Lecture 6
Empirical risk minimization, Support Vector Machines

Pavel Laskov\textsuperscript{1}  Blaine Nelson\textsuperscript{1}

\textsuperscript{1}Cognitive Systems Group
Wilhelm Schickard Institute for Computer Science
Universität Tübingen, Germany

Advanced Topics in Machine Learning, 2012
Part I

Empirical Risk Minimization
**Loss Function** $\mathcal{L} (\cdot, \cdot)$: a non-negative function that measures disagreement between its arguments

- When we have an estimator function, $f$, we use a loss function to measure how well the estimator agrees with data
- If we have paired data $(x, y)$, the loss of the estimator is $\mathcal{L} (f (x), y)$

**Risk** $R (f, P)$ is the expected risk for data drawn from distribution $P$

$$ R (f, P) = \mathbb{E}_{(x, y) \sim P} [\mathcal{L} (f (x), y)] $$

The **empirical risk** is the average loss of an estimator for a finite set of data drawn from $P$:

$$ R_N (f) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L} (f (x_i), y_i) $$
Examples of Common Loss Functions

- **0-1 Loss**: gives 0 loss when arguments agree, 1 otherwise
  \[ \mathcal{L}_{0-1}(x, y) = I[x \neq y] \]

- **Absolute Loss**: gives the absolute difference in \( x \) & \( y \)
  \[ \mathcal{L}_{abs}(x, y) = |x - y| \]

- **Squared Loss**: gives the squared difference in \( x \) & \( y \)
  \[ \mathcal{L}_{sq}(x, y) = (x - y)^2 \]
The idea of risk minimization is not only measure the performance of an estimator by its risk, but to actually search for the estimator that minimizes risk over distribution \( P \); i.e.,

\[
f^* = \arg\min_{f \in F} R(f, P)
\]

Naturally, \( f^* \) gives the best expected performance for loss \( L \) over the distribution of any estimator in \( F \).
Empirical Risk Minimization

Because we don’t know distribution $P$ we instead minimize empirical risk over a training dataset drawn from $P$

$$f^\dagger = \arg\min_{f \in \mathcal{F}} R_N(f)$$

This general learning technique is called empirical risk minimization

- Under stationarity & other regularity conditions, empirical risk minimizers converge ($f^\dagger \to f^*$), etc.
- To prevent overfitting, the risk is often regularized to penalize complex hypotheses; this is called regularization

$$f^\dagger = \arg\min_{f \in \mathcal{F}} [R_N(f) + c\rho(f)]$$
L2 Regression & Ridge Regression

These forms of regression minimize squared loss for a linear regressor:
\[ f(x) = \mathbf{w}^\top \mathbf{x} + b \]

Ridge regression adds a regularizer for the squared norm \( \| \mathbf{w} \| \)

Principal Component Analysis (PCA)

PCA optimization minimizes the squared norm of the residual from its subspace
Part II

Convex Optimization Theory
Suppose we want to optimize the function $f$ over space $\mathcal{X}$:

$$\min_{x \in \mathcal{X}} [f(x)]$$

Minimizing a function can be done by finding points where the derivative is zero, $\nabla_x f(x) = 0$, a stationary point of $f$.

Minima can also be found by maximizing $-f$.

For this approach, $f$ must be everywhere differentiable.

Not all zero points of $\nabla_x f$ are extrema & may only be local optima.
Suppose we want to optimize the function $f$ over space $\mathcal{X}$:

$$\min_{x \in \mathcal{X}} f(x)$$  \hspace{1cm} (1)

such that

$$g_i(x) \leq 0 \quad \forall \ i = 1 \ldots M$$

$$h_j(x) = 0 \quad \forall \ j = 1 \ldots P$$

where $g_i$ are inequality constraints and $h_j$ are equality constraints.

Such optimizations are called (optimization) programs.
Lagrangian

The **Lagrangian function** for an optimization problem

\[
\min_{x \in \mathcal{X}} \quad f(x)
\]

such that

\[
\begin{align*}
& g_i(x) \leq 0 \quad \forall \ i = 1 \ldots M \\
& h_j(x) = 0 \quad \forall \ j = 1 \ldots P
\end{align*}
\]

is the function

\[
\mathcal{L}(x, \lambda, \nu) = f(x) + \sum_{i=1}^{M} \lambda_i g_i(x) + \sum_{j=1}^{P} \nu_j h_j(x)
\]

where \( \lambda \) & \( \nu \) are **Lagrange multipliers** or **dual variables**
Convex Objective Functions

- A function $f$ is **linear** (or affine) when it can be expressed as:
  \[ f(x) = w^\top x + b \]
  for some constant $w$ and $b$.
- A function $f$ is **convex** when $\forall x, y$ and $t \in [0, 1]$:
  \[ f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \]

Alternatively, $f$ is convex iff its Hessian matrix is **positive semidefinite**.

