



Digital Media Laboratory
Sharif University of Technology

Statistical Pattern Recognition

Support Vector Machine (SVM)

Hamid R. Rabiee

Hadi Asheri, Jafar Muhammadi, Nima Pourdaghani

Spring 2012

<http://ce.sharif.edu/courses/90-91/2/ce725-1/>

Agenda

- ✧ **Introduction**
 - ✧ **Some Concepts**
- ✧ **Linear, Hard-Margin SVM**
- ✧ **Linear, Soft-Margin SVM**
- ✧ **Nonlinear SVM**
- ✧ **SVMs vs. Neural Networks**
- ✧ **Conclusion**



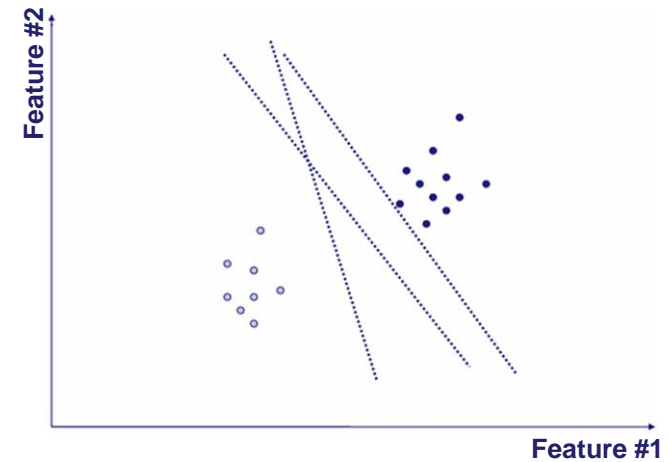
Support Vector Machines

- ✧ **Decision surface is a hyperplane (line in 2D) in feature space.**
- ✧ **SVM three main ideas:**
 - ✧ **Define what an optimal hyperplane is: maximize margin**
 - ✧ **Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications**
 - ✧ **Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space**



Concepts

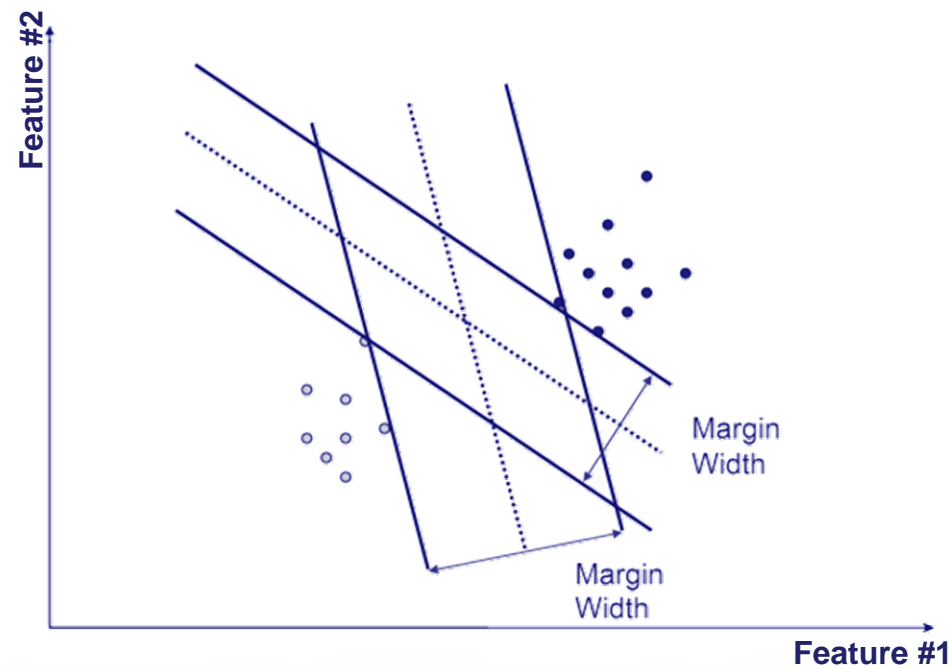
- ✧ **Which of the hyperplanes should we choose?**
 - ✧ Intuitively, a hyperplane that passes too close to the training examples will be sensitive to noise and, therefore, less likely to generalize well for data outside the training set.
 - ✧ Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will have better generalization capabilities.



