data-sets.
Issues and future work

- Possible Kernalization to support non linearity.
- Class imbalance and sub quadratic convergence due to LBFGS used.
- Implementation of complete second order instead of LBFGS.
- Improve performance on real life data-sets
Issues and future work

- Possible kernelization to support non-linearity.
- Class imbalance and sub quadratic convergence due to LBFGS used.
- Implementation of complete second order instead of LBFGS.
- Improve performance on real life data-sets
Issues and future work

- Possible Kernalization to support non linearity.
- Class imbalance and sub quadratic convergence due to LBFGS used.
- Implementation of complete second order instead of LBFGS.
- Improve performance on real life data-sets
Thank you

- Special Thanks to my committee members, Dr Bruno Welfert and Dr Baoxin Li.
- My Sysbio team Archana, Mike and Ina and all those present.
- And most of all my guide Dr Seungchan Kim.
Thank you

- Special Thanks to my committee members, Dr Bruno Welfert and Dr Baoxin Li.
- My Sysbio team Archana, Mike and Ina and all those present.
- And most of all my guide Dr Seungchan Kim.
Thank you

- Special Thanks to my committee members, Dr Bruno Welfert and Dr Baoxin Li.

- My Sysbio team Archana, Mike and Ina and all those present.

- And most of all my guide Dr Seungchan Kim.
Thank you

• Special Thanks to my committee members, Dr Bruno Welfert and Dr Baoxin Li.

• My Sysbio team Archana, Mike and Ina and all those present.

• And most of all my guide Dr Seungchan Kim.
For Further Reading

Y. Bengio and Y. Grandvalet.
No unbiased estimator of the variance of k-fold cross-validation.

O. Chapelle.
Training a support vector machine in the primal.

R.O. Duda, P.E. Hart, and D.G. Stork.
Strong feature sets from small samples.

John Langford.
Clever methods of overfitting.

The 2007 WHO classification of tumours of the central nervous system.
For Further Reading


Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1.

L. Xu, K. Crammer, and D. Schuurmans.
Robust support vector machine training via convex outlier ablation.
*In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 21, page 536. Menlo
Converting primal to dual form

- Primal definition of the SVM:

\[
\min_{w, b} \|w\|^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)
\]

under the constraints:

\[y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0\]

- For \( p = 1 \) the above can be re-written by substituting

\[w = \sum_{i \in [1, N]} \alpha_i y_i x_i\]

instead of \( \|w\|^2 \) in above

\[
L(\alpha) = \sum_{i \in [1, N]} \alpha_i - \frac{1}{2} \sum_{i,j \in [1, N]} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

s.t. \( \alpha_i \geq 0 \)
Gradient

\[ \nabla = \begin{cases} 
2 \cdot w - \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i \cdot \frac{x_i}{\|w\|} - \frac{f_i}{\|w\|} \cdot g_i \cdot \frac{w}{\|w\|} & \text{if } 0 < y_i(w \cdot x_i + b) \leq 1 \\
2 \cdot w + \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i \cdot \frac{x_i}{\|w\|} - \frac{f_i}{\|w\|} \cdot g_i \cdot \frac{w}{\|w\|} & \text{if } y_i(w \cdot x_i + b) < 0
\end{cases} \]

where

\[
\begin{align*}
f_i &= \exp \left( - \left( \frac{\|w \cdot x_i + b\|}{\|w\|\sqrt{2\sigma_{ck}}} \right)^2 \right) \\
g_i &= \frac{w \cdot x_i + b}{\|w\|}
\end{align*}
\]
Outline

5 SVM dual formulation

6 First and second order forms for R-SVM
   - Hessian
Appendix

SVM dual formulation

First and second order forms for R-SVM

Hessian

\[
H = \begin{cases} 
    l - \frac{1}{2\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} & \quad \text{if } 0 < y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \leq 1 \\
    l + \frac{1}{2\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} & \quad \text{if } y (\mathbf{w} \cdot \mathbf{x}_i + b) < 0 \\
\end{cases}
\]

\[
\begin{bmatrix}
    x_1 \partial \left( \frac{f_i}{||\mathbf{w}||} \right) \\
    x_2 \partial \left( \frac{f_i}{||\mathbf{w}||} \right) \\
    \vdots \\
    x_P \partial \left( \frac{f_i}{||\mathbf{w}||} \right)
\end{bmatrix}
- \begin{bmatrix}
    \partial \left( \frac{w_1}{||\mathbf{w}||} \ast (h_i \ast g_i) \right) \\
    \partial \left( \frac{w_2}{||\mathbf{w}||} \ast (h_i \ast g_i) \right) \\
    \vdots \\
    \partial \left( \frac{w_P}{||\mathbf{w}||} \ast (h_i \ast g_i) \right)
\end{bmatrix}
\]