Program (1) is a **convex program** if $f(\cdot)$ is convex, every inequality constraint $g_i(\cdot)$ is convex, & every equality constraint $h_j(\cdot)$ is linear.

Convex programs can be solved efficiently with special purpose solvers.
KKT Theorem

Theorem 1 (Karush-Kuhn-Tucker Theorem)

If Program (1) is superconsistent & convex & $\mathcal{L}(\cdot)$ is continuously differentiable, then $\mathbf{x}^*$ is a solution iff there is $\mathbf{\lambda}^* \succeq 0$ & $\mathbf{\nu}^* \in \mathbb{R}^P$ s.t.

1. $\forall \mathbf{x}, \mathbf{\lambda} \succeq 0$ we have

$$
\mathcal{L}(\mathbf{x}^*, \mathbf{\lambda}, \mathbf{\nu}) \leq \mathcal{L}(\mathbf{x}^*, \mathbf{\lambda}^*, \mathbf{\nu}^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{\lambda}^*, \mathbf{\nu}^*)
$$

2. $\nabla \mathcal{L}(\cdot) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{M} \lambda_i^* \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^{P} \nu_j^* \nabla h_j(\mathbf{x}^*) = \mathbf{0}$

3. $\forall i = 1, \ldots, M$ $\lambda_i^* \cdot g_i(\mathbf{x}^*) = 0$

- The first consequence means a solution $\mathbf{x}^*, \mathbf{\lambda}^*, \mathbf{\nu}^*$ is a saddle point
- The second allows us to find $\mathbf{x}^*$ in terms of the dual variables by differentiating & setting it equal to $\mathbf{0}$
- The third gives the KKT (complementary slackness) conditions
Lagrangian Optimization Example

Here we show an illustration of maximizing $f(x, y)$ with respect to a constraint that $g(x, y) \leq c$.
Solving \( \frac{\partial}{\partial x_j} \mathcal{L} (x, \lambda, \nu) = 0 \) w.r.t. each primal variable \( x_j \) allows us to express optimal primal variables \( x^* \) in terms of dual variables.

Substituting \( x^* \) into \( \mathcal{L} (x, \lambda, \nu) \) gives us the dual Lagrangian:

\[
\mathcal{L}_d (\lambda, \nu) = \mathcal{L} (x^*, \lambda, \nu) = \inf_x \mathcal{L} (x, \lambda, \nu)
\]

Maximizing \( \mathcal{L}_h (\lambda, \nu) \) with \( \lambda \succeq 0 \) gives optimal dual variables \( \lambda^* \) & \( \nu^* \).

Using these we obtain solutions for the corresponding optimal primal variables \( x^*(\lambda^*, \nu^*) \) at this stationary point of the Lagrangian.

When the program is *convex*, \( x^*(\lambda^*, \nu^*) \) is a solution to Program (1) & \( f (x^*) = \mathcal{L}_d (\lambda^*, \nu^*) \).
Part III

Support Vector Machines (SVM)
Two-Class Classification

Suppose we have a training set \( \mathbb{D} = \{(x_i, y_i)\} \) with \( y_i \in \{-1, +1\} \) — a classification task.

We want to find a function \( f \) such that \( f(x_i) < 0 \iff y_i = -1 \) then we can predict \( y \) with the classifier: \( \text{sign}(f(x_i)) \).

A classifier is consistent if \( y_i f(x_i) > 0 \) for all \( i = 1 \ldots N \).

If there is a consistent classifier \( f \in \mathcal{F} \) for dataset \( \mathbb{D} \), that dataset is separable for \( \mathcal{F} \).

We consider linear functions in feature space of the form

\[
f(x) = w^T \phi(x) + b
\]

when is a dataset is separable under this class we call it linearly separable.
Maximal Margin Classifier

- When a dataset is separable, often there are many consistent classifiers; which is best?
  - If data points lie very close to the boundary, the classifier may be consistent but is more “likely” to make errors on new instances from the distribution.
  - Hence, we prefer classifiers that maximize the minimal distance of data points to the separator.