Concepts

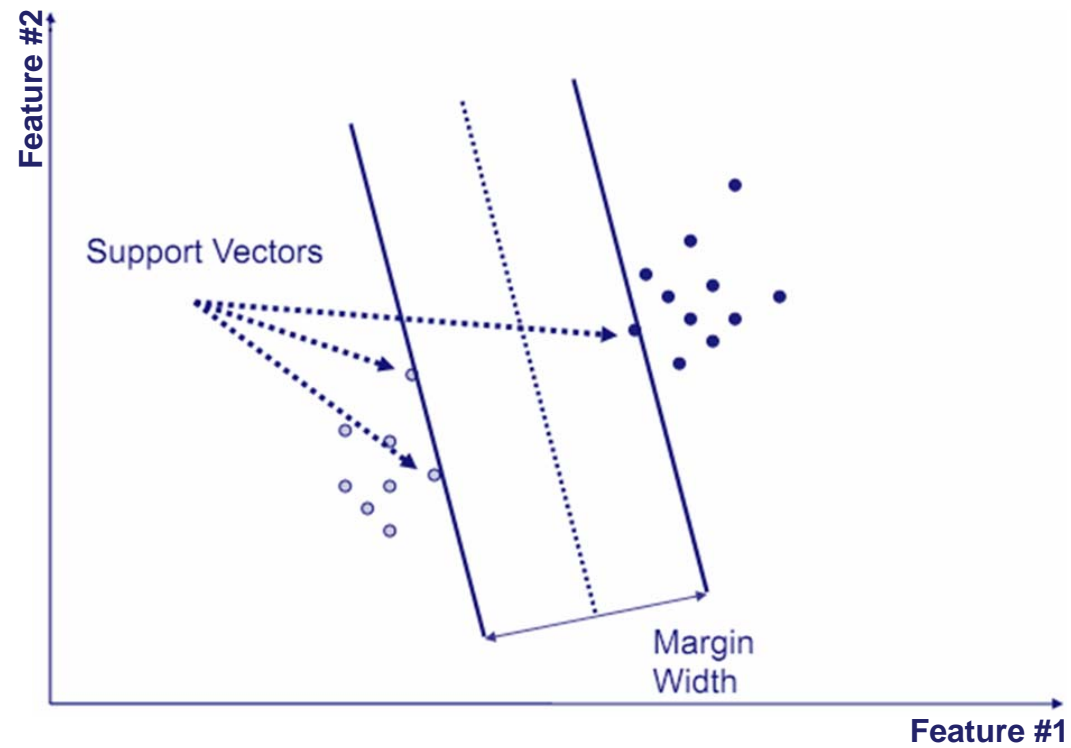
✧ Maximizing the Margin

- ✧ The optimal separating hyperplane will be the one with the largest margin, which is defined as the minimum distance of an example to the decision surface.



Concepts

✧ Support Vectors



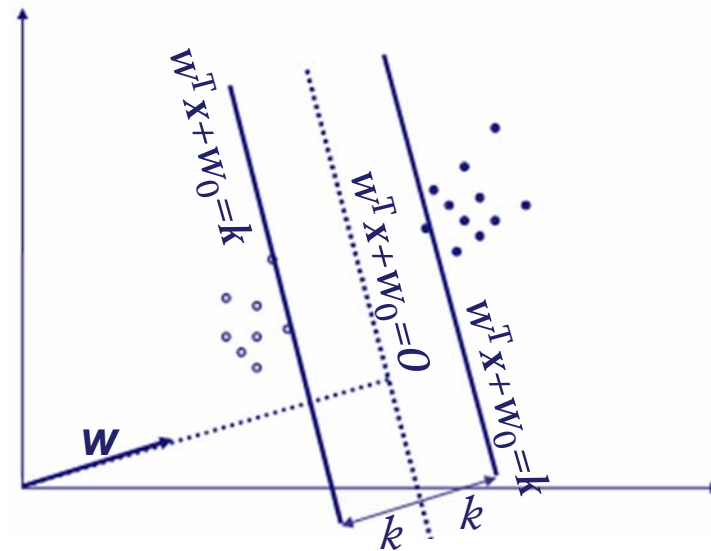
Linear, Hard-Margin SVM

- ✧ The distance of the data point x_0 from the line $G(x)=w^T x+w_0$ will be $\frac{G(x_0)}{\|w\|}$
- ✧ The width of the margin is: $\frac{2|k|}{\|w\|}$
- ✧ So, the problem is:

$$\max \frac{2|k|}{\|w\|} \quad s.t. \begin{cases} w^T x + w_0 \geq k, \forall x \in C_1 \\ w^T x + w_0 \leq -k, \forall x \in C_2 \end{cases}$$