- **Margin** $\gamma_f$: the gap between data points & the classifier boundary. For *linear* classifiers this is

$$\gamma_f = \min_i y_i f(x_i)$$

- **Maximal Margin Classifier**: a classifier in the family $\mathcal{F}$ that maximizes the margin

$$f^\dagger = \arg\max_{f \in \mathcal{F}} \gamma_f$$
We want to find the *linear* separator with the largest margin.

Thus we solve the following *convex program*:

\[
\max_{w, b, \gamma} \gamma \\
\text{such that } y_i(w^T \phi(x_i) + b) \geq \gamma \quad \forall \ i \\
\|w\|^2 \leq 1
\]

A solution \((w^*, b^*)\) to Program (2) is called a hard margin support vector machine (SVM).
The Lagrangian for Program (2) is

\[
\mathcal{L}(\mathbf{w}, b, \gamma, \alpha, \lambda) = -\gamma - \sum_{i=1}^{N} \alpha_i [y_i (\mathbf{w}^\top \phi(\mathbf{x}_i) + b) - \gamma] + \lambda (\|\mathbf{w}\|^2 - 1)
\]

The derivatives of \(\mathcal{L}(\cdot)\) w.r.t. primal variables are:

\[
\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\cdot) = -\sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i) + 2\lambda \mathbf{w} = 0 \quad \Rightarrow \quad \mathbf{w} = \frac{1}{2\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i)
\]

\[
\frac{\partial}{\partial \gamma} \mathcal{L}(\cdot) = -1 + \sum_{i=1}^{N} \alpha_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i = 1
\]

\[
\frac{\partial}{\partial b} \mathcal{L}(\cdot) = -\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \Rightarrow \quad \mathbf{\alpha}^\top \mathbf{y} = 0
\]
Substituting $w$ into $L(\cdot)$ yields a form, which we optimize w.r.t. $\lambda$ to find

$$
\lambda^* = \frac{1}{2} \left( \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j) \right)^{1/2} = \frac{1}{2} \sqrt{\alpha^\top G \alpha}
$$

where $G_{i,j} = K_{i,j} y_i y_j$.

This yields the following Dual Lagrangian:

$$
L(\alpha) = -\sqrt{\alpha^\top G \alpha}
$$
A solution to the Hard Margin SVM is found by solving

\[
\min_{\alpha} \quad W(\alpha) = \alpha^\top G \alpha
\]

such that

\[
\alpha^\top y = 0 \quad \alpha^\top 1 = 1 \quad \alpha \succeq 0
\]

yielding \( \alpha^* \) for which \( \gamma^* = \sqrt{W(\alpha^*)} \) and \( w^* \propto \sum_i \alpha_i^* y_i \phi(x_i) \)

The SVM can alternatively be minimized by minimizing \( \|w\|^2 \) with a functional margin of 1. The dual is then

\[
\max_{\alpha} \quad W(\alpha) = 1^\top \alpha - \frac{1}{2} \alpha^\top G \alpha
\]

such that

\[
\alpha^\top y = 0 \quad \alpha \succeq 0
\]

giving \( \gamma^* = (1^\top \alpha)^{-1/2} \)
The KKT conditions for the Hard Margin SVM are

\[ \alpha_i^* [y_i (w^* \phi(x_i) + b^*) - \gamma^*] = 0 \quad \forall \ i \]

This means either 1) \( \alpha_i^* = 0 \) or 2) \( y_i (w^* \phi(x_i) + b^*) = \gamma^* \)

The first case, \( i \in \mathbb{R} \), does not further restrict the \( i^{th} \) point

The second case, \( i \in \mathbb{S} \), requires that the point \( \phi(x_i) \) lies on the margin; \( i.e., \) it is exactly \( \gamma \) distant from the separating hyperplane—these points are called support vectors
Properties of Hard Margin SVM

**Sparseness:**
- All points with $\alpha_i$ (i.e., those not on the boundary) do not contribute to the classification function.
- The SVM has thus *learned* which points are critical to making the distinction between 2 classes.
- This also reduces the computation for the classifier... to classify a new point $x$, the kernel $\kappa(x_i, x)$ only needs to be computed for $i \in S$.
Part IV

Soft Margin Support Vector Machine
Soft Margin Classifier

- Not all datasets are linearly separable
- We can allow violations & penalize their sum in our objective function
- We allow outliers to violate the margin, but we don’t want too many violations
  - The 0-1 Loss counts # of violations:
    \[ \mathcal{L}(y, f(x)) = I[y \cdot f(x) < 0] \]
    - This loss is non-convex & hard to optimize
- Instead we can use an upper bound on 0-1 Loss
  - The margin violation of each data point is given by a Hinge Loss
  \[ \xi_i = (\gamma - y \cdot f(x))_+ \]
  \[ = \max(0, \gamma - y \cdot f(x)) \]
  - This is a convex upper bound on 0-1 Loss
Soft Margin Classifier

- Not all datasets are linearly separable
- We can allow violations & penalize their sum in our objective function
- We allow outliers to violate the margin, but we don’t want too many violations
  - The 0-1 Loss counts # of violations:
    \[ \mathcal{L}(y, f(x)) = I[y \cdot f(x) < 0] \]
    This loss is non-convex & hard to optimize
- Instead we can use an upper bound on 0-1 Loss
  - The margin violation of each data point is given by a Hinge Loss
    \[ \xi_i = (\gamma - y \cdot f(x))_+ = \max(0, \gamma - y \cdot f(x)) \]
    This is a convex upper bound on 0-1 Loss
Soft Margin SVM Optimization

- We want to find the *linear* separator with the largest margin / smallest slack penalty.

- Thus we solve the following *convex program*:

\[
\begin{align*}
\max_{w, b, \xi, \gamma} & \quad \gamma - C \sum_{i=1}^{N} \xi_i \\
\text{such that} & \quad y_i (w^T \phi(x_i) + b) \geq \gamma - \xi_i \quad \forall \ i = 1 \ldots N \\
& \quad \|w\|^2 \leq 1 \\
& \quad \xi \succeq 0
\end{align*}
\]  

where \( C > 0 \) controls the trade-off between margin maximization & margin violation.

- A solution \((w^*, b^*, \xi^*)\) to Program (3) is called a *soft margin support vector machine (SVM)*.
The Lagrangian for Program (3) is

\[ \mathcal{L}(w, b, \xi, \gamma, \alpha, \beta, \lambda) = -\gamma + C \sum_{i=1}^{N} \xi_i + \lambda(\|w\|^2 - 1) - \sum_{i=1}^{N} \beta_i \xi_i - \sum_{i=1}^{N} \alpha_i [y_i(w^\top \phi(x_i) + b) - \gamma + \xi_i] \]

The derivatives of \( \mathcal{L}(\cdot) \) w.r.t. primal variables are:

\[ \frac{\partial}{\partial w} \mathcal{L}(\cdot) = -\sum_{i=1}^{N} \alpha_i y_i \phi(x_i) + 2\lambda w = 0 \quad \Rightarrow \quad w = \frac{1}{2\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \]

\[ \frac{\partial}{\partial \gamma} \mathcal{L}(\cdot) = -1 + \mathbf{1}^\top \alpha = 0 \quad \Rightarrow \quad \mathbf{1}^\top \alpha = 1 \]

\[ \frac{\partial}{\partial b} \mathcal{L}(\cdot) = -\alpha^\top \mathbf{y} = 0 \quad \Rightarrow \quad \alpha^\top \mathbf{y} = 0 \]

\[ \frac{\partial}{\partial \xi} \mathcal{L}(\cdot) = C \mathbf{1} - \alpha - \beta = 0 \quad \Rightarrow \quad \alpha = C \mathbf{1} - \beta \]
Substituting $\mathbf{w}$ into $\mathcal{L}(\cdot)$ yields a form, which we optimize w.r.t. $\lambda$ to again find $\lambda^* = \frac{1}{2} \sqrt{\mathbf{\alpha}^\top \mathbf{G}\mathbf{\alpha}}$