- ✧ Canonical form can be used, without loss the generality, then the problem becomes:

$$\max \frac{2|k|}{\|w\|} \quad s.t. \begin{cases} w^T x + w_0 \geq 1, \forall x \in C_1 \\ w^T x + w_0 \leq -1, \forall x \in C_2 \end{cases}$$



Linear, Hard-Margin SVM

✧ If class 1 corresponds to 1 and class 2 corresponds to -1, we can rewrite

$$\begin{aligned}w^T x + w_0 &\geq 1, \quad \forall x \text{ with } y_i = 1 \\w^T x + w_0 &\leq -1, \quad \forall x \text{ with } y_i = -1\end{aligned}$$

As $y_i(w^T x + w_0) \geq 1, \quad \forall x_i$

✧ So, the problem becomes

$$\max \frac{2}{\|w\|} \text{ or } \min \frac{1}{2} \|w\|^2 \quad \text{s.t. } y_i(w^T x_i + w_0) \geq 1, \quad \forall x$$



Linear, Hard-Margin SVM

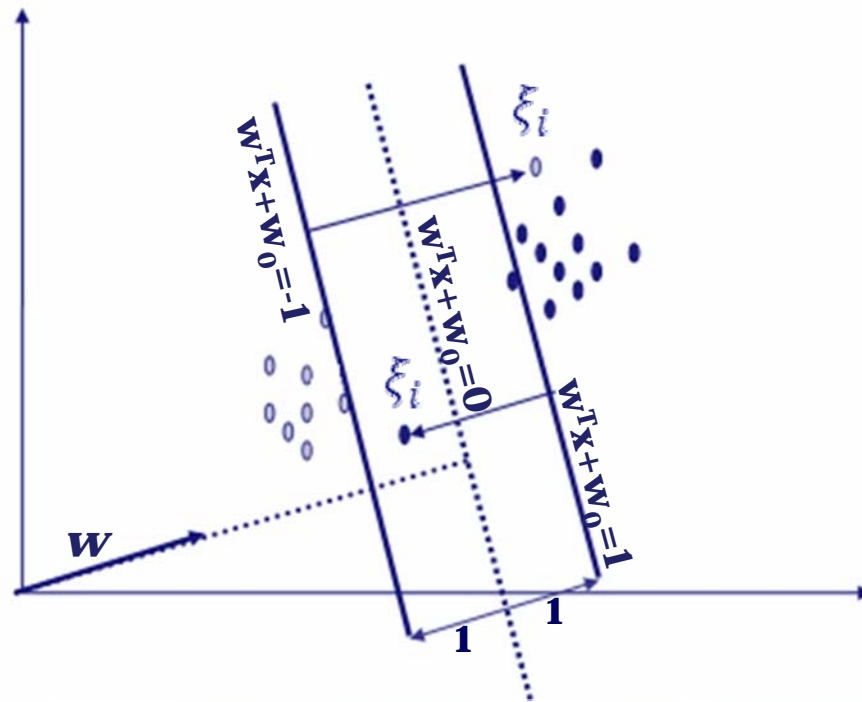
- ✧ **Find w, w_0 that solves $\min \frac{1}{2} \|w\|^2$ s.t. $y_i(w^T x_i + w_0) \geq 1, \forall x$**
 - ✧ **This equation will yield the support vectors, too.**
- ✧ **Problem is convex so, there is a unique global minimum value (when feasible)**
- ✧ **There is also a unique minimizer, i.e. (w, w_0) values that provide the minimum**
- ✧ **Non-solvable if the data is not linearly separable**
- ✧ **Quadratic Programming**
 - ✧ **Very efficient computationally with modern constraint optimization engines**
- ✧ **Refer to Farsi notes for more details**

- ✧ **Q: How to extend the linear definition for non-linearly separable problems?**



Linear, Soft-Margin SVM

- ✧ Will be used in case of non-linearly separable data
- ✧ Introduce slack variables (ξ_i), Allow some instances to fall within the margin, but penalize them

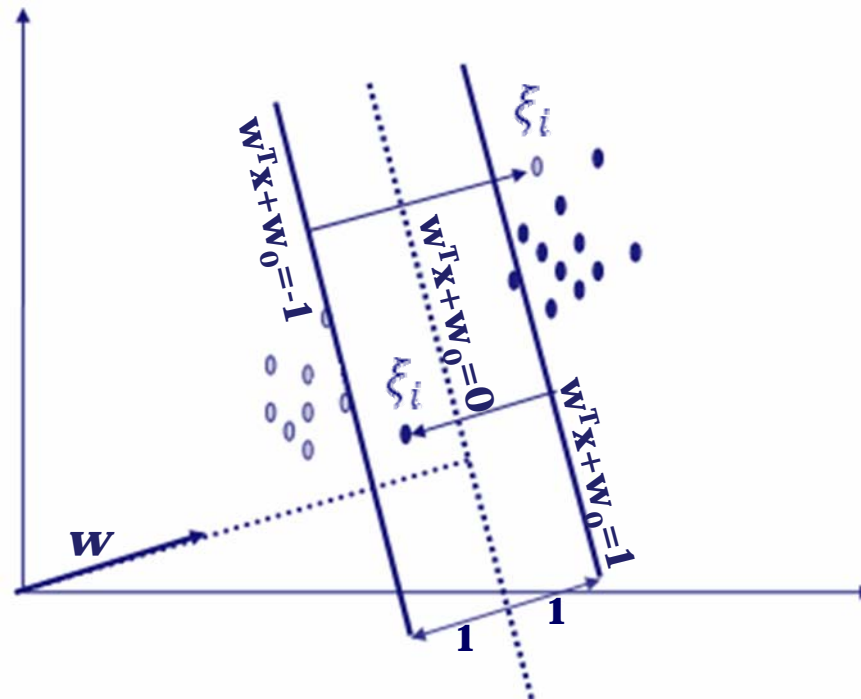


Linear, Soft-Margin SVM

- ✧ **Constraint becomes:** $y_i(w^T x_i + w_0) \geq 1 - \xi_i, \forall x_i, \xi_i \geq 0$
- ✧ **Objective function penalizes for misclassified instances and those within the margin**

$$\min \left\{ \frac{1}{2} \|w\|^2 + c \sum_i \xi_i \right\}$$

- ✧ **C trades-off margin width and misclassifications**



Linear, Soft-Margin SVM

✧ Find w , w_0 (and ξ_i s) that solves

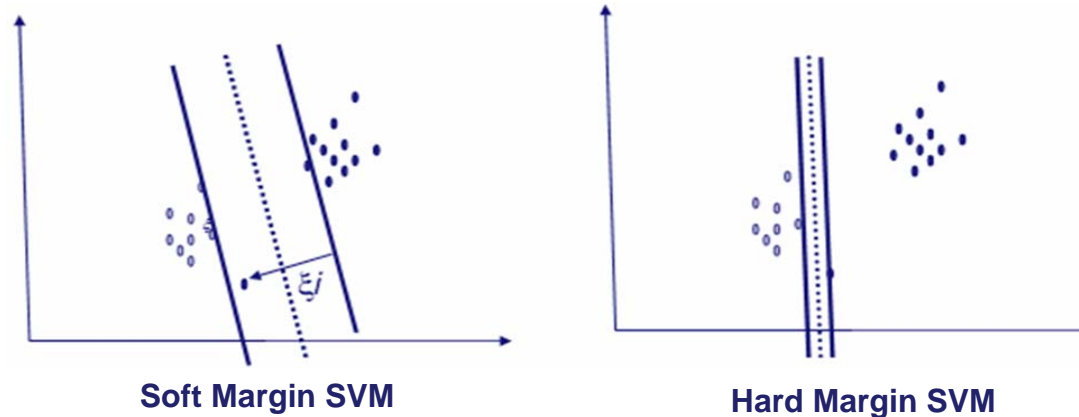
$$\min \left\{ \frac{1}{2} \|w\|^2 + c \sum_i \xi_i \right\} \quad \text{s.t. } y_i (w^T x_i + w_0) \geq 1 - \xi_i, \forall x_i, \xi_i \geq 0$$

- ✧ SVM tries to maintain ξ_i to zero while maximizing margin.
- ✧ Notice: SVM does not minimize the number of misclassifications (NP-complete problem) but the sum of distances from the margin hyperplanes.
- ✧ For bigger c values, we get closer to the hard-margin solution.