This again yields the following Dual Lagrangian:

$$\mathcal{L}(\mathbf{\alpha}) = -\sqrt{\mathbf{\alpha}^\top \mathbf{G}\mathbf{\alpha}}$$
A solution to the soft margin SVM is found by solving

$$\min_\alpha W(\alpha) = \alpha^\top G \alpha$$

such that

$$\alpha^\top y = 0 \quad \alpha^\top 1 = 1 \quad 0 \leq \alpha \leq C$$

yielding $\alpha^*$ for which $w^* \propto \sum_i \alpha_i^* y_i \phi(x_i)$

The SVM can alternatively by minimizing $\|w\|^2$ with a functional margin of 1. The dual is then

$$\max_\alpha W(\alpha) = 1^\top \alpha - \frac{1}{2} \alpha^\top G \alpha$$

such that

$$\alpha^\top y = 0 \quad 0 \leq \alpha \leq C$$
The KKT conditions for the soft margin SVM are

\[ \xi_i^* \beta_i^* = \xi_i^* (\alpha_i^* - C) = 0 \quad \forall \ i = 1 \ldots N \]

\[ \alpha_i^* [y_i (w^* \phi(x_i) + b^*) - \gamma^* + \xi_i^*] = 0 \quad \forall \ i = 1 \ldots N \]

The 1\textsuperscript{st} condition means either \( \xi_i^* = 0 \) or \( \alpha_i^* = C \)

The 2\textsuperscript{nd} means either \( \alpha_i^* = 0 \) or \( y_i (w^* \phi(x_i) + b^*) + \xi_i^* = \gamma^* \)

Combining the 2 cases, \( 0 < \alpha_i^* < C \) implies the \( i \)\textsuperscript{th} point lies on the margin (\textit{unbounded support vectors}, \( i \in \mathbb{S} \)) whereas \( \alpha_i^* = C \) implies \( \xi_i \neq 0 \) (\textit{error support vectors}, \( i \in \mathbb{E} \))

Since any point \( i \in \mathbb{S} \) lies on the boundary, it must satisfy

\[ y_i (w^* \phi(x_i) + b^*) + \xi_i^* = \gamma^* \]

This lets us compute \( \gamma^* \) \& \( b^* \) from the unbounded support vectors
Part V

One-Class Novelty Detection
Novelty Detection with Hyperspheres

- We seek a center $\mathbf{c}$ that minimizes the maximum distance to data

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \max_{i} \| \phi(x_i) - \mathbf{c} \|$$

- The corresponding radius is given by

$$r^* = \max_{i} \| \phi(x_i) - \mathbf{c}^* \|$$
Novelty Detection with Hyperspheres

We seek a center \( c \) that minimizes the maximum distance to data

\[
    c^* = \arg\min_c \max_i \| \phi(x_i) - c \|
\]

The corresponding radius is given by

\[
    r^* = \max_i \| \phi(x_i) - c^* \|
\]
The primal program for the smallest enclosing hypersphere is

\[
\begin{align*}
\min_{c,r} & \quad r^2 \\
\text{such that} & \quad \|\phi(x_i) - c\|^2 \leq r^2 \quad \forall \ i = 1 \ldots N
\end{align*}
\]

The corresponding dual program is

\[
\begin{align*}
\max_{\alpha} & \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i K_{i,i} - \alpha^\top K \alpha \\
\text{such that} & \quad \mathbf{1}^\top \alpha = 1 \quad \text{and} \quad \alpha \succeq 0
\end{align*}
\]

A solution \( \alpha^* \) yields \( r^* = \sqrt{W(\alpha^*)} \) \& \( c^* = \sum_i \alpha_i^* \phi(x_i) \)
To shrink the hypersphere, we exclude outliers with penalized slack:

$$\xi_i = \left( \|c - \phi(x_i)\|^2 - r^2 \right)_+$$

The primal program for the soft minimal hypersphere is

$$\begin{align*}
\min_{c, r, \xi} & \quad r^2 + C \mathbf{1}^\top \xi \\
\text{such that} & \quad \|\phi(x_i) - c\|^2 \leq r^2 + \xi_i \quad \forall \ i = 1 \ldots N \\
& \quad \xi \succeq 0
\end{align*}$$

The corresponding dual program is

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{N} \alpha_i K_{i,i} - \alpha^\top K \alpha$$

such that $$\mathbf{1}^\top \alpha = 1$$ and $$0 \preceq \alpha \preceq C$$
Soft Minimal Hypersphere II

- A solution $\alpha^*$ yields $c^* = \sum_i \alpha_i^* \phi(x_i)$
- Points with $\alpha_i = 0$ lie within the hypersphere, those with $\alpha_i = C$ are outside it, & all others lie on it
- Again this yields sparsity since points within it have $\alpha_i = 0$
- For any $i$, $0 < \alpha_i < C \Rightarrow$

\[
\begin{align*}
    r^2 &= \|\phi(x_i) - c\|^2 \\
        &= (\phi(x_i) - \sum_j \alpha_j^* \phi(x_j))^\top (\phi(x_i) - \sum_j \alpha_j^* \phi(x_j)) \\
        &= K_{i,i} - 2 \sum_j \alpha_j^* K_{i,j} + \alpha^\top K \alpha
\end{align*}
\]