Comparison

- ✧ **Soft-Margin always have a solution**
- ✧ **Soft-Margin is more robust to outliers**

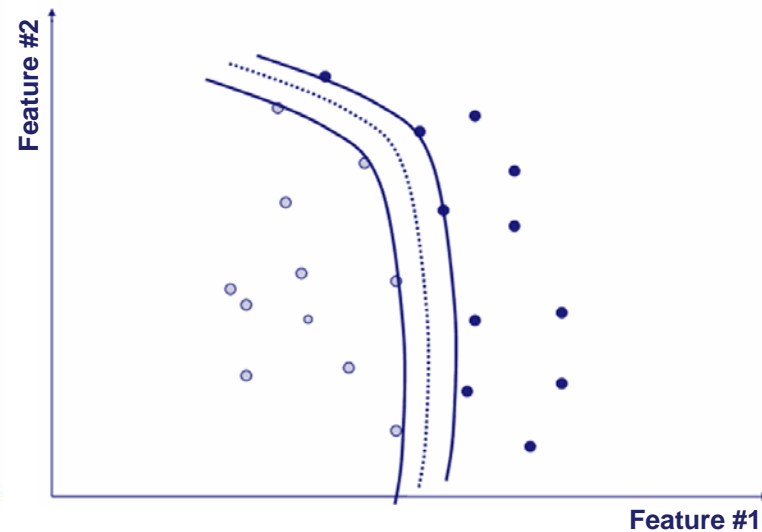
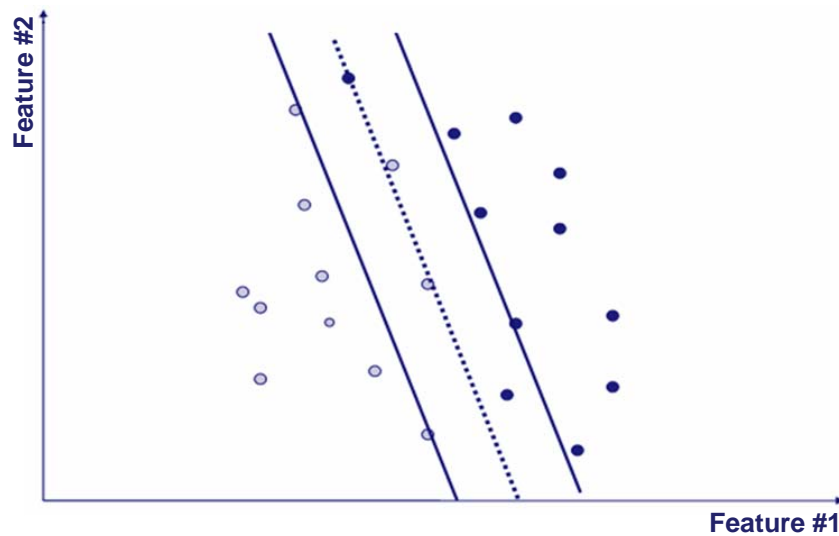


- ✧ **Hard-Margin does not require to guess the cost parameter (requires no parameters at all)**