Thus, the optimal radius can be selected by averaging:

\[
r^* = \sqrt{\frac{1}{|S|} \sum_{i \in S} (K_{i,i} - 2 \sum_j \alpha_j^* K_{i,j} + \alpha^\top K \alpha)}
\]
Part VI

Revisiting Regression
Ridge Regression

- Ridge regression can be formulated as risk minimization; the loss in this case is *squared error*.

- The primal program for ridge regression is

\[
\min_w \sum_{i=1}^N \xi_i^2 \quad \text{(6)}
\]

such that

\[
\xi_i = y_i - w^\top \phi(x_i) \quad \forall \ i = 1 \ldots N
\]

\[
\|w\|^2 = B
\]

where \( B > 0 \) regularizes the solution.

- The dual is given by

\[
\min_{\alpha \geq 0} 2\alpha^\top y - \lambda \alpha^\top \alpha - \alpha^\top K\alpha
\]

which has the solution \( \alpha^* = (K + \lambda I)^{-1}y \) & \( w^* = \sum_{i=1}^N \alpha_i^* \phi(x_i) \)
**$\epsilon$-Insensitive Losses**

- Penalizing the regressor for small errors $|\xi_i| < \epsilon$ is overly strict.
- Instead, we can ignore such errors in our loss. This leads to the $\epsilon$-insensitive loss:

$$L (y, f(\phi(x))) = |y - f(\phi(x))|_{\epsilon} = \max(0, |y - f(\phi(x))| - \epsilon)$$

- Alternatively, we can use a quadratic $\epsilon$-insensitive loss

$$L (y, f(\phi(x))) = |y - f(\phi(x))|^2_{\epsilon}$$

- These alternative notions of loss encourage sparseness in the regressor.
The primal program for \( \epsilon \)-insensitive regression is

\[
\min_{w,b,\xi,\hat{\xi}} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} (\xi_i + \hat{\xi}_i)
\]

such that

\[
(w^T \phi(x_i) - b) - y_i \leq \epsilon + \xi_i \quad \forall \ i = 1 \ldots N
\]

\[
y_i - (w^T \phi(x_i) - b) \leq \epsilon + \hat{\xi}_i \quad \forall \ i = 1 \ldots N
\]

\[
\xi, \hat{\xi} \succeq 0
\]

The corresponding dual program is

\[
\max_{\alpha} \quad \alpha^T y - \epsilon \sum_i |\alpha_i| - \frac{1}{2} \alpha^T K \alpha
\]

such that

\[
1^T \alpha = 0
\]

\[
-C \preceq \alpha \preceq C
\]
Quadratic $\epsilon$-insensitive Regression

The primal program for quadratic $\epsilon$-insensitive regression is

$$\min_{w, b, \xi, \hat{\xi}} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} (\xi_i^2 + \hat{\xi}_i^2)$$

such that

$$(w^\top \phi(x_i) - b) - y_i \leq \epsilon + \xi_i \quad \forall i = 1 \ldots N$$

$$y_i - (w^\top \phi(x_i) - b) \leq \epsilon + \hat{\xi}_i \quad \forall i = 1 \ldots N$$

The corresponding dual program is

$$\max_{\alpha} \quad \alpha^\top y - \epsilon \sum_i |\alpha_i| - \frac{1}{2} \alpha^\top (K + \frac{1}{C} I) \alpha$$

such that

$$1^\top \alpha = 0$$
Summary

- We introduced the risk minimization framework from Statistics
- We formalized the optimization tools we need to solve large risk minimization problems
- We then applied risk minimization optimizations to problems for...
  - Classification (SVMs)
  - Novelty Detection (minimal hyperspheres)
  - & Regression ($\epsilon$-insensitive regression)
- These techniques are efficient, but impractical for large problems.
- In next lecture, we will see how problems like these can be solved approximately for large datasets
The Majority of the work from this talk can be found in the lecture's accompanying book, “Kernel Methods for Pattern Analysis.” Also, for a list of applications of the SVM to real-world problems see http://www.clopinet.com/SVM.applications.html