Nonlinear SVM

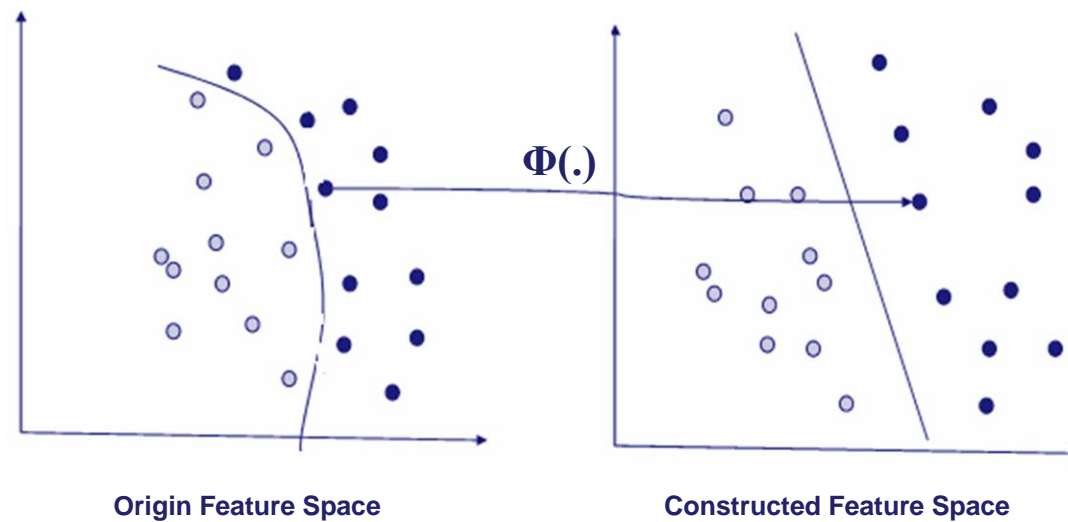
- ✧ **Disadvantages of SVM linear decision surfaces in comparison with nonlinear ones**



Nonlinear SVM

✧ Linear classification in high-dimensional spaces

✧ Find function $\Phi(x)$ to map to a different space



Nonlinear SVM

- ✧ **Using $\Phi(\mathbf{x})$ to map to a different space, SVM formulation in this new space becomes**

$$\min \left\{ \frac{1}{2} \|w\|^2 + c \sum_i \xi_i \right\} \quad \text{s.t. } y_i (w^T \phi(x_i) + w_0) \geq 1 - \xi_i, \forall x_i, \xi_i \geq 0$$

- ✧ **Data appear as $\Phi(\mathbf{x})$, weights w are now weights in the new space**
- ✧ **Explicit mapping expensive if $\Phi(\mathbf{x})$ is very high dimensional**
 - ✧ Solving the problem without explicitly mapping the data is desirable
 - ✧ Use kernel trick
 - ✧ **We'll study "Kernel Methods" in the next session.**
- ✧ **In explicit mapping case, we can solve this optimization problem using its dual form.**



Nonlinear SVM

✧ The Dual of the SVM Formulation

✧ Original SVM formulation

- ✧ n inequality constraints
- ✧ n positivity constraints
- ✧ n number of x variables

$$\min \left\{ \frac{1}{2} \|w\|^2 + c \sum_i \xi_i \right\}$$
$$s.t. \ y_i (w^T x_i + w_0) \geq 1 - \xi_i, \forall x_i, \xi_i \geq 0$$

✧ The (Wolfe) dual of this problem

- ✧ one equality constraint
- ✧ n positivity constraints
- ✧ n number of a variables (Lagrange multipliers)
- ✧ Objective function more complicated

$$\min_{\alpha_i} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j) - \sum_i \alpha_i$$
$$s.t. \ C \geq \alpha_i \geq 0, \sum_i \alpha_i y_i = 0$$



SVMs vs. Neural Networks

SVMs

- ✧ Kernel maps to a high dimensional spaces
- ✧ Search space has a unique minimum
- ✧ Training is extremely efficient
- ✧ Classification extremely efficient
- ✧ Kernel and cost are the two parameters to select
- ✧ Very good accuracy in typical domains
- ✧ Extremely robust

Neural Networks

- ✧ Hidden Layers map to lower dimensional spaces
- ✧ Search space has multiple local minima
- ✧ Training is expensive
- ✧ Classification extremely efficient
- ✧ Requires number of hidden units and layers
- ✧ Very good accuracy in typical domains
- ✧ Could be robust



Conclusions

- ✧ **SVMs express learning as a mathematical program taking advantage of the rich theory in optimization**
- ✧ **SVM uses the kernel trick to map indirectly to extremely high dimensional spaces**
- ✧ **SVMs extremely successful, robust, efficient, and versatile while there are good theoretical indications as to why they generalize well**



Any Question?

End of Lecture 10

Thank you!

Spring 2012

<http://ce.sharif.edu/courses/90-91/2/ce725-1/